

Neue Methoden und Techniken für die Evaluation von Visualisierungen

Von der Fakultät Informatik, Elektrotechnik
und Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Michael Longinus Raschke

aus Stuttgart-Bad Cannstatt

Hauptberichter: Prof. Dr. Thomas Ertl

Mitberichter: Prof. Dr. Margit Pohl

Tag der mündlichen Prüfung: 30. November 2015

Institut für Visualisierung und Interaktive Systeme
der Universität Stuttgart

2015

*Für meine Eltern,
Janina,
und Marita, die Liebe meines Lebens*

Wir sind auf der Welt, um sie zu verstehen.

Danksagung

Zu allererst möchte ich mich ganz herzlich bei meinem Doktorvater, Herrn Prof. Thomas Ertl für die Möglichkeit bedanken, bei ihm am Institut promovieren zu dürfen. Ganz besonders lernte ich an ihm zu schätzen, dass er seinen Doktoranden die nötige Freiheit für eine erfolgreiche wissenschaftliche Arbeit gibt, dass ich aber gleichzeitig immer auch das Gefühl gehabt habe, auf einem guten Weg mit meiner Arbeit zu sein!

Ganz herzlich bedanke ich mich bei Frau Prof. Margit Pohl, dass sie den Mitbericht übernommen hat. Herrn Jun.-Prof. Ngoc Thang Vu möchte ich für den Prüfungsbeisitz und Frau Prof. Melanie Herschel für die freundliche Übernahme des Vorsitzes in der Promotionsprüfungskommission danken.

Im ersten Jahr lernte ich sehr viel von Prof. Thomas Schlegel, der damals das Team Interaktive Systeme am Institut leitete. Vielen Dank dafür!

Ein Dank geht an Christiane Taras, die im ersten Jahr am Institut meine Zimmerkollegin war. Ebenfalls möchte ich mich ganz herzlich bei Andre Burkovski und Sebastian Klenk für die vielen inspirierenden Gespräche bedanken. Ein Ergebnis war dabei die 10sekundenKunst, die wir zusammen mit Benjamin Höferlin erfanden. Im ersten Jahr begann ich, ermutigt durch Prof. Thomas Ertl, mit Marianne Richter, einer Philosophin, zusammenzuarbeiten. Vielen Dank für die vielen, dabei entstandenen Quergedanken zur philosophischen Blickrichtung auf die Visualisierung! Bei Florian Haag, meinem zweiten Zimmerkollegen, möchte ich mich für die schöne Zeit bedanken, die wir zusammen in unserem Büro hatten. Vielen Dank an Philipp Heim und Steffen Lohmann für die Inspiration zum Einsatz von Techniken des Semantic Webs in der Evaluation von Visualisierungen.

Ganz herzlich danken möchte ich meiner Kollegin Tanja Blascheck, die mich während meiner Doktorandenzeit zuerst bei der Durchführung meiner Eye-Tracking-Experimente, dann als Diplomandin und später als Zimmerkollegin zur Seite gestanden hat. Zusammen mit unserem Kollegen Michael Burch haben wir das Forschungsgebiet des Eye-Trackings am Institut für uns entdeckt und viele spannende Arbeiten dazu durchgeführt. Später stießen dann zum „VIS-Eye-Tracking-Team“ noch Dominik Herr, Kuno Kurzhals und Hermann Pflüger dazu. Vielen Dank euch allen für die spannenden Gespräche, Inspirationen und den Spaß, den wir zusammen auf Konferenzen hatten.

Ebenso hat es mir auch eine sehr große Freude bereitet mit den Kollegen der Mensch-Computer-Abteilung am Institut zusammenzuarbeiten. Besonders möchte ich mich bei Prof. Albrecht Schmidt, dem Leiter der

MCI-Abteilung, für die vielen inspirierenden Gespräche über die Zukunft der Mensch-Computer-Interaktion bedanken. Ich erinnere mich noch sehr gerne an unsere Reise zur CHI 2014! Bastian Pfleging, Tilman Dingler und Thomas Kubitzka aus der MCI-Gruppe möchte ich ebenfalls danken, mit denen ich neue Interaktionstechniken für die Analyse von Blickdaten entwickelt habe. Und natürlich auch Markus Funk für unser Brain-Painting-Projekt!

Große Anerkennung für ihre Unterstützung und viele interessante Gespräche zu Visualisierung, Mensch-Computer-Interaktion und Computergraphik verdienen auch die restlichen Mitglieder des Instituts. Ganz besonders habe ich mich über die sehr kameradschaftliche Atmosphäre am Institut gefreut! Ihr seid die Besten!

Ganz besonders erinnere ich mich auch an die vielen Gespräche mit Herrn Prof. em. Gunzenhäuser über Kunst, Informatik und Visualisierung und seine Tipps für eine erfolgreiche Promotion.

Während meiner Promotion hatte ich die Möglichkeit, mit unglaublich talentierten, kreativen und engagierten Studenten zusammenarbeiten zu dürfen. Besonders bedanken möchte ich mich bei Xuemei Chen, Stephan Engelhardt, Philipp Schuster und Stefan Strohmaier. Ebenfalls bedanken möchte ich mich bei Jenny Emmrich und Tanja Agapkin für ihre tatkräftige Unterstützung bei der Durchführung meiner Eye-Tracking-Experimente. Bei den vielen anderen, die hier nicht explizit genannt sind, möchte ich mich für die sehr gute Zusammenarbeit bedanken.

Bernhard Schmitz und Michael Wörner möchte ich für das letzte sehr spannenden halbe Jahr als Doktorand am Institut danken, in denen wir die industrielle Anwendung der am Institut entwickelten Eye-Tracking-Analysemethoden erprobten.

Zum Schluss möchte ich mich bei meiner Familie bedanken. Ihr unterstützt mich bei allem was ich tue, euch kann ich bei allem um einen Ratschlag bitten und ihr seid meine Schulter, an der ich mich auch anlehnen kann, wenn etwas mal nicht so läuft wie ich es geplant habe.

Einen ganz besonderes Dank gilt meiner Frau Marita! Marita, Du gibst mir den nötigen Rückhalt und die Unterstützung, um geistig abheben zu können, aber auch sicher wieder auf dem Boden des realen Lebens landen zu können! Von ganzem Herzen möchte ich Dir dafür danken!

Inhaltsverzeichnis

Zusammenfassung	xi
Abstract	xiii
1 Einleitung	1
1.1 Motivation und Ziel der Arbeit	2
1.2 Aufbau und Beitrag der Arbeit	3
2 Grundlagen und Stand der Technik	5
2.1 Grundlagen und Ziel der Visualisierungsforschung	5
Visualisierungs-Pipeline	5
Ziel der visuellen Informationsdarstellung	6
Visuelle Analytik	7
2.2 Visuelle Wahrnehmung	9
Das visuelle System	9
Dreistufiges Modell der visuellen Wahrnehmung	11
2.3 Eye-Tracking	13
2.4 Eye-Tracking-Experimente	14
Ablauf eines Eye-Tracking-Experiments	16
Eye-Tracking-Datentypen	18
Visualisierung von Eye-Tracking-Daten	20
2.5 Vergleich von Scan-Paths	26
Idealer Scan-Path	27
Levenshtein-Distanz	28
2.6 Kognition	30
Das menschliche Gehirn	31
Methode der kognitionswissenschaftlichen Arbeitsweise	32
2.7 Modellierung und Simulation der visuellen Suche	32
Augen-Gedanken-Hypothese (Eye-Mind-Hypothesis)	33
Mentale Repräsentation und Modellierung von visuellem Wissen	34
Leseregeln von Visualisierungen	37
Operatorenbasierte Ansätze in der Mensch-Computer-Interaktion	38
Perzeptions- und Kognitionsmodelle für visuelle Suchstrategien	41
Das Kognitionssimulationsframework ACT-R	42

Simulation der visuellen Suche mit ACT-R	44
CogTool	45
3 Interdisziplinärer Ansatz für die Entwicklung von neuen Methoden und Techniken zur Evaluation von Visualisierungen	47
3.1 Interdisziplinärer Ansatz	51
3.2 Beitrag dieser Arbeit	52
4 Visualisierung von Eye-Tracking-Daten	55
4.1 Parallele Scan-Path-Visualisierung	55
4.2 Visueller Vergleich von Scan-Paths	65
Konzept der Ähnlichkeitsbestimmung	66
Analyse-Pipeline zum Vergleich von Fixationspfaden	70
4.3 Demonstration der PSP-Visualisierungstechnik	73
Analyse von einfachen visuellen Aufgaben	74
Lesestrategien beim Lesen eines Balkendiagramms	79
4.4 Zusammenfassung	81
5 Visuelle Analyse der Wissensverarbeitung	85
5.1 Konzept der visuellen Analyse von Wissensverarbeitungsprozessen .	86
5.2 Wissenmodell auf drei Ebenen	89
5.3 Automatische Annotation von Visualisierungen	92
Konzept der semantischen Annotation	93
Implementierung	94
Diskussion	96
5.4 Visuelle Analyse von kognitiven Prozessen	98
Darstellung der Wissensverarbeitung in PSP-Visualisierungen . . .	98
Anwendungsbeispiele	101
Kognitive Verzerrung	106
5.5 Leseregeln für Visualisierungen	108
5.6 Zusammenfassung und Diskussion	111
6 Modellierung und Simulation der visuellen Suche	115
6.1 Simulation der visuellen Suche	116
Grundlegende Architektur der Simulation	117
Komponenten des Frameworks	117
Anwendung des Simulations-Frameworks	127
Diskussion und Vergleich mit existierenden Arbeiten	129
6.2 Operatorenbasiertes Diagram-Viewing-Modell	132

Das Diagram-Viewing-Modell	132
Ermittlung der Modellparameter	135
Modellparameter	138
Validierung	139
Diskussion	143
6.3 Vergleich der beiden Methoden	146
7 Zusammenfassung und Ausblick	147
7.1 Zusammenfassung	147
7.2 Ausblick	150
Literaturverzeichnis	155

Zusammenfassung

Visualisierungen umgeben uns wie selbstverständlich im Alltag und bei der Arbeit, um abstrakte Informationen darzustellen und komplexe Zusammenhänge zu verstehen. Lag bisher das Hauptaugenmerk der Entwicklung von Visualisierungstechniken auf der Frage, wie möglichst viele Daten in möglichst kurzer Zeit, in einer möglichst hohen Auflösung dargestellt werden können, so gewann in der Visualisierungsforschung in den letzten Jahren die Fragestellung an Bedeutung, ob eine Visualisierung auch nützlich und leicht lesbar ist.

Um diese Fragestellung umfassend beantworten zu können, war das Ziel dieser Arbeit die Entwicklung von neuen Methoden und Techniken zur Untersuchung der Wahrnehmung von Visualisierungen, sowie zur Evaluation von Visualisierungstechniken. Dazu wurde ein interdisziplinärer Ansatz gewählt, der die drei wissenschaftlichen Forschungsgebiete Eye-Tracking, Wissensrepräsentation und Kognitionswissenschaften miteinander verbindet. Eye-Tracking-Experimente wurden für die Analyse des Blickverhaltens bei der Arbeit mit Visualisierungen eingesetzt. Die Repräsentation visuellen Wissens erlaubt es, semantische Eigenschaften von Scan-Paths untersuchen zu können. Simulationsmethoden aus den Kognitionswissenschaften ermöglichen es, das Blickverhalten vorherzusagen.

Eye-Tracking-Experimente werden in der Visualisierungsforschung dazu eingesetzt, um Augenbewegungen von Probanden, welche Aufgaben mit Visualisierungen durchführen, aufzunehmen. Ein nicht zu unterschätzender Zeitaufwand bei der Auswertung dieser Art von Experimenten nimmt die anschließende Analyse der Augenbewegungen ein. Um den Aufwand der Analyse dieser Scan-Paths zu reduzieren und ähnliche Augenbewegungsmuster über die Probanden hinweg zu identifizieren, wurde die parallele Scan-Path-Visualisierungstechnik entwickelt, die eine übersichtliche Darstellung von mehreren Scan-Paths erlaubt. Damit können Lesestrategien von Visualisierungen über mehrere Probanden hinweg erkannt und miteinander verglichen werden. Die parallele Scan-Path-Visualisierung wurde zusätzlich mit automatischen Mustererkennungsverfahren erweitert. Dieser sogenannten visuelle Analytik-Ansatz erlaubt es, Scan-Paths quantitativ miteinander zu vergleichen und führt zu einer effizienten Analyse von sehr großen Eye-Tracking-Datensätzen.

Für die Modellierung von Wissen über Visualisierungen wurde ein Wissensmodell mit drei Ebenen entwickelt. Jede Ebene beschreibt in Form einer Ontologie eine unterschiedliche Abstraktionsebene des Wissens über

Visualisierungen und die darin enthaltenen graphischen Elemente. Elemente aus diesen Ontologien werden mit bestimmten Bereichen in einer Visualisierung oder mit einzelnen graphischen Elementen in Visualisierungen verknüpft. Dieser Ansatz ermöglicht es nicht nur wie bisher zu analysieren, welche Bereiche in einer Visualisierung auf einem Bildschirm in welcher Reihenfolge betrachtet worden sind (WO-Raum), sondern auch, was für graphische Elemente dort wahrgenommen (WAS-Raum) und wie diese kognitiv weiterverarbeitet wurden. Es wird gezeigt, wie mit der parallelen Scan-Path-Visualisierungstechnik, basierend auf dieser Annotation, Wissensverarbeitungsprozesse visualisiert werden können. Damit können auch Bereiche in Visualisierungen, die möglicherweise zu einer kognitiven Verzerrung führen, erkannt und im Detail weiter untersucht werden.

Für die Simulation der visuellen Suche wurde eine auf dem Kognitionssimulationsframework ACT-R basierende Simulation entwickelt, die Leseprozesse in Visualisierungen simuliert, und es erlaubt, diese mit empirisch ermittelten Daten zu vergleichen. Zusätzlich stellt diese Arbeit erstmalig ein operatorenbasiertes Modell zur Vorhersage von Durchführungszeiten von visuellen Aufgaben vor. Dieses operatorenbasierte Diagramm-Viewing-Modell verwendet das Konzept des aus der Mensch-Computer-Interaktionsforschung bekannten Keystroke-Level-Modells und erweitert es für die Vorhersage von Durchführungszeiten von visuellen Aufgaben.

Neben einer Effizienzsteigerung bei der Auswertung von Eye-Tracking-Experimenten führt die Kombination der visuellen Analyse von Scan-Paths mit ontologiebasierten Wissensmodellen zu einem tieferen Verständnis der Leseprozesse von Visualisierungen. Semantische Charakteristika von Scan-Paths können besser untersucht werden und die Wahrscheinlichkeit für kognitive Verzerrungen bei der Arbeit mit Visualisierungen durch eine geeignete Anpassung des Visualisierungskonzepts verringert werden. Insgesamt können die in dieser Arbeit vorgestellten Methoden und Techniken zu einem stärker benutzerorientierten, iterativen Entwicklungsprozess von Visualisierungen führen. In diesem Entwicklungsprozess können Ergebnisse der Eye-Tracking-Analyse oder Ergebnisse aus Simulationen dazu eingesetzt werden, um zu untersuchen, wie Visualisierungen von verschiedenen Benutzergruppen wahrgenommen werden.

Abstract

Visualizations have become a part of our daily life and a state of the art tool at work to analyze abstract data quickly or to understand complex correlations. In the past, one of the goals of visualization research was to show as many possible data points very quickly on high resolution screens. However, during the last years the question of having a visualization technique which is useful and easy to read became important.

This work introduces new methods and techniques for the analysis of the perception of visualizations and for the evaluation of visualizations. Therefore, an interdisciplinary approach was chosen combining three scientific research fields eye tracking, knowledge representation and cognitive science. Eye tracking is used to record and analyze gaze behavior in visualization evaluation experiments. Knowledge representation of visual knowledge is used to study semantic characteristics of readings strategies of visualizations. And, cognition simulations predict gaze behavior of participants in evaluation experiments.

Eye tracking is one of the state of the art techniques in visualization research for recording gaze behavior of participants who have to perform given tasks with visualizations. A not to be underestimated share of time is required for the analysis of the recorded eye movements. The first contribution of this work is a new visualization technique, the so called Parallel Scan-Path Visualization Technique, which allows the analyzer to quickly find interesting eye movement patterns in large eye tracking data sets. Additionally, reading strategies of individual participants or groups of participants can visually be compared with each other. The Parallel Scan-Path Visualization Technique also serves as the visualization component in a new visual analytics approach for the analysis of eye tracking data. Thereby, the analyst is supported by automatic pattern recognition algorithms. This new approach leads to quantitative comparable results during the scan-path comparison process and to a more efficient analysis of large eye movement data sets.

A knowledge model with three levels was developed to describe knowledge about visualizations. Every level of the model represents visual knowledge in a different abstraction in a separate ontology. Elements of every level of this knowledge model are linked to certain areas in visualizations and to graphical elements shown in these areas. Besides analyzing where participants have focused on in visualizations (WHERE space), this approach also allows us to study what they have focused on (WHAT space).

This work shows how this approach can be used in combination with the Parallel Scan-Path Visualization Technique to visually study cognitive processes. Thereby, areas in visualizations can be identified which could possibly lead to a cognitive bias.

In order to simulate gaze behavior, this thesis contributes a cognition simulation based on the ACT-R framework. With this cognition simulation, reading strategies of visualizations can be simulated and results of these simulations can be compared with empirical collected data. Additionally, this thesis contributes a first operator-based model for predicting visual task completion times. This so called Diagram Viewing Model uses the Keystroke-Level Model approach from Human-Computer Interaction research for the computation of visual task completion times. However, the evaluation of this approach shows that there are strong limitations relating to this first model in simulating gaze behavior.

The combination of a visualization-based analysis of scan-paths and an ontology-based knowledge model leads to a deeper understanding of reading processes of visualizations. Semantic characteristics of scan-paths better can be analyzed and the probability for a cognitive bias during working with visualizations can be reduced. In summary, the methods and techniques presented in this thesis lead to a more user-centric, iterative design process of visualizations. In this design process both results from eye tracking experiments and simulations can efficiently be used to better understand the perception of visualizations.

Einleitung

Eine Geschichte über den berühmten Physiker Galileo Galilei erzählt, dass er zwei unterschiedlich schwere Kugeln vom schiefen Turm in Pisa hinunterfallen ließ. Er wollte damit beweisen, dass die Zeit, die ein Körper für eine Strecke im freien Fall benötigt, nicht von seiner Masse abhängt. Und tatsächlich konnte Galilei zeigen, dass beide Kugeln gleichzeitig unten auf dem Boden ankommen. Das gleiche Experiment führen heute immer noch Schüler und Studenten im Unterricht und Studium durch. Den von Galilei gefundenen Zusammenhang zwischen zurückgelegtem Weg und Zeit visualisieren sie meist mit einem Strecke-Zeit-Diagramm (Abbildung 1.1).

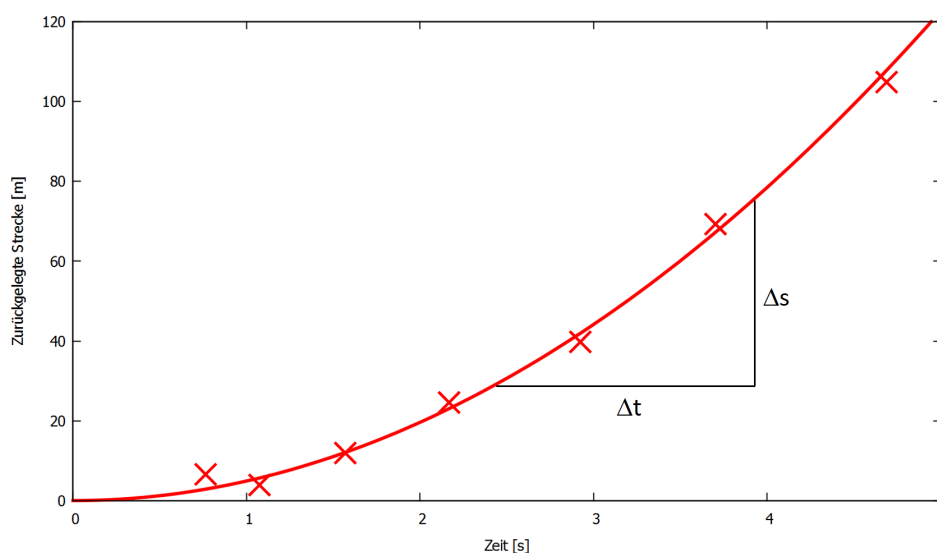


Abbildung 1.1: Visualisierung des freien Falls. Durch die graphische Darstellung des Zusammenhangs zwischen der gemessenen, zurückgelegten Strecke pro Zeit (rote Kreuze) kann das Fallgesetz $s(t) = \frac{1}{2}gt^2$ leichter gefunden werden.

Die Visualisierung von Vorgängen in der Natur ist heute in den Naturwissenschaften zu einem Standardwerkzeug geworden, um experimentelle Daten analysieren oder diese mit Simulationsergebnissen vergleichen zu können.¹ Aber nicht nur in der Wissenschaft helfen Visualisierungen abstrakte Zusammenhänge leichter erkennen zu können, Visualisierungen fanden auch verstärkten Eingang in unseren Alltag. Diese reichen von einfachen Wetterkarten in der morgendlichen Zeitung, über S-Bahnkarten auf dem Weg zur Arbeit, bis hin zu Geschäftsgraphiken oder Visualisierungen am Arbeitsplatz.

Lag bisher meist das Hauptaugenmerk bei der Entwicklung von Visualisierungstechniken in der Visualisierungsforschung auf der Frage, wie möglichst viele Daten in möglichst kurzer Zeit und in einer möglichst hohen Auflösung dargestellt werden können, gewann in den letzten Jahren die Fragestellung an Bedeutung, ob eine Visualisierung auch nützlich und leicht lesbar ist. Die klassische Methode, um die Wahrnehmungseffizienz von Visualisierungen zu untersuchen, sind Benutzerexperimente. Dabei wird in einem kontrollierten Experiment die Durchführungszeit, die ein Proband für eine Aufgabe benötigt hat, zusammen mit der Fehlerrate der korrekt gegebenen Antworten zu dieser Aufgabe gemessen. Sind sowohl die Durchführungszeit, als auch die Fehlerrate für eine gegebene Visualisierung niedrig, so kann davon ausgegangen werden, dass die in dieser Visualisierung dargestellte Information für die im Experiment untersuchte Aufgabe gut erkannt und verstanden werden kann.

1.1 Motivation und Ziel der Arbeit

Da Eye-Tracking-Geräte in den letzten zehn Jahren in ihrer Anschaffung immer günstiger und in ihrer Bedienung einfacher geworden sind, werden neben Durchführungszeiten und Fehlerraten verstärkt auch die Augenbewegungen für die Evaluation von Visualisierungen aufgezeichnet. Damit wird es möglich, einen Einblick in die von den Benutzern angewandten, visuellen Suchstrategien zu erhalten. Neben der Analyse der Aufmerksamkeitsverteilung können dadurch auch die verwendeten Lösungsstrategien für gegebene Aufgaben in einem Experiment im Detail analysiert werden.

Jedoch wird durch die zusätzliche Analyse der Eye-Tracking-Daten der Aufwand für die Auswertung von Benutzerexperimenten nochmals höher. Es wurden zwar in den letzten beiden Jahrzehnten eine Vielzahl von Visualisierungstechniken für die Analyse der Augenbewegungen vorgestellt, die meisten davon besitzen aber den Nachteil, dass sie bei einer höheren

¹Diese Arbeitsweise wurde im Beitrag „Michael Raschke and Thomas Ertl: Geometrical Intelligence“ während des *Leonardo-Kollegs Erlangen* im November 2010 diskutiert.

Anzahl von visualisierten Probandendaten unübersichtlich werden. Hinzu kommt, dass bisher eine allgemein anwendbare Methode fehlt, um Lesestrategien und die dahinter liegenden kognitiven Prozesse näher untersuchen zu können. Es existieren zwar Publikationen, die vielfältigste Fragen der Perzeption und Kognition bei der Arbeit mit Visualisierungen beantworten, aber ein Framework, mit dem kognitive und perzeptive Prozesse für verschiedene Fragestellungen und in unterschiedlichen Detailgraden analysiert werden können, fehlt bisher. Daher ist es auch nicht möglich, Modelle und Simulationen von Augenbewegungen in Evaluationsexperimenten in großem Umfang einzusetzen, um den Aufwand für die Evaluation von Visualisierungstechniken zu reduzieren.

Das Ziel dieser Arbeit war es, neue Methoden und Techniken für die Evaluation von Visualisierungen zu entwickeln, mit der perzeptive und kognitive Prozesse in der Visualisierungsforschung untersucht werden können. Dieses Ziel umfasste die Entwicklung einer neuartigen Methode zur Analyse von kognitiven und perzeptiven Prozessen mit Hilfe von Eye-Tracking, die Erstellung eines geeigneten Wissensmodells über Visualisierungskonzepte und den darin angezeigten graphischen Elementen, die Formulierung eines Vorhersagemodells für Durchführungszeiten von visuellen Aufgaben und die Implementierung einer Kognitionssimulation zur detaillierten Simulation von visuellen Suchstrategien.

1.2 Aufbau und Beitrag der Arbeit

Abbildung 1.2 zeigt die Struktur dieser Arbeit mit ihren einzelnen Beiträgen verteilt auf die einzelnen Kapitel.

Kapitel 2 gibt eine Einführung in die Visualisierung, das Eye-Tracking, sowie in die Modellierung und Simulation von perzeptiven und kognitiven Prozessen. Zu beiden Gebieten wird zusätzlich der Stand der Technik vorgestellt.

Kapitel 3 stellt den dieser Arbeit zugrunde liegenden interdisziplinären Ansatz vor, auf dessen Basis Methoden aus den drei Forschungsgebieten des Eye-Trackings, der Wissensrepräsentation und den Kognitionswissenschaften verbunden worden sind (Raschke et al., 2014a).

In *Kapitel 4* wird die parallele Scan-Path-Visualisierungstechnik vorgestellt, die den Vergleich von Augenpfaden von mehreren Probanden und die Identifikation von ähnlichen Augenbewegungsmustern erleichtert (Raschke et al., 2012). Darauf aufbauend wird eine Methode präsentiert, mit der Scan-Paths verglichen werden können (Raschke et al., 2014d).

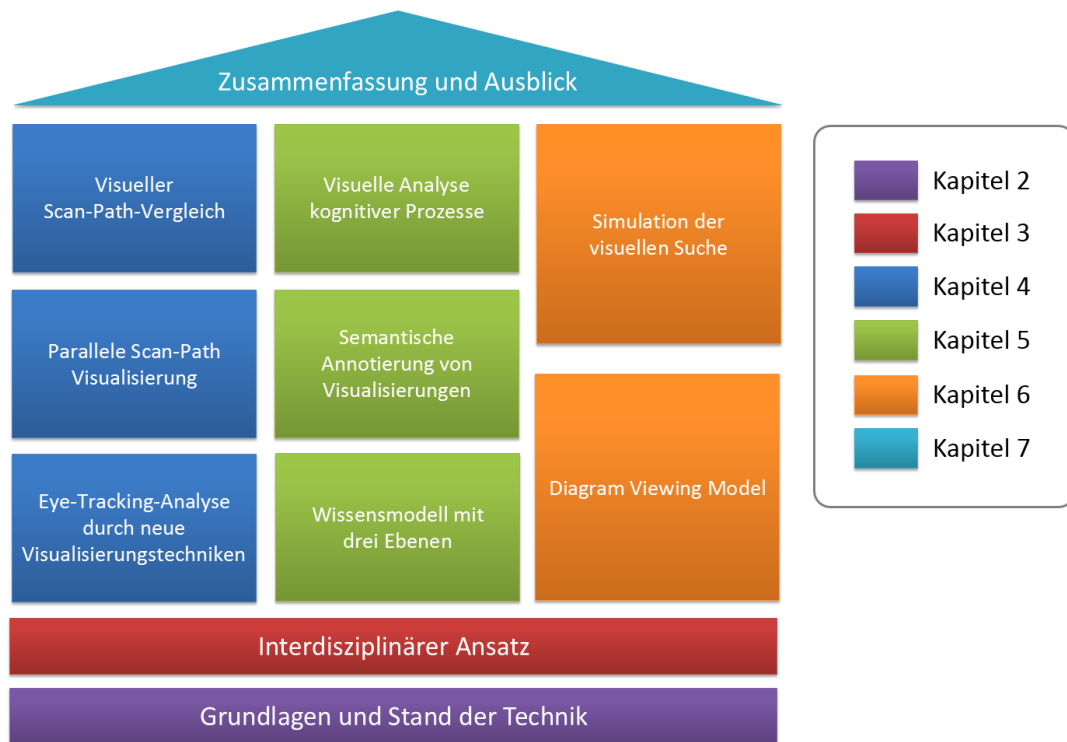


Abbildung 1.2: Aufbau der vorliegenden Arbeit mit ihren Einzelbeiträgen, aufgeteilt in die drei Hauptkapitel Eye-Tracking-Visualisierung (blau), Visuelle Analyse der Wissensverarbeitung (grün) und Modelle der visuellen Suche (orange).

Kapitel 5 beginnt mit der Vorstellung eines Wissensmodells für graphische Elemente und Visualisierungskonzepte auf drei Ebenen (Raschke et al., 2013). Auf Basis einer Methode zur Annotierung von Visualisierungen (Raschke et al., 2014c) wird gezeigt, wie mit Hilfe der parallelen Scan-Path-Visualisierungstechnik kognitive und perzeptive Prozesse untersucht werden können (Raschke et al., 2014b).

In *Kapitel 6* werden zwei unterschiedliche Ansätze vorgestellt, um Visualisierungen zu evaluieren. Der erste Ansatz besteht in einer Kognitionssimulation, die es erlaubt, Lesestrategien zu berechnen (Raschke et al., 2013). Der zweite Ansatz verwendet ein operatorenbasiertes Diagram-Viewing-Model für die Vorhersage der Durchführungszeit von visuellen Aufgaben.

Kapitel 7 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick zu möglichen weiteren Fragestellungen, die mit Hilfe der in dieser Arbeit präsentierten Methoden und Techniken untersucht werden können.

Grundlagen und Stand der Technik

Die folgenden Abschnitte geben einen Überblick über die Grundlagen dieser Arbeit und den Stand der Techniken auf den Gebieten der Visualisierungsforschung, der visuellen Wahrnehmung, des Eye-Trackings und der Analyse von Blickverhaltensdaten, dem Vergleich von Scan-Paths und der Simulation der visuellen Suche.

2.1 Grundlagen und Ziel der Visualisierungsforschung

Visualisierungen stellen Daten in einer visuell erfassbaren Form dar. In der Visualisierungsforschung unterscheidet man grundsätzlich zwei Klassen von Visualisierungen: *Wissenschaftliche Visualisierungen* und *Informationsvisualisierungen*. Wissenschaftliche Visualisierungen stellen zwei- oder dreidimensional, raum-zeitliche Daten dar, Informationsvisualisierungen abstrakt vorliegende Datenstrukturen, die nicht kontinuierlich sind und keine raum-zeitliche Abbildung erlauben. Das Modell der Visualisierungspipeline beschreibt die Visualisierung von Daten in vier Schritten (Abschnitt 2.1.1). Das Ziel von Visualisierungen besteht darin, den Erkenntnisgewinn auf Basis der vorliegenden Daten zu fördern (Abschnitt 2.1.2). Werden Visualisierungen mit automatischen Analyseverfahren kombiniert, spricht man von visueller Analytik (Abschnitt 2.1.3).

2.1.1 Visualisierungs-Pipeline

Das Modell der Visualisierungspipeline beschreibt, wie in vier Schritten aus Rohdaten eine Visualisierung erstellt wird (Haber und McNabb, 1990; Card et al., 1999). Abbildung 2.1 zeigt diese vier Schritte. Zunächst werden die vorliegenden Rohdaten analysiert. In diesem Schritt können fehlende Datenpunkte interpoliert oder korrigiert werden. Im nächsten Schritt werden die Daten durch den Benutzer gefiltert, um festzulegen, welche Daten visualisiert werden sollen. Die dabei ausgewählten Daten werden im dritten

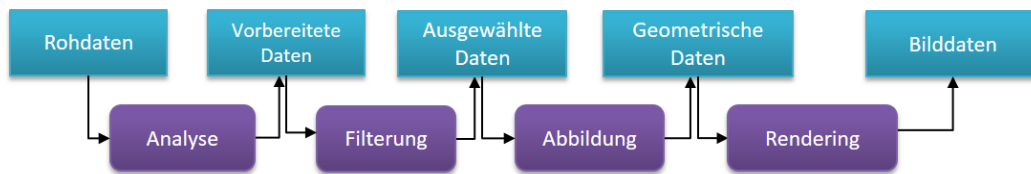


Abbildung 2.1: Das Modell der Visualisierungspipeline beschreibt, wie in vier Schritten aus Rohdaten eine Visualisierung erzeugt wird.

Schritt auf geometrische Formen und Positionen in der Visualisierung abgebildet. Im letzten Schritt werden diese geometrischen Daten gerendert und in Form von Bilddaten ausgegeben. In dieser Arbeit wurde für die Analyse des Blickverhaltens die Visualisierungspipeline mit semantischen Informationen erweitert, die entlang der Visualisierungspipeline bis zur fertigen Visualisierung mittransportiert werden (Abschnitt 5.3).

2.1.2 Ziel der visuellen Informationsdarstellung

Einen großen Einfluss auf den Erfolg einer Visualisierungstechnik hat der kognitive Aufwand, der notwendig ist, um Informationen mit Hilfe der Visualisierung zu lesen und Zusammenhänge darin zu erkennen. Dabei spielen eine Vielzahl von Eigenschaften der Visualisierungen eine Rolle, wie die Helligkeit, der Kontrast und die Farbe von einzelnen graphischen Elementen in der Visualisierung, präattentive Eigenschaften der Anordnung, Gestaltgesetze und Texturen (Ware, 2013, 2008). Sind diese Visualisierungsparameter gut gewählt und geschickt miteinander kombiniert, kann ein Zusammenhang in der Visualisierung mit einem geringen kognitiven Aufwand erkannt werden. Das Gegenteil ist jedoch auch möglich. Dann können Informationen und ihr Zusammenhang aus einer Visualisierung nur schwer gelesen oder sogar vollständig falsch interpretiert und damit nicht verstanden werden. Das Ziel bei der Entwicklung einer Visualisierung sollte deshalb darin bestehen, die Wahrscheinlichkeit für das Auftreten einer *kognitiven Verzerrung* beim Lesen von Informationen aus dieser Visualisierung durch eine geschickte Wahl der Visualisierungsparameter möglichst gering zu halten.

Eine (interaktive) Visualisierung kann als eine interne Schnittstelle zwischen Mensch und Computer für das Lösen eines Problems gesehen werden (Ware, 2013). Norman erkannte, dass Verständnisprobleme bei der Verwendung von Systemen häufig auf unterschiedliche mentale Modelle von Ersteller (Autor) und Nutzer (Betrachter) zurückgeführt werden können (Norman, 1987). Wenden wir seinen Ansatz auf Visualisierungen

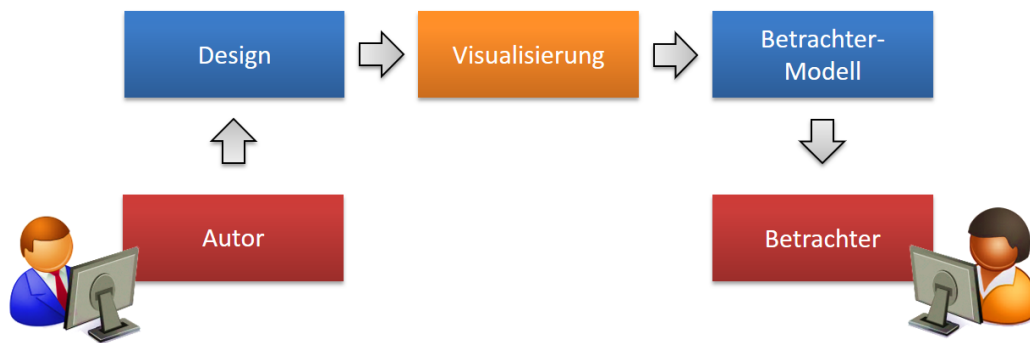


Abbildung 2.2: Das Ziel des Autors einer Visualisierung besteht darin, eine Information graphisch an einen Betrachter der Visualisierung zu übermitteln. Unterschiede in den mentalen Modellen von Ersteller und Nutzer sind häufig der Grund für Verständnisprobleme bei der Betrachtung von Visualisierungen.

an (Abbildung 2.2), so besteht das Ziel des Autors einer Visualisierung darin, eine Information graphisch an einen Betrachter der Visualisierung zu übermitteln. Dies erfolgt durch ein geeignetes Design der Visualisierung durch den Autor, das einer Visualisierung zu Grunde liegt. Beim Betrachten der Visualisierung erzeugt der Betrachter ein mentales Modell der Visualisierung. Basierend auf seiner Interpretation der Darstellung erhält er dadurch ein Verständnis des Inhalts. Im Idealfall stimmen die mentalen Modelle von Ersteller und Nutzer überein. Dies kann zum Beispiel durch eine geschickte Anpassung der Visualisierungsparameter an den Einsatzzweck und an die Fähigkeiten des Betrachters erreicht werden. Stimmen die beiden mentalen Modelle jedoch nicht überein, kommt es zu einem Verständnisproblem zwischen Autor und Betrachter. Tritt dieses auf, kann die in der Visualisierung dargestellte Information nur noch unzureichend oder sogar falsch verstanden werden.

2.1.3 Visuelle Analytik

Das Gebiet der visuellen Analytik beschreibt die Verbindung von hochinteraktiven, visuellen Benutzungsschnittstellen mit automatischen Analyseverfahren. Beim Einsatz der visuellen Analytik wird sowohl die menschliche Fähigkeit verwendet, visuell Muster zu erkennen, als auch die Leistungsfähigkeit von Computern ausgenutzt, formale Verfahren und Algorithmen auf große Datenmengen anzuwenden.

In Abbildung 2.3 ist der explorative Ansatz der visuellen Analytik nach Keim et al. dargestellt (Keim et al., 2010). Grundsätzlich werden in der visuellen Analytik Daten in Wissen transformiert. Keims Modell zeigt wie

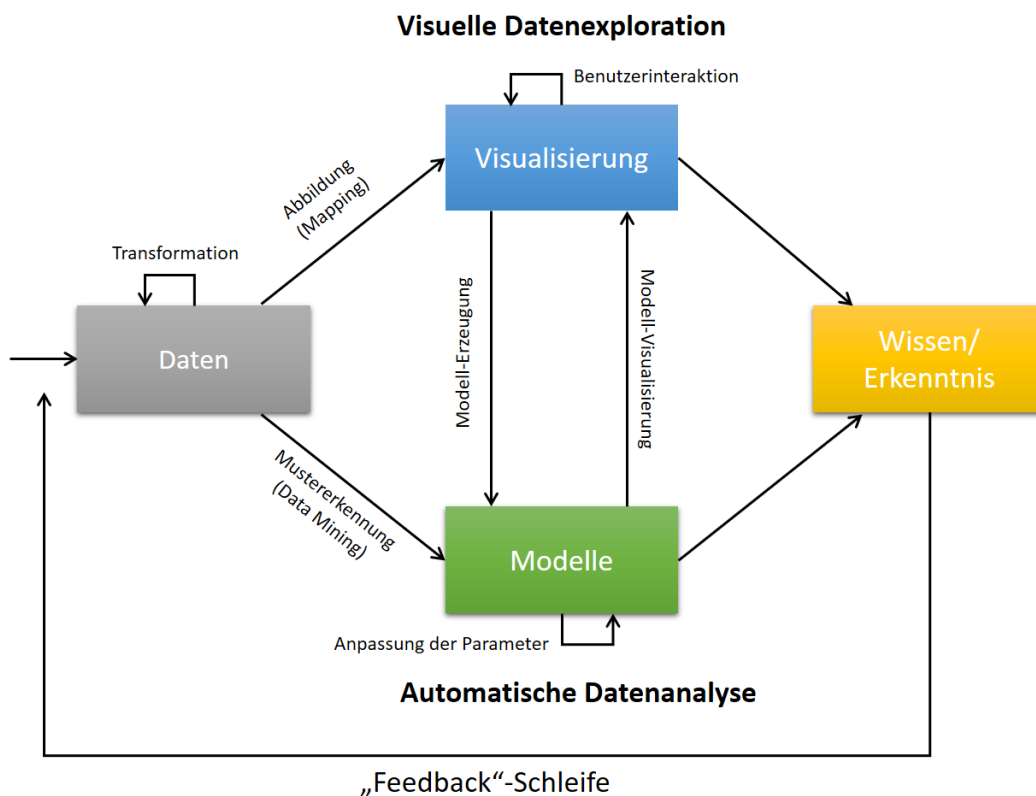


Abbildung 2.3: Explorativer Ansatz der visuellen Analytik nach Keim et al.

dieser Prozess durch die visuelle Analytik mit Hilfe von Modellen und Visualisierungen unterstützt werden kann. Ein Modell repräsentiert dabei die formale „Aufbereitung“ von Daten mit Hilfe von Mustererkennungsverfahren (Data Mining). Diese aufbereiteten Daten werden dem Benutzer über ein Modell in Form einer Visualisierung dargestellt. Eine Visualisierung kann auch ohne diesen Schritt direkt erzeugt werden (Mapping). Der Benutzer betrachtet anschließend die abgebildeten Daten und ändert das zugrunde liegende Visualisierungsmodell. Zusätzlich kann er den Umfang der verwendeten Daten anpassen (Feedback-Schleife).

Neben Anwendungen in den Bereichen der Sicherheit, Gesundheitsforschung, des Energiemanagements, der Umwelt- und Klimaforschung, des Handels, der Logistik, Ernährungs- und Landwirtschaft sowie bei Versicherungen und im Wissensmanagement (Thomas und Kielman, 2009) wird die visuelle Analytik auch im Rahmen dieser Arbeit eingesetzt (Kapitel 4). Auf dem Gebiet der Eye-Tracking-Analyse stellt die visuelle Analytik einen vielversprechenden Ansatz dar, die in Zukunft umfangreich anfallenden Mengen an Eye-Tracking-Daten effizient untersuchen zu können (Blascheck et al., 2015a).

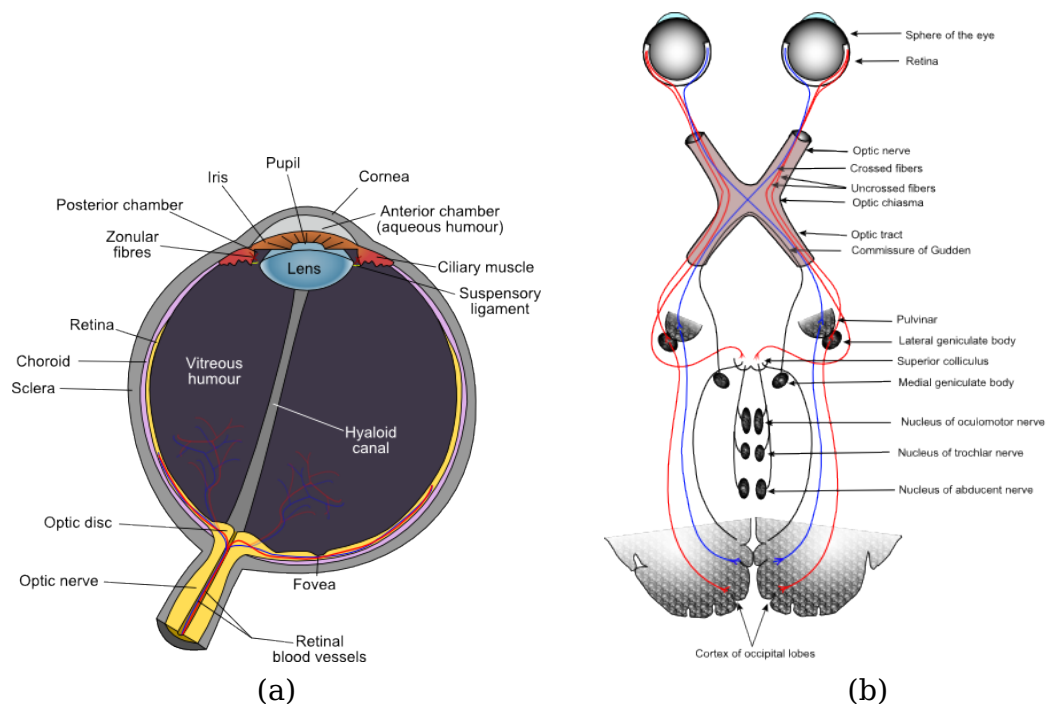


Abbildung 2.4: a) Das menschliche Auge: Licht fällt durch die Iris und die Pupille in das Auge, wird von der Linse gebrochen und auf die Netzhaut geworfen¹. b) Optisches Chiasma: Die Sehnerven der beiden Augen kreuzen sich und leiten die visuelle Information an den visuellen Cortex im hinteren Bereich des Gehirns weiter.²

2.2 Visuelle Wahrnehmung

Visualisierungen werden über das visuelle System wahrgenommen. Dieses besteht im Wesentlichen aus den Augen, die Licht in elektrische Signale umsetzen, und dem Sehnerv, der diese Signale an den visuellen Cortex im Gehirn weiterleitet (Zimbardo und Gerrig, 2008). Neben der anatomischen Beschreibung der Wahrnehmung (Abschnitt 2.2.1) wird in diesem Abschnitt ein dreistufiges Modell der Perzeption von Ware vorgestellt (Ware, 2013) (Abschnitt 2.2.2).

2.2.1 Das visuelle System

Das menschliche Auge (Abbildung 2.4a) funktioniert wie eine Kamera. Das Licht tritt durch die Hornhaut ein. Um die Menge an Licht ändern zu können, die in das Auge eintritt, kann sich der Durchmesser der Iris mit Hilfe von Muskeln ändern. Direkt hinter der Iris befindet sich die Pupille.

¹Quelle: https://en.wikipedia.org/wiki/File:Schematic_diagram_of_the_human_eye_en.svg

²Quelle: <https://en.wikipedia.org/wiki/File:Gray722-svg.svg>

Um ein scharfes Bild zu erzeugen, verändert die Augenlinse ihre Form und ändert über diese Akkommodation ihre optischen Eigenschaften. Sie wird flacher, um entfernte Objekte scharf zu stellen, und gekrümmter, um ein scharfes Bild von nahen Objekten zu erzeugen. Das Licht durchquert anschließend den Glaskörper und fällt auf die Netzhaut, eine dünne Schicht auf der Rückseite des Augapfels.

Die Umwandlung von Licht in Nervensignale geschieht in der Netzhaut. Dazu wandeln zwei Arten von lichtempfindlichen Zellen, die Stäbchenzellen und Zapfenzellen, das einfallende Licht in elektrische Signale um. Die 120 Millionen Stäbchenzellen im Auge arbeiten am effizientesten bei schwachem Licht. Mit ihnen kann das Auge bei geringer Helligkeit, in der Nacht oder in der Dämmerung, visuelle Reize wahrnehmen. Die Stäbchen erlauben nur ein monochromatisches Sehen, da alle Stäbchenzellen nur für das Licht im Bereich der Wellenlänge von ca. 500 nm (Blaugrün) empfindlich sind. Im Außenbereich um die Netzhautmitte befinden sich mehr Stäbchenzellen als Zapfenzellen, wodurch der Mensch bei Dämmerung in der Peripherie besser sieht als im Zentrum. Sieben Millionen Zapfenzellen sind auf das helle Tageslicht spezialisiert. Nahe des Zentrums der Netzhaut befindet sich eine kleine Region mit einer hohen Dichte von Zapfenzellen. Die Fovea ist die Region des schärfsten Sehvermögens, in der sowohl Farben, aber auch räumliche Details mit der größten Genauigkeit wahrgenommen werden können. Um den Bereich der Fovea herum schließt sich der sogenannte parafoveale Sehbereich an. Dieser brauchbare Sehbereich (useful field of view) kann für die periphere Wahrnehmung von graphischen Elementen in Visualisierungen verwendet werden. Bei einer hohen Dichte an graphischer Information in diesem Feld ist dieser brauchbare Sehbereich zwischen 1° und 4° groß (Wickens, 1992), bei einer geringen Dichte von graphischer Information bis zu 15° (Drury und Clement, 1978). Eine erhöhte Konzentration während der Durchführung einer visuellen Aufgabe führt zu einer Verringerung des brauchbaren Sehfeldes (Williams, 1985). Dieser „Tunnelblick“ des Sehfelds spielt bei der Analyse von Eye-Tracking-Daten eine wichtige Rolle, da die Genauigkeit der Position einer Fixation eine ausschlaggebende Rolle für eine korrekte Analyse der Aufmerksamkeitsverteilung und Blickpfade spielt.

Signale von Stäbchen- und Zapfenzellen werden durch Bipolarzellen kombiniert und an Ganglienzellen weitergeleitet. Jede Ganglienzelle integriert dann die Impulse der Bipolarzellen zu einer einzigen Folge von Impulsen. Die Axone der Ganglienzellen bilden den Sehnerv, der diese integrierte, visuelle Information an den visuellen Cortex des Gehirns im Hinterkopf weiterleitet. Zwischen den Augen und dem primären visuellen Cortex kreuzen sich der Sehnerv des linken und des rechten Auges im optischen Chiasma (Abbildung 2.4b). Die Axone der Sehnerven beider

Augen teilen sich dort in zwei Bündel. Dabei verbleibt eine Hälfte auf der gleichen Seite des Körpers, von der sie stammen. Die Axone der zur Nase zugewandten Hälfte des Sichtfeldes überkreuzen sich. Damit beinhalten die beiden zum visuellen Cortex weiterlaufenden Sehnerven Informationen von jeweils beiden Augen. Zum visuellen Cortex gehören der primäre visuelle Cortex sowie weitere Gebiete im Hinterhauptslappen des Gehirns. In diesen findet eine Parallelverarbeitung der visuellen Information statt (siehe dazu Abschnitt 2.6.1).

2.2.2 Dreistufiges Modell der visuellen Wahrnehmung

Ware stellt ein einfaches Modell der visuellen Wahrnehmung mit drei Stufen vor (Ware, 2013). Dieses ist in Abbildung 2.5 schematisch dargestellt und beschreibt die Verarbeitung der perceptiven Wahrnehmung in den drei Schritten:

1. Wahrnehmung von einfachen Eigenschaften einer visuellen Szene durch parallele Verarbeitung des visuellen Reizes,
2. Mustererkennung und Aufteilung in zwei Pfade,
3. Abspeichern der erkannten visuellen Objekte im visuellen Arbeitsgedächtnis.

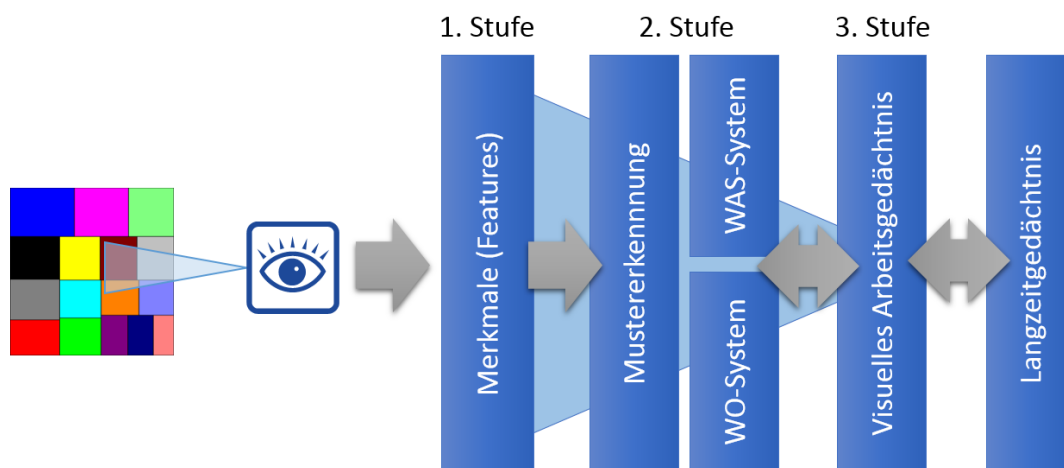


Abbildung 2.5: Dreistufiges Modell der visuellen Wahrnehmung nach Ware mit paralleler Verarbeitung von einfachen Eigenschaften einer visuellen Szene, Mustererkennung und dem visuellen Arbeitsgedächtnis.

Erste Stufe: Parallele Verarbeitung von einfachen Eigenschaften einer visuellen Szene

Visuelle Reize werden parallel durch eine sehr große Anzahl von Nervenzellen im Auge und im visuellen Cortex verarbeitet. Dazu sprechen spezielle Nervenzellen auf bestimmte Eigenschaften in einer visuellen Szene an, wie beispielsweise die Orientierung von Kanten oder die Farbe eines Bereichs auf der Netzhaut. In diesem Schritt arbeiten Milliarden von Nervenzellen parallel. Das Ergebnis dieses ersten Schrittes nennt Treisman eine Merkmalskarte der visuellen Szene (Treisman, 1985). Eine gute Visualisierung nutzt diese Eigenschaft der visuellen Wahrnehmung aus. In diesem Schritt spielt ein höheres, kognitiv gesteuertes Ziel bei der Betrachtung einer Visualisierung noch keine Rolle.

Zweite Stufe: Mustererkennung

In der zweiten Stufe wird das visuelle Sehfeld in einzelne Regionen und einfache Muster unterteilt. Dies können kontinuierliche Konturen oder Regionen mit der gleichen Farbe oder der gleichen Textur sein. Der Schritt der Mustererkennung ist extrem flexibel und wird sowohl von den aus dem ersten Schritt gewonnenen Informationen wie auch aus der von kognitiven Prozessen gesteuerten Wahrnehmung beeinflusst. Marr nennt diesen Schritt den „2-1/2 sketch“ (Marr, 1982), Rensink den „Proto-Object Flux“ (Rensink, 2002).

In diesem Schritt teilt sich die Wahrnehmung auch in zwei Pfade auf (Neurowissenschaften, 2001; Pasternak et al., 2003; Rao et al., 1997): ein Pfad verarbeitet die visuelle Information bezüglich der darin enthaltenen Muster, also wie Objekte aussehen. Dieses System wird das *WAS-System* genannt. Der andere Pfad verarbeitet die räumliche visuelle Information, wo sich Objekte im Raum befinden. Dieses System wird in der Literatur auch *WO-System* genannt.

Dritte Stufe: Visuelles Arbeitsgedächtnis

Im letzten Schritt werden erkannte Muster im visuellen Arbeitsgedächtnis gehalten. Eine Reihe von visuellen Anfragen wird erstellt, um mit einer visuellen Suchstrategie beantwortet zu werden. Diese visuelle Suchstrategie hängt maßgeblich von der Aufgabe ab, die mit einer Visualisierung durchgeführt werden soll. So könnte der Betrachter in einem Scatterplot gleich eingefärbte Punkte suchen, um einen bestimmten Datensatz besser zu verstehen. Nur eine begrenzte Anzahl von visuellen Objekten kann bei dieser Suchstrategie im Arbeitsgedächtnis behalten werden. Diese sind

aus Mustern und Informationen aus dem Langzeitgedächtnis zusammengesetzt, die aus dem zweiten Schritt stammen. In dieser Stufe spielen erlernte Leseregeln von Visualisierungen eine wichtige Rolle (siehe dazu Abschnitt 2.7).

2.3 Eye-Tracking

Mit Hilfe von Eye-Tracking können Augenbewegungen aufgezeichnet werden. Typische Anwendungsfelder sind das Marketing, die Wahrnehmungspsychologie, Usability-Forschung und Mensch-Computer-Interaktion (Duchowski, 2002). Zur Aufnahme der Augenbewegungen wurden im Laufe der letzten Jahrzehnte verschiedene Techniken entwickelt (Duchowski, 2007):

- **Elektrookulographie:** Mit Hilfe von Elektroden, die um das Auge herum angebracht werden, wird die Aktivität der Augenmuskeln gemessen und damit die Blickrichtung bestimmt.
- **Sclera-Kontaktlinsen-Methode:** Der Proband trägt eine Kontaktlinse, an die ein kleiner Spiegel oder eine Spule angebracht ist. Die Spule befindet sich in einem Magnetfeld. Durch eine induzierte Spannung wird die Blickrichtung ermittelt.
- **Photo-Oculography (POG) / Video-Oculography (VOG):** Beide Methoden verwenden rotations- oder translationsunabhängige Merkmale des Auges wie die Form der Pupille, die Position des Limbus oder corneale Reflektionen. Beide Methoden lassen meist keine Blickrichtungsmessung zu und sind aufwändig in ihrer Datenaufbereitung.
- **Videobasierte Methode:** Diese Technik wird heute in kommerziell angebotenen Eye-Tracking-Geräten eingesetzt. Videobasierte Eye-Tracking-Geräte verwenden Kameras in Kombination mit Bildverarbeitungsmethoden, um die Blickrichtung des Auges zu ermitteln. Dazu wird Infrarotlicht in die Richtung des Auges gestrahlt. Die Position der Reflektion des Infrarotlichts auf der Hornhaut (corneale Reflexion) wird relativ zum Mittelpunkt der Pupille gemessen. Aufgrund der Anatomie des Auges (Abschnitt 2.2.1) entstehen vier Reflektionen des Infrarotlichts im Auge, die sogenannten Purkinje Reflektionen (Crane, 1994). Bei der videobasierten Methode wird die erste und vierte Purkinje Reflektion verwendet. Diese Dual-Purkinje-Image (DPI) Eye-Tracker (Crane und Steele,

1985) liefern nach einer Kalibration präzise Informationen über Blickpunkte, die sich in einer Ebene vor dem Probanden befinden. Die Aufnahme­rate hängt dabei vom Eye-Tracking-Gerät ab und liegt in der Regel zwischen 60 Hz und 240 Hz, je nach Einsatzzweck des Eye-Tracking-Geräts.

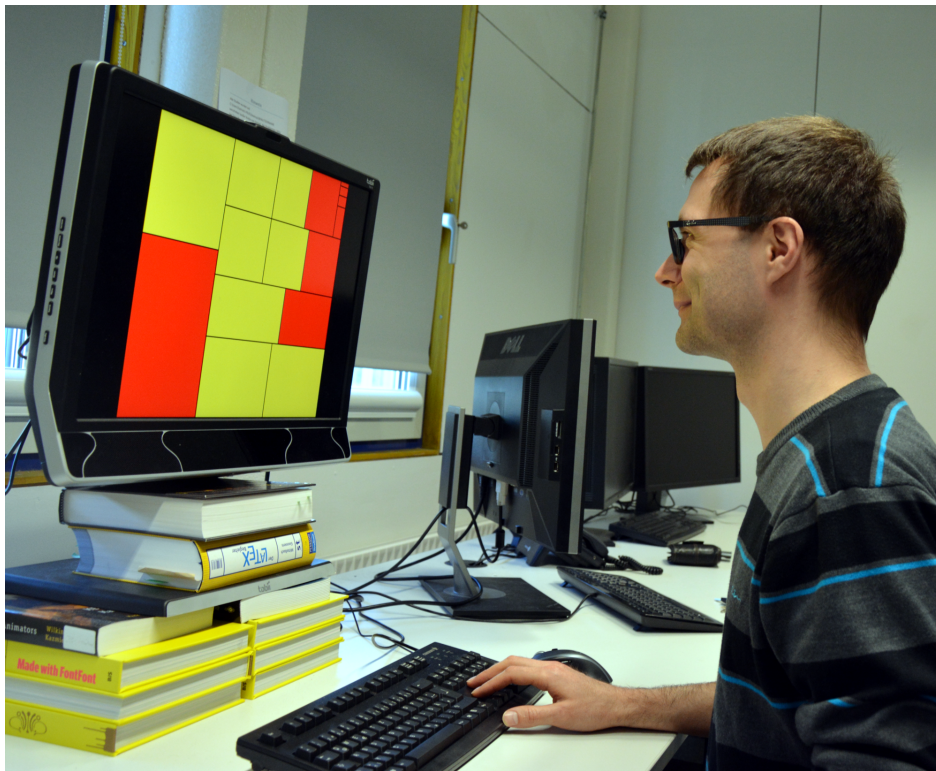
Bei der videobasierten Methode unterscheidet man grundsätzlich zwei Arten von Eye-Tracking-Geräten (Abbildung 2.6): Stationäre Geräte, die beispielsweise in einem PC-Bildschirm integriert sind (Desktop-Systeme), und am Kopf getragene, mobile Systeme (Head-Mounted-Systeme).

2.4 Eye-Tracking-Experimente

Da die Anschaffungskosten für Eye-Tracking-Geräte seit der Jahrtausend­wende stark gefallen sind, wird Eye-Tracking in der Visualisierungsforschung immer häufiger eingesetzt. Dabei können zwei Arten von Experimenten unterschieden werden.

Bei Evaluationsexperimenten werden Visualisierungen auf ihre Lesbarkeit getestet. Mit dieser Art von Experimenten wurden beispielsweise verschiedene Layouttechniken von Graphdarstellungen (Burch et al., 2011, 2012; Goldberg und Helfman, 2011), verschiedene Visualisierungstechniken von Tag-Clouds (Lohmann et al., 2009) oder von zweidimensionalen Scatter-Plots (Etemadpour et al., 2015) verglichen.

Bei Wahrnehmungsexperimenten werden graphische Eigenschaften von Visualisierungen untersucht, die zu einer guten oder schlechten Lesbarkeit einer Visualisierung führen. Dieser Klasse von Experimenten lassen sich auch Experimente zur Untersuchung von Lesestrategien zuordnen. Ein Beispiel dafür ist die Analyse des Lesens von Graphen (Huang und Eades, 2005; Huang et al., 2009; Pohl et al., 2009). Die Autoren konnten zeigen, dass die Versuchsteilnehmer in ihrem Blickverhalten der Geodäte zwischen zwei im Graphen markierten Punkten mit ihren Augen folgen. Ein ähnliches Verhalten zeigt sich auch beim Lesen von Metro-Karten (Burch et al., 2014a,b). Weitere Studien untersuchen beispielsweise die Verbesserung der Wahrnehmung durch das zusätzliche Einblenden von graphischen Hilfselementen (Steinberger et al., 2011), die Wahrnehmung von geometrischen Formen (Prats et al., 2010) oder den Einfluss von individuellen kognitiven Eigenschaften des Betrachters auf die Wahrnehmung von Visualisierungen (Toker et al., 2013).



(a)



(b)

Abbildung 2.6: Zwei Beispiele für videobasierte Eye-Tracking-Systeme: Der stationäre Tobii T60 XL von Tobii Technology AB (a) und die am Kopf getragenen SMI Eye Tracking Glasses 2 von SensoMotoric Instruments (b).

2.4.1 Ablauf eines Eye-Tracking-Experiments

Ein Eye-Tracking-Experiment im Labor hat den gleichen Aufbau wie ein kontrolliertes Experiment zur Erfassung von Durchführungszeiten und Fehlerraten, siehe dazu auch (Field und Hole, 2003). Hinzu kommen allerdings Schritte während der Durchführung des Experiments, die für eine möglichst genaue Aufzeichnung der Augenbewegungen notwendig sind.

Um ein signifikantes Ergebnis zu erhalten, ist es wichtig, vor der Erstellung des Experiments Hypothesen zu formulieren, die im Experiment bewiesen werden sollen. Hypothesen können zum Beispiel über eine bestimmte Aufmerksamkeitsverteilung in der Visualisierung oder zu einem bestimmten Leseverhalten der einzelnen graphischen Elemente aufgestellt werden. Basierend auf den formulierten Hypothesen werden geeignete Stimuli erstellt, die den Probanden im Experiment gezeigt werden. Zusätzlich werden Aufgabenstellungen formuliert, mit denen möglichst eindeutig die Hypothesen bewiesen oder widerlegt werden können. Wichtig bei der Erstellung des Experiments ist es, den Einfluss von Störvariablen zu minimieren, um ein möglichst aussagekräftiges Ergebnis zu erhalten. Der Einfluss der Störvariablen sowie der technische Aufbau des Experiments kann durch das Durchführen von einer oder mehrerer Pilotstudien getestet werden. Die Auswertung der Pilotstudie erlaubt es dann, die Anzahl der gezeigten Stimuli an die für das Experiment zur Verfügung stehende Zeit anzupassen und die Stimuli oder die Aufgabenstellung nochmals zu optimieren.

In dieser Arbeit wurde für alle Experimente das stationäre Eye-Tracking-System Tobii T60 XL verwendet. Die Experimente folgten immer demselben Ablauf:

1. Nachdem der Proband begrüßt wurde, füllt er einen Fragebogen zu seinen demographischen Daten aus. Er gibt zusätzlich an, ob er Vorkenntnisse im Bereich der im Experiment durchgeführten Aufgaben besitzt. Das Ziel dieses Fragebogens ist es, möglichst viele Störvariablen im Experiment abzufragen, um deren Einfluss bei der Auswertung minimieren zu können.
2. Anschließend führt der Proband einen Sehtest durch. In dieser Arbeit wurden dazu eine Sehprobentafel nach Snellen für die Sehschärfe und mehrere Ishihara-Farbtafeln (Ishihara, 1917) für den Test des Farbsehvermögen eingesetzt.
3. Dann erhält der Proband eine kurze Einführung in die Aufgabe. Diese Einführung sollte immer schriftlich erfolgen. Damit wird sichergestellt, dass alle Probanden mit dem gleichen Wissens-

stand in das Experiment starten. Bevor das eigentliche Experiment beginnt, bekommt der Proband die Möglichkeit, eventuell vorhandene Fragen zur Aufgabenstellung an den Versuchsleiter zu richten.

4. Bevor die Augenbewegungen des Probanden aufgezeichnet werden, muss das Eye-Tracking-System kalibriert werden. In dieser Arbeit wurde dazu die im Eye-Tracking-Gerät integrierte Kalibrierung mit neun Punkten eingesetzt. Die Kalibrierung kann während des Experiments mehrmals durchgeführt werden, um eine hohe Präzision der Augenpositionen des Probanden auf dem Stimulus sicherzustellen. Eine Kopfstütze kann zusätzlich eingesetzt werden, um die Genauigkeit bei der Aufnahme der Augenbewegungen zu erhöhen.
5. Dann beginnt die eigentliche Präsentation der Stimuli:
 - a) Bevor ein Stimulus gezeigt wird, wird in der Mitte des Bildschirms ein schwarzes Kreuz auf einem gleichmäßigen, hellen Hintergrund gezeigt. Der Proband wird gebeten, dieses Kreuz zu fokussieren. Damit soll sichergestellt werden, dass alle Probanden von der gleichen Stelle auf dem Stimulus ihre visuelle Suche zur Lösung der gegebenen Aufgabe beginnen. In diesem Schritt werden die Blickpunkte des Probanden aufgezeichnet, um später die Fokussierung des Kreuzes für jeden Probanden überprüfen zu können.
 - b) Sobald der Proband das Kreuz fokussiert hat, drückt er eine Taste auf der Computertastatur und der eigentliche Stimulus wird gezeigt.
 - c) Der Proband betrachtet den Stimulus und führt die gegebene Aufgabe durch.
 - d) Wenn der Proband sicher ist, dass er die gegebene Aufgabe beantworten kann, drückt er abermals eine Taste auf der Computertastatur. Der Stimulus verschwindet und das schwarze Kreuz wird wieder gezeigt. Dann gibt er mündlich die Antwort auf die gestellte Aufgabe. Die Zeiten zwischen beiden Tastatureingaben wird gespeichert, sowie die Antwort notiert. Während der Durchführung der Aufgabe werden die Blickpunkte des Probanden durch das Eye-Tracking-System aufgezeichnet.
 - e) Die Schritte a) bis d) werden solange durchgeführt bis alle Stimuli des Experiments gezeigt worden sind.

6. Falls mehrere Aufgabenblöcke im Experiment durchgeführt werden sollen, erhält der Proband die Möglichkeit, eine Pause zu machen, um sich zu entspannen. Zwischen den einzelnen Blöcken kann der Proband zusätzliche Fragebögen, wie beispielsweise den NASA Task-Load-Index (Hart und Staveland, 1988), ausfüllen. Dann beginnt der nächste Aufgabenblock mit dem gleichen Ablauf wie im vierten bzw. fünften Schritt beschrieben.
7. Wenn notwendig, füllt der Proband abschließend einen finalen Fragebogen aus.

2.4.2 Eye-Tracking-Datentypen

In der Literatur wurden eine Vielzahl von Datentypen und Metriken, die aus diesen abgeleitet werden, für die Analyse von Eye-Tracking-Experimenten vorgestellt (Jacob und Karn, 2003; Poole und Ball, 2005; Holmqvist et al., 2011). Für die Analyse von kognitiven und perceptiven Prozessen werden in dieser Arbeit die folgenden Metriken verwendet (Abbildung 2.7):

- *Blickpunkt*: Dieser gibt die vom Eye-Tracking-Gerät gemessene Position auf einem Stimulus an, die ein Proband zu diesem Zeitpunkt mit seinen Augen betrachtet. Die Aufnahmezeit des Eye-Tracking-Geräts gibt an, wie viele Blickpunkte pro Sekunde maximal aufgezeichnet wurden.
- *Fixation*: Mehrere Blickpunkte werden über ihre räumlichen und zeitlichen Eigenschaften zu einer Fixation zusammengefasst. Dazu existieren verschiedene Algorithmen (Salvucci und Goldberg, 2000; Holmqvist et al., 2011). Klassischerweise werden meist Blickpunkte in einem Radius von 20 bis 50 Pixel auf dem Bildschirm und in einem Zeitfenster von 200 *ms* bis 300 *ms* zu einer Fixation zusammengefasst. Fixationen werden durch einen Fixationsindex beschrieben. Dieser startet beispielsweise bei Beginn des Experiments bei 1 und zählt dann bis zum Ende des Experiments für jede berechnete Fixation weiter. Eine Fixation besitzt eine *Fixationsdauer*, die meist in Millisekunden angegeben wird, sowie eine eindeutige Position auf dem Stimulus in X-, Y- und wenn nötig Z-Koordinaten.

Teile dieses Abschnitts basieren auf:

Blascheck, Tanja; Kurzhals, Kuno; Raschke, Michael; Burch, Michael; Weiskopf, Daniel; Ertl, Thomas: State-of-the-Art of Visualization for Eye Tracking Data. In: *EuroVis STAR*, S. 63-82, 2014.

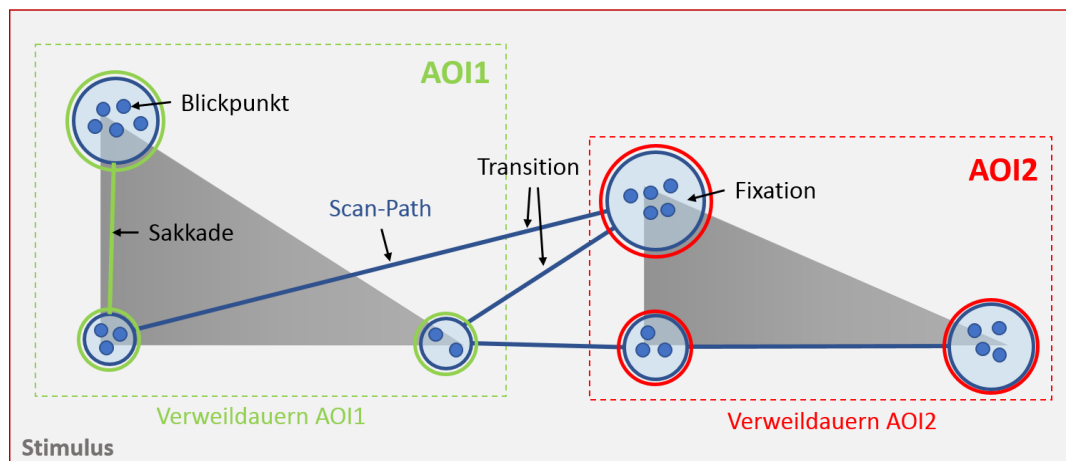


Abbildung 2.7: Die wichtigsten Datentypen und Metriken für die Analyse von Eye-Tracking-Experimenten sind *Blickpunkte*, *Fixationen*, *Sakkaden*, *Scan-Paths*, *Areas-of-Interest (AOIs)*, *Verweildauern* und *Transitionen*. Abbildung nach Blascheck et al. 2014.

- **Sakkade:** Schnelle Augenbewegungen von einer Position auf dem Stimulus zu einer anderen Position werden als Sakkaden bezeichnet. Diese haben in der Regel eine Zeitdauer von 30 ms bis 80 ms . Während einer Sakkade ist die Wahrnehmung von visuellen Reizen unterdrückt. Sakkaden werden typischerweise durch ihre Amplitude (Entfernung von einer Fixation zu der folgenden auf dem Stimulus), der Sakkadendauer in Millisekunden und der Sakkadengeschwindigkeit in Grad pro Sekunden beschrieben.
- **Scan-Path:** Beschreibt eine Sequenz von Fixationen und Sakkaden. Ein Scan-Path besteht aus mindestens einer Fixation, gefolgt von einer Sakkade mit einer abschließenden Fixation. Ein Scan-Path gibt an, welche Bereiche ein Proband in welcher Reihenfolge betrachtet hat. Scan-Paths können bei einer geeigneten Form durch eine konvexe Hülle umschlossen werden, die angibt, welchen Gesamtbereich der Proband durch sein Blickverhalten auf einem Stimulus abgedeckt hat (Goldberg und Kotval, 1999). Scan-Paths werden über ihre Länge in Pixeln oder ihrer zeitlichen Dauer in Millisekunden charakterisiert.
- **Area-of-Interest (AOI) / Region-of-Interest (ROI):** Damit werden wichtige Bereiche auf dem Stimulus, die für die Untersuchung einer Hypothese näher betrachtet werden, bezeichnet. AOIs können entweder vor der Analyse definiert werden oder auf Basis

der gemessenen Eye-Tracking-Daten. Ist letzteres der Fall, können die AOIs entweder automatisch erzeugt (Privitera und Stark, 2000; Santella und DeCarlo, 2004) oder manuell mit Hilfe eines AOI-Editors definiert werden. Meist werden AOIs aufgrund der semantischen Eigenschaften der graphischen Elemente in Visualisierungen definiert. Die Annotierung von Gebieten auf dem Stimulus mit AOIs wird in der Regel durch die Definition von Bounding Boxen umgesetzt. AOI-basierte Metriken wie Verweildauer, Transitionsanzahl und Transitionsfrequenz können dann direkt in der AOI-Visualisierung durch Einfärbung der Bounding Boxen dargestellt werden (Rodrigues et al., 2012). Speziell mit AOIs können die folgenden drei Metriken verwendet werden:

- *Verweildauer*: Ist die Summe der Fixationszeiten eines Probanden innerhalb einer AOI.
- *Transitionsanzahl*: Ist die Anzahl der Wechsel der Aufmerksamkeit von einer AOI zu einer anderen.
- *Transitionsfrequenz*: Ist die Anzahl der Aufmerksamkeitswechsel zwischen zwei AOIs pro Zeitintervall.

Nicht direkt ableitbar aus den Eye-Tracking-Daten, aber für die Durchführung eines Eye-Tracking-Experiments ebenso wichtig, ist der *Stimulus*. Ein Stimulus kann entweder ein statisches Bild sein, wie beispielsweise ein Balkendiagramm, eine Videosequenz oder ein Programm, mit dem der Proband interagieren soll.

2.4.3 Visualisierung von Eye-Tracking-Daten

Eye-Tracking-Experimente können auf zwei Arten analysiert werden:

1. Eine hypothesenbasierte Analyse erlaubt es, „harte“ Signifikanzwerte zu berechnen, mit denen die vor dem Experiment aufgestellten Hypothesen bewiesen oder widerlegt werden können (Holmqvist et al., 2011).
2. Eine explorative „weiche“ Analyse erlaubt es, zunächst ergebnisoffen die Augenbewegungen zu analysieren, um dann anschließend Ergebnisse in einem zweiten Schritt, wenn nötig mit einem hypothesenbasierten Experiment, nochmals genauer zu beweisen.

Teile dieses Abschnitts wurden veröffentlicht in:

Blascheck, Tanja; Kurzhals, Kuno; Raschke, Michael; Burch, Michael; Weiskopf, Daniel; Ertl, Thomas: State-of-the-Art of Visualization for Eye Tracking Data. In: *EuroVis STAR*, S. 63-82, 2014.

Im Falle einer hypothesenbasierten Analyse zielt der Einsatz von Visualisierungstechniken auf das Gewinnen eines Überblicks über die aufgenommenen Daten, zum Unterstreichen der statistischen Ergebnisse oder zur Illustration des Blickverhaltens ab.

Bei einer explorativen Analyse spielen Visualisierungstechniken eine zentrale Rolle bei der Untersuchung der Daten. Eine besondere Herausforderung bei der visuellen Analyse von Augenbewegungen ist die große Anzahl an Fixationen, die idealerweise zusammen mit ihrer semantischen Information, gut erkennbar, graphisch dargestellt werden müssen. So sind in einem Benutzerexperiment mit 30 Probanden, drei Aufgaben und 30 Stimuli Augenbewegungen in Form von $30 \times 3 \times 30 = 2.700$ Scan-Paths auszuwerten. Diese 2.700 Scan-Paths bestehen selbst wieder aus einigen wenigen, bis hundert Fixationen.

In diesem Abschnitt werden Visualisierungstechniken für Eye-Tracking-Daten, wie in (Blascheck et al., 2014) vorgestellt, in folgender Weise unterteilt:

- Zeitachsenbasierte Visualisierungen von Fixationspunkten (fixationsbasiert)
- Aufmerksamkeitskarten (fixationsbasiert)
- Scan-Path-Visualisierungen (fixationsbasiert)
- Space-Time-Cube-Visualisierungen (fixationsbasiert)
- Zeitachsenbasierte AOI-Visualisierungen (AOI-basiert)
- Visualisierung der AOI-Transitionen (AOI-basiert)

Zeitachsenbasierte Visualisierungen von Fixationspunkten

Zeitachsen werden verwendet, um zeitliche Charakteristika von Daten darzustellen. So wird auf einer Achse eines Koordinatensystems die Zeit abgetragen, auf der anderen Achse die zu untersuchende Größe (Abbildung 2.8a). Meist wird ein zweidimensionales Kartesisches Koordinatensystem dazu verwendet. So können die zeitlichen Veränderungen von Fixationspositionen aufgeteilt, in ihre X- und Y-Koordinaten, in zwei Visualisierungen dargestellt werden (Goldberg und Helfman, 2010a). Diese Darstellungsform kann sowohl für statische als auch interaktive oder dynamische Stimuli und für mehrere Probanden verwendet werden (Grindinger et al., 2010). Die Aufteilung des Verlaufs der Fixationsposition in X- und Y-Koordinaten führt zu keiner Linienkreuzung bei der Darstellung des Scan-Paths eines Probanden oder zu einer reduzierten Anzahl von Linienkreuzungen bei der Darstellung der Scan-Paths von mehreren Probanden. Jedoch wird dieser

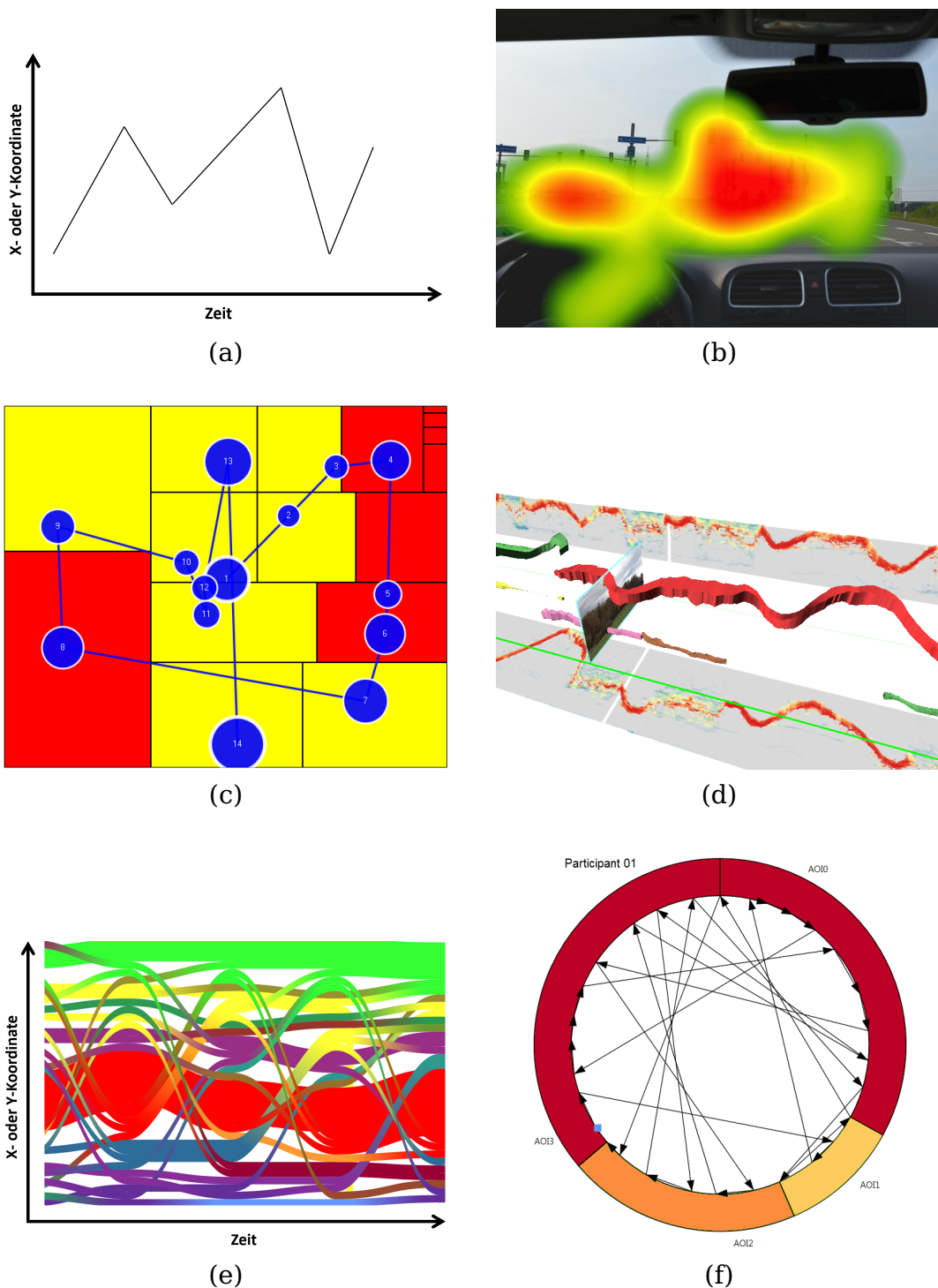


Abbildung 2.8: Sechs verschiedene Visualisierungstechniken für Eye-Tracking-Daten: a) Zeitachsenbasierte Visualisierungen von Fixationspunkten, b) Heat-Map, c) Scan-Path-Visualisierung, d) Space-Time-Cube-Visualisierung (mit freundlicher Unterstützung von Kuno Kurzhals), AOI-Rivers (mit freundlicher Unterstützung von Michael Burch), Circular-Heat-Map-Visualisierung (mit freundlicher Unterstützung von Tanja Blascheck).

Vorteil dadurch erkauft, dass durch die Aufteilung der Scan-Paths in zwei Diagramme ein höherer mentaler Aufwand nötig ist, um räumliche Eigenschaften der Scan-Paths zu erkennen oder miteinander zu vergleichen. Einen anderen Ansatz verfolgen die Gaze Stripes. Diese stellen Vorschaubilder an der Fixationsposition aus dem Stimulus dar und ordnen diese in eigenen Zeitachsen für jeden Probanden an (Kurzthals et al., 2015).

Aufmerksamkeitskarten

Aufmerksamkeitskarten zeigen direkt auf dem Stimulus die Positionen der einzelnen Fixationen der Probanden an. Dies kann in Form einer einfachen Auftragung der Fixationen als Punkte erfolgen (Mackworth und Mackworth, 1958). Die Kombination von Fixationen mehrerer Probanden und einem Stimulus wird als Bee Swarm bezeichnet (Tobii Technology AB, 2008). Bee Swarms zeigen Fixationen aller Probanden für einen Zeitpunkt. Wenn die Fixationen auf dem Stimulus über die Zeit aufsummiert werden, spricht man von *Aufmerksamkeitskarten* (Attention-Maps, Fixation-Map oder Heat-Map). In dieser Arbeit wird der Name *Heat-Map* verwendet (Abbildung 2.8b). Das Hauptziel von Heat-Maps ist es, einen schnellen Überblick über die Aufmerksamkeitsverteilung von Probanden auf einem Stimulus zu erhalten. Heat-Maps werden oft verwendet, um AOIs zu definieren.

Neben verschiedenen Techniken wie Heat-Maps (Špakov und Miniotas, 2007; Blignaut, 2010), können mit Aufmerksamkeitskarten auch absolute oder relative Fixationsdauern visualisiert werden oder sie zeigen die prozentualen Anteile der Aufmerksamkeit von Probanden auf dem Stimulus (Bojko, 2009). Aufmerksamkeitskarten können auf unterschiedliche Weise dargestellt werden, z. B. als Luminanzkarten (Velichkovsky und Hansen, 1996), dreidimensionale Landkarten (Latimer, 1988; Wooding, 2002), zweidimensionale topographische Karten mit Konturlinien (Goldstein et al., 2007; Dorr et al., 2010b) oder mit Color-Coding (Bojko, 2009; Duchowski et al., 2012). Zusätzlich können Filter angewendet werden, um die weniger fokussierten Gebiete weniger stark hervorzuheben (Dorr et al., 2010a). Motion-Compensated-Heat-Maps verwenden Informationen des optischen Flusses (Horn und Schunck, 1981) der Fixationen, um eine optimale Darstellung der Aufmerksamkeitsverteilung mehrerer Probanden zu erzeugen (Kurzthals und Weiskopf, 2013).

Scan-Path-Visualisierungen

Zeitliche und räumliche Eigenschaften der Eye-Tracking-Daten können dadurch dargestellt werden, dass Fixationen mit Sakkaden verbunden wer-

den. Noton und Stark führten dazu den Begriff des „Scan-Paths“ ein (Noton und Stark, 1971a,b). In einer typischen Scan-Path-Visualisierung wird jede Fixation durch einen Kreis dargestellt, dessen Radius die Fixationsdauer an dieser Position widerspiegelt. Sakkaden zwischen diesen Fixationen werden als Linien dargestellt (Scinto et al., 1986). Ein einfacher Scan-Path zeigt nur die aufgenommenen Sakkaden ohne Fixationskreise (Yarbus, 1967). Meist werden zur Analyse der Augenbewegungen Scan-Path-Visualisierungen, wie die in Abbildung 2.8c) dargestellte, verwendet.

Ein großer Nachteil dieser Visualisierungstechnik ist es, dass sie schnell unübersichtlich wird, wenn Scan-Paths von mehreren Probanden dargestellt werden. Aber auch, wenn sich die Scan-Paths ähneln und dadurch die Fixationskreise und Sakkadenlinien oft übereinander liegen, wird es schwierig, interessante Muster im Verlauf der Scan-Paths zu erkennen. Um den Effekt des *Visual Clutters* (Rosenholtz et al., 2005) zu reduzieren, wurden in der Literatur mehrere Ansätze vorgestellt. Eine Möglichkeit besteht darin, Scan-Path-Verläufe zu mitteln oder zusammenzufassen (Hembrooke et al., 2006; Hurter et al., 2013; Chen et al., 2013), die verwendete Farbdichte der Scan-Paths zu verringern (Goldberg und Helfman, 2010a) oder die räumliche Information der Scan-Paths in ihre einzelnen Komponenten aufzuteilen und einzeln darzustellen (Croft et al., 2006; Burch et al., 2014c). Zusätzlich wurde vorgeschlagen, auch nur einen bestimmten zeitlichen Ausschnitt des Scan-Paths zu visualisieren (Weibel et al., 2012).

Space-Time-Cube-Visualisierungen

Space-Time-Cube-Visualisierungen zeigen die raumzeitlichen Eigenschaften von Scan-Paths in einer dreidimensionalen Darstellung (Abbildung 2.8d). Diese Visualisierungstechnik kann sowohl für statische (Li et al., 2010) als auch für dynamische Stimuli eingesetzt werden (Duchowski und McCormick, 1998; Kurzhals und Weiskopf, 2013). Mit einer Space-Time-Cube-Visualisierung können Cluster an Fixationen visualisiert werden. Ein Vorteil dieser Visualisierungstechnik ist es, dass sie auf einen Blick eine direkte Erkennung von interessanten Mustern über mehrere Probanden hinweg erlaubt.

Zeitachsenbasierte AOI-Visualisierungen

Neben der Gruppe der punktbasierten Visualisierungstechniken für Eye-Tracking-Daten können AOIs verwendet werden, um die Analyse auf Bereiche im Stimulus zu konzentrieren. Dazu wurden in den letzten Jahren eine Vielzahl von Arbeiten zur zeitleistenbasierten AOI-Visualisierung vorgestellt. Diese Visualisierungstechnik stellt ähnlich wie die punktbasierten

Visualisierungstechniken zeitliche Charakteristika der Scan-Paths bezüglich der fokussierten AOIs dar. Die Zeit wird dabei entlang einer Achse in der Visualisierung abgetragen. Die andere Achse repräsentiert entweder die fokussierten AOIs oder die Probanden.

Zeitleistenbasierte AOI-Visualisierungen können nur einen Probanden abbilden (Räihä et al., 2005; Crowe und Narayanan, 2000; Holsanova, 2001). Dazu wird auf der einen Achse in der Visualisierung die Zeit abgetragen, auf der anderen der Scan-Path über die einzelnen AOIs. Ein Anwendungsgebiet dieser Visualisierungstechnik ist die Untersuchung des Leseverhaltens von Texten, bei der Rekursionen des Lesers erkannt werden (Špakov, 2008; Beymer und Russell, 2005). Die Fixationsverteilung auf den AOIs kann zusätzlich farblich hervorgehoben werden (Kim et al., 2012). Zeitleistenbasierte AOI-Visualisierungen, ähnlich der Scan-Path-Visualisierung, werden schnell unübersichtlich bei einer größeren Probandenanzahl. Als Lösung können rechtwinklige Linienverläufe verwendet und jeder Proband in einer eigenen Visualisierung dargestellt werden (Itoh et al., 2000). Ein weiteres Visualisierungskonzept stellt die einzelnen AOIs als Rechtecke übereinander angeordnet dar und die Zeitabschnitte werden eingefärbt, in denen einzelne Probanden eine AOI fokussiert haben (Weibel et al., 2012). Eine weitere Möglichkeit besteht darin, die Fixationen auf den AOIs über mehrere Probanden hinweg zu mitteln und in derselben Visualisierung darzustellen. Die Visualisierung zeigt dann entweder eine Zeitleiste für jede einzelne AOI (Kurzhal et al., 2014) oder es wird der „Fluß der Aufmerksamkeit“ der gemittelten Probanden in Form eines AOI-Rivers abgebildet (Burch et al., 2013) (Abbildung 2.8e). AOI Transition Trees stellen darüber hinaus Frequenzen der Transitionen und Muster in den Transitionen dar (Kurzhal und Weiskopf, 2015). Zusätzlich zeigt der Transition Tree kleine Vorschaubilder der AOIs dar.

Eine Visualisierungstechnik, in der anstatt der fokussierten AOIs die Probanden auf der zweiten Achse abgetragen werden, sind Scarf-Plots (Richardson und Dale, 2005). In Scarf-Plots wird jeder AOI eine gleiche Farbe zugeordnet (Ristovski et al., 2013). Scarf-Plots können auch mit Zeitleisten kombiniert werden (Kurzhal et al., 2014). In die Klasse der zeitachsenbasierten AOI-Visualisierungen kann ebenfalls die in dieser Arbeit vorgestellte parallele Scan-Path-Visualisierung (Kapitel 4) eingeordnet werden. Das Konzept dieser Visualisierungstechnik wurde auch in (Bednarik und Tukiainen, 2008) verwendet, um das Blickverhalten von Programmierern zu analysieren. Die in dieser Arbeit vorgestellten Varianten der Darstellung von Scan-Paths über AOIs hinweg gehen jedoch deutlich über die Möglichkeiten der in der Literatur bereits verwendeten Darstellung hinaus.

Visualisierung der AOI-Transitionen

Die klassische Visualisierungstechnik für Transitionen zwischen AOIs für einen Probanden ist die Transitionsmatrix ohne (Goldberg und Kotval, 1999) und mit eingefärbten Einträgen (Li et al., 2012). Die Transitionen von mehreren Probanden können durch das Zusammenführen der Einzelmatrizen erfolgen (Goldberg und Helfman, 2010b). Eine weitere Möglichkeit, Transitionen darzustellen, ist die Verwendung eines graphenbasierten Visualisierungskonzepts. Abbildung 2.8f) zeigt eine Circular-Heat-Map-Visualisierung. Die AOIs werden dazu als Kreissegmente angeordnet. Die Größe des Kreissegments zeigt verschiedene Charakteristika der AOIs wie Fixationsanzahl oder -dauer und kann zusätzlich eingefärbt werden. Die Dicke der einzelnen Transitionen gibt ihre Anzahl wieder. Mit einer baumstrukturierten Visualisierungstechnik (West et al., 2006; Tsang et al., 2010; Ristovski et al., 2013) können verschiedene Suchstrategien identifiziert werden. Diese Visualisierungstechnik hat jedoch den Nachteil, dass die Bäume bei langen Experimenten groß werden.

Zusammenfassung

In diesem Abschnitt wurden verschiedene Visualisierungstechniken für die Analyse von Eye-Tracking-Daten für verschiedene Analysefragen vorgestellt. Jedoch besitzen die meisten von ihnen den Nachteil, dass sie zu einer unübersichtlichen Darstellung der Blickdaten tendieren, wenn eine größere Anzahl von Probanden visualisiert werden soll. Eine mögliche Lösung besteht darin, mit visueller Analytik interessante Muster in den Eye-Tracking-Daten erkennen zu können. Andrienko et al. (Andrienko et al., 2012) diskutieren dazu, welche Techniken aus dem Gebiet der visuellen Analytik sich besonders gut auf Eye-Tracking-Daten anwenden lassen. Auch zeigen die Arbeiten von Kurzhals und Weiskopf (Kurzhals und Weiskopf, 2013), Kurzhals et al. (Kurzhals et al., 2014) und Blascheck et al. (Blascheck et al., 2015b) wie die Analyse von Eye-Tracking-Daten durch die visuelle Analytik profitieren kann. Die in Abschnitt 4.2 vorgestellte Technik zum Vergleich von Scan-Paths folgt ebenfalls dem Ansatz der visuellen Analytik, geht aber noch einen Schritt weiter (Abschnitt 5), und erlaubt es dem Analysten, zusätzlich verschiedenartige kognitive Prozesse direkt aus der Visualisierung abzulesen.

2.5 Vergleich von Scan-Paths

Ein Vergleich von Scan-Paths ist notwendig, um Gruppen von Probanden mit einem ähnlichen Blickverhalten identifizieren zu können. Aus ähnlichen

Scan-Paths können dann allgemeingültige Lesestrategien für Visualisierungen definiert werden. Allerdings wurden in der Literatur bisher nur wenige automatische Methoden vorgestellt, um Scan-Paths von mehreren Probanden miteinander vergleichen zu können. Beispiele sind die von Privitera und Stark (Privitera und Stark, 2000) und Duchowski et al. (Duchowski et al., 2010) vorgestellten Methoden des String-Editings, die Berechnung eines Entropiemaßes von Fixationsverteilungen (Krejtz et al., 2014) oder das Zusammenfassen von Fixationen in Cluster (Kurzahls et al., 2014).

Wie Duchowski et al. betonen, besteht eine wichtige Notwendigkeit für Techniken, die einen quantitativen Vergleich von Scan-Path-Metriken erlauben (Duchowski et al., 2010) („There is a pressing need for quantitative scanpath comparison metrics.“). In dieser Arbeit wird für den Vergleich von Scan-Paths das Modell des *idealen Scan-Paths* verwendet (Abschnitt 2.5.1), um eine optimale Lesestrategie einer Visualisierung für eine gegebene Aufgabe zu definieren. Für den Vergleich dieses idealen Scan-Paths mit aufgenommenen Augenbewegungen stellt diese Arbeit in Abschnitt 4.2 eine Methode vor, die auf der Levenshtein-Distanz (Abschnitt 2.5.2) basiert.

2.5.1 Idealer Scan-Path

Die Interpretation von Scan-Paths ist eine komplexe Aufgabe. Die Scan-Paths einzelner Probanden können stark variieren, obwohl die gleiche Aufgabe mit demselben Stimulus durchgeführt haben. Um die Interpretation der aufgenommenen Scan-Paths zu unterstützen, stellen Conversy et al. ein deskriptives Modell für das visuelle Durchsuchen von Visualisierungen nach aufgabenrelevanten Informationen vor (Conversy et al., 2010, 2011). Die Autoren beschreiben dazu eine Methode, um einen *idealen Scan-Path* zu definieren, mit dem eine Visualisierung gelesen wird. Dieser ideale Scan-Path umfasst einzelne visuelle Aufgaben, die nacheinander mit der Visualisierung ausgeführt werden müssen, um eine bestimmte Information in einer Visualisierung zu erfassen (siehe dazu auch Abschnitt 6.2.1). Beim Lesen des Wertes der Kategorie „H“ in einem Balkendiagramm könnten diese einzelnen visuellen Aufgaben zum Beispiel sein (Abbildung 2.9):

1. „Finde die passende Kategorie (Fixation 1 bis 4)“
2. „Lese den Wert des Balkens ab (Fixation 5)“
3. „Suche das obere Ende des Balkens (Fixation 6)“

Diese drei visuellen Aufgaben werden durch einen idealen Scan-Path abgedeckt, der die ideale Lesestrategie für diesen Stimulus und für diese Aufgabe modelliert. Dieser ideale Scan-Path kann mit aufgenommenen Augenbewegungen der Probanden verglichen werden, um abzuleiten, in

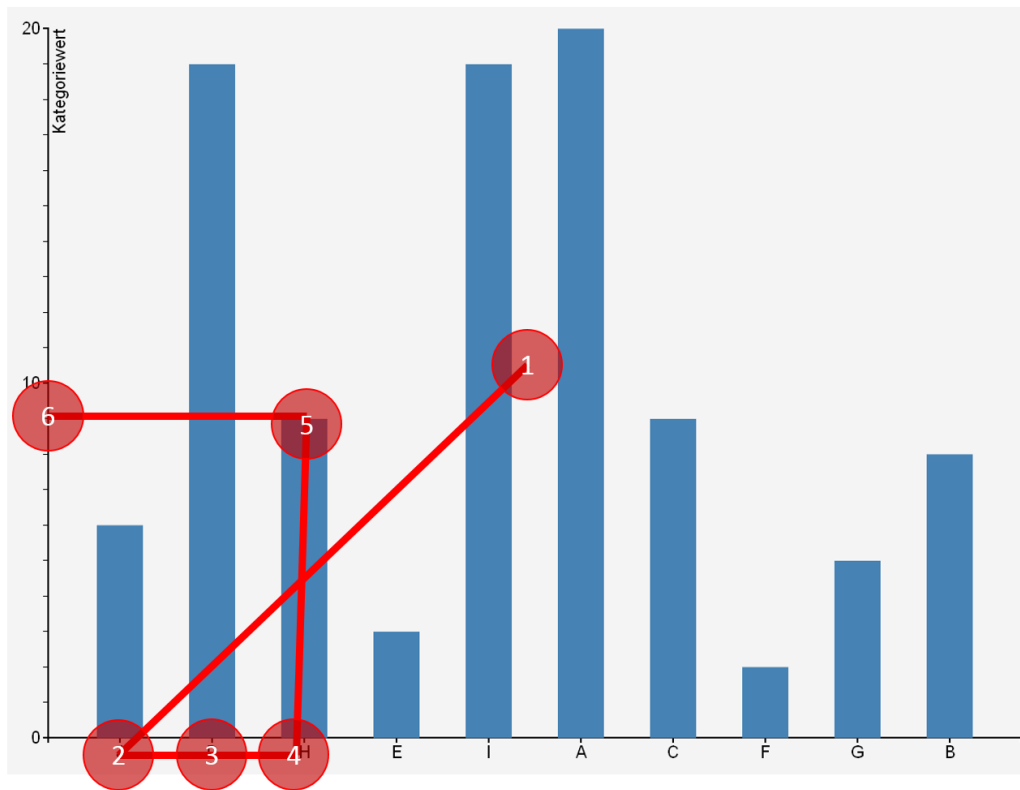


Abbildung 2.9: Ein *idealer Scan-Path* beschreibt die optimale Lesestrategie für eine gegebene Visualisierung und Aufgabe. Hier ist ein idealer Scan-Path für die Aufgabe „Lesen Sie den Wert der Kategorie 'H' ab.“ dargestellt.

welchen Bereichen der Visualisierungen die Probanden Schwierigkeiten hatten, die dargestellte Information schnell zu erfassen oder zu verstehen.

2.5.2 Levenshtein-Distanz

Die Berechnung der Levenshtein-Distanz zwischen zwei Zeichenketten (Levenshtein, 1965) kann im Eye-Tracking für den Vergleich von Scan-Paths herangezogen werden. Dazu werden AOIs mit Zeichen aus einem Alphabet identifiziert. Trifft eine Fixation eines Scan-Paths in eine dieser AOIs, wird dieser Fixation das der AOI zugeordnete Zeichen zugewiesen. Ein Scan-Path über die AOIs führt damit zu einer Kette von Zeichen. Mehrere Arbeiten demonstrieren, wie man durch die Berechnung der Ähnlichkeit dieser Zeichenketten Rückschlüsse auf die Ähnlichkeit der Scan-Paths ziehen kann (Privitera und Stark, 2000; Duchowski et al., 2010; Kurzhals et al., 2014). Diese Methode wird ebenfalls in dieser Arbeit verwendet und soll deshalb hier kurz beschrieben werden.

		S	O	N	N	T	A	G
	0	2	4	6	8	10	12	14
S	2	0	2	4	6	8	10	12
A	4	2	1	3	5	7	9	11
M	6	4	3	2	4	6	8	10
S	8	6	5	4	3	5	7	9
T	10	8	7	6	5	3	5	7
A	12	10	9	8	7	5	3	5
G	14	12	11	10	9	7	5	3

Tabelle 2.1: Kostentabelle für die Levenshtein-Distanz zwischen der Zeichenkette „SAMSTAG“ und „SONNTAG“. Die Levenshtein-Distanz kann im Element unten rechts in der Matrix abgelesen werden und beträgt in diesem Beispiel 3.

Zur Bestimmung der Ähnlichkeit von Zeichenketten werden die Kosten berechnet, um eine Zeichenkette mit Hilfe der Operationen „Ersetzen“, „Einfügen“ und „Löschen“ von einzelnen Zeichen in eine andere Zeichenkette zu transformieren.

Als Eingabe für die Berechnung der Levenshtein-Distanz liegen die zwei Zeichenketten A und B vor. Die Zeichenkette A hat n Zeichen. Die Zeichenkette B hat die Länge m Zeichen. Mit $A[i]$ wird im Folgenden das Zeichen an der Position i in der Zeichenkette A bezeichnet, mit $B[j]$ die Position des Zeichens an der Stelle j in der Zeichenkette B . Für die Operationen zum Ändern der Zeichenketten A an der Stelle i können unterschiedliche Kosten vorgegeben werden:

- Kosten $[c_i^i]$ für das Einfügen (insert) eines Zeichens
- Kosten $[c_i^r]$ für das Ersetzen (replace) eines Zeichens
- Kosten $[c_i^d]$ für das Löschen (delete) eines Zeichens

Oft wird $[c_i^i] = [c_i^d] = 2$ und $[c_i^r] = 1$ gesetzt. Zur Bestimmung der Ähnlichkeit wird eine Tabelle mit $(n + 1) \times (m + 1)$ Einträgen, wie in Tabelle 2.1 gezeigt, angelegt. Die Einträge der ersten Zeile in der Tabelle werden nach der Formel

$$T(x, 0) = c^d \cdot x$$

initialisiert. Damit wird jedes einzelne Zeichen der Zeichenkette gelöscht. Die erste Spalte der Tabelle wird nach

$$T(0, y) = c^i \cdot y$$

initialisiert. Damit werden alle einzelnen Zeichen eingefügt, um die Zielzeichenkette zu erhalten. Die restlichen Einträge von Tabelle 2.1 werden mit der Vorschrift

$$T(x, y) = \min \begin{cases} T(x-1, y) + c^i \\ T(x-1, y-1), & \text{falls } A(x) = B(y) \\ T(x-1, y-1) + c^r, & \text{sonst} \\ T(x, y-1) + c^d \end{cases}$$

erzeugt. Der Eintrag an der Position (n, m) enthält dann die Kosten, um z. B. die Zeichenkette „SONNTAG“ in die Zeichenkette „SAMSTAG“ zu transformieren. In diesem Beispiel betragen die Kosten 3.

2.6 Kognition

Homo sapiens ist Lateinisch und bedeutet der verstehende, vernünftige Mensch. Diese Beschreibung gibt im Allgemeinen zum Ausdruck, dass es die Intelligenz ist, mit der wir uns von anderen Lebewesen unterscheiden. Die Kognitionswissenschaft sieht diese Intelligenz als einen Prozess der Informationsverarbeitung (Görz et al., 2014; Anderson, 2001). Der zentrale Vorgang bei der Verarbeitung von Information über unsere Umwelt ist die Wahrnehmung der Umgebung (zur visuellen Wahrnehmung siehe Abschnitt 2.2) und ihre Abbildung in das Gedächtnis, sowie ihr Vergleich mit bereits dort gespeicherten Information (siehe Abschnitt 2.7.2). Das Ziel eines kognitiven Systems ist es dabei, auf Basis von vorhandenem Wissen zukünftige Ereignisse vorauszusagen. Diese Vorhersage erlaubt es einem kognitiven System, angepasst an Vorgänge in seiner Umwelt handeln zu können. Dazu verknüpft ein kognitives System Muster in seinem Gedächtnis mit Objekten und Abläufen in seiner Umgebung. Dieser Umweltbezug erlaubt es einem kognitiven System, auf seine Umwelt rückkoppelnd zu reagieren. Da die Kognitionswissenschaft, als eine interdisziplinäre Wissenschaft, die Methoden ihrer Mutterdisziplinen Philosophie und Logik, theoretische Informatik, Psychologie, Neurowissenschaft und Künstliche Intelligenz verbindet, kann sie auf formalen Methoden zur Beschreibung des Denkens und auf experimentellen Methoden zur Validierung der Ergebnisse aufbauen. Mit Hilfe von berechenbaren Modellen können damit Denkprozesse simuliert und mit empirisch ermittelten Ergebnissen verglichen werden. Durch die enge Verbindung der Kognitionswissenschaft zur

Informatik führte dieser Ansatz zu einer weiten Anwendbarkeit der Modelle und Methoden in den Gebieten der Künstlichen Intelligenz, Mensch-Computer-Interaktion, Sprachverarbeitung und Robotik. In dieser Arbeit wird der kognitionswissenschaftliche Ansatz verwendet, um die menschliche Wahrnehmung von Visualisierungen sowie die dabei ablaufenden informationsverarbeitenden Prozesse bei der Durchführung von Aufgaben mit Visualisierungen zu untersuchen.

In diesem Unterkapitel wird zunächst kurz die Anatomie des menschlichen Gehirns beschrieben (Abschnitt 2.6.1) gefolgt von einer Darstellung der kognitionswissenschaftlichen Arbeitsweise (Abschnitt 2.6.2).

2.6.1 Das menschliche Gehirn

Über den Sehnerv erreichen die elektrischen Signale der Augen (Abschnitt 2.2.1) das Gehirn. Das Gehirn des Menschen besitzt ein ungefähres Gewicht von $1,5\text{ kg}$ und hat ein Volumen von ca. 1130 cm^3 bei einer Frau und von ca. 1260 cm^3 bei einem Mann (Cosgrove et al., 2007). Die wichtigsten zellulären Bestandteil des Gehirns sind seine Nervenzellen, Gliazellen als Stützgewebe der Nervenzellen und ein weitverzweigtes System von Blutadern. Mit seinen ca. 85 Milliarden Nervenzellen (Azevedo et al., 2009) und etwa 100 Billionen Synapsen (10^{14}) ist das menschliche Gehirn die komplexeste Struktur, die im Universum bekannt ist.

Die oberste Schicht des Gehirns, der *cerebrale Cortex* oder auch *Neocortex*, ist der jüngste, während der Evolution entstandene Teil, des Gehirns. Flach ausgebreitet, bedeckt der Neocortex eine Fläche von ca. 2500 cm^2 (Toro et al., 2008). Die Dicke des Neocortex selbst beträgt ca. 2 bis 5 mm. Dies entspricht in etwa der Fläche einer großen Stoffserviette und der Dicke von sechs Visitenkarten. Die beiden Hemisphären des Gehirns teilen sich auf seiner Oberfläche in vier Bereiche auf, die durch große Furchen, die sogenannten Fissuren, geteilt werden: den Frontal- oder Stirnlappen, den Parietal- oder Scheitellappen, den Occipital- oder Hinterhauptslappen und den Temporal- oder Schläfenlappen. Alle vier Bereiche spielen eine Rolle, wenn visuelle Reize verarbeitet werden. Der Hinterhauptslappen beinhaltet das Sehzentrum zur Verarbeitung der vom Sehnerv eintreffenden Reize. Der vordere Teil des Frontallappens, der präfrontale Cortex, ist bei Planungsprozessen beteiligt. Die räumliche Vorstellung findet im Scheitellappen statt. An der Objekterkennung ist der Schläfenlappen beteiligt.

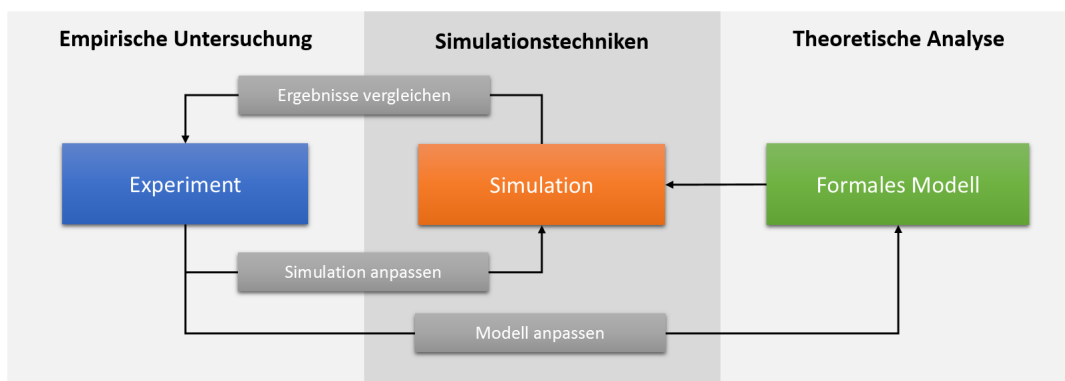


Abbildung 2.10: Die Kognitionswissenschaft verbindet die theoretische Analyse kognitiver Funktionsbereiche mit empirischen Untersuchungsmethoden und der Simulation kognitiver Prozesse.

2.6.2 Methode der kognitionswissenschaftlichen Arbeitsweise

Durch ihren interdisziplinären Charakter vereint die Kognitionswissenschaft Arbeitsweisen aus der theoretischen Analyse kognitiver Funktionsbereiche und Aufgabenstellungen mit dem Ziel der Formalisierung (aus Philosophie, Logik und Linguistik), mit empirischen Untersuchungsmethoden (aus der Psychologie und den Neurowissenschaften) und der Modellierung kognitiver Prozesse mit Hilfe der Informatik. Für die Entwicklung neuer Methoden und Techniken zur Evaluation von Visualisierungen verwendet diese Arbeit speziell aus diesen Gebieten:

1. *Benutzerexperimente* aus der Psychologie zur empirischen Erfassung von Modellparametern und zur Überprüfung von kognitiven Modellen und Simulationsergebnissen (Abschnitt 2.3).
2. *Ontologien* aus der Philosophie für die Repräsentation von Wissen über Visualisierungen auf drei Abstraktionsebenen (Abschnitt 2.7.2)
3. *Modellierungstechniken und Simulationsprogramme* aus der Informatik zur Konstruktion von virtuellen, kognitiven Maschinen (Abschnitt 2.7).

2.7 Modellierung und Simulation der visuellen Suche

Die Auswertung von Eye-Tracking-Experimenten basiert meist auf der Annahme, dass das, was der Mensch visuell wahrnimmt, auch maßgeblich beeinflusst, was er in diesem Moment denkt. Diese Annahme wird die Augen-Gedanken-Hypothese genannt (Abschnitt 2.7.1). Um die visuelle Suche modellieren und simulieren zu können, muss Wissen über Visualisierung-

gen und den darin vorhandenen graphischen Elementen in einer maschinell verarbeitbaren Form beschrieben werden (Abschnitt 2.7.2). Dies erfolgt in dieser Arbeit durch die Abbildung von graphischen Elementen und ihrer Relation zueinander in Form einer Ontologie. Leseregeln beschreiben dann, wie die einzelnen graphischen Elemente in einer Visualisierung in welcher Reihenfolge gelesen werden müssen, um eine bestimmte Information daraus ermitteln zu können. In der Mensch-Computer-Interaktion stellt die Modellierung der Interaktion, mit einem System mit Hilfe von einzelnen Operationen, einen sehr erfolgreichen Ansatz dar. Dieser wird im Abschnitt 2.7.4 kurz beschrieben, da später darauf aufgebaut wird. Zur Simulation der visuellen Suche können neben Perzeptions- und Kognitionsmodellen (Abschnitt 2.7.5) auch Simulationswerkzeuge, wie das in der Kognitionswissenschaft weit verbreitete ACT-R, eingesetzt werden (Abschnitt 2.7.6). Das Unterkapitel schließt mit einer Vorstellung von in der Literatur vorgestellten ACT-R-Prototypen zur Simulation der visuellen Suche (Abschnitt 2.7.7), sowie mit einer kurzen Beschreibung der Benutzersimulation CogTool aus der Mensch-Computer-Interaktionsforschung (Abschnitt 2.7.8). Diese Benutzersimulation diente in dieser Arbeit als Inspiration für die Entwicklung eines ähnlichen Werkzeugs für die Visualisierungsforschung, das in Abschnitt 6.1 vorgestellt wird.

2.7.1 Augen-Gedanken-Hypothese (Eye-Mind-Hypothesis)

Eine grundsätzliche Frage bei der Analyse von Augenbewegungen ist es, wie stark Augenbewegungen mentale Prozesse widerspiegeln. Just und Carpenter formulierten dazu die *Augen-Gedanken-Hypothese (Eye-Mind-Hypothesis)* (Just und Carpenter, 1976, 1980). In Textleseexperimenten konnten sie zeigen, dass die Fixationsdauer auf Wörtern in einem Text davon abhängt, ob diese als bekannt vorausgesetzt werden konnten, ob eine bestimmte Information an dieser Textstelle in die Gesamtbedeutung des Textes integriert werden musste, oder ob ein Satz nach seiner logischen Aussage hin verarbeitet wurde (Just und Carpenter, 1980). Daraus kann die Schlussfolgerung gezogen werden, dass die Fixationsdauer mit der kognitiven Last korreliert, die notwendig ist, um eine visuell wahrgenommene Information zu verarbeiten. Übertragen auf Fragen in der Visualisierungsforschung kann, wenn diese Hypothese als wahr angenommen wird¹, davon ausgegangen werden, dass ein Betrachter die in einem Bereich einer Visualisierung dargestellte Information kognitiv verarbeitet, solange er diesen Bereich fokussiert. Zusätzlich zeigt die Reihenfolge der

¹In dieser Arbeit wird diese Hypothese als wahr angenommen. Pathologische Fälle und Unaufmerksamkeit beim Betrachten einer Visualisierung werden dabei ausgeschlossen.

Fixationen, welche Informationen er in einer Visualisierung nacheinander wahrgenommen und kognitiv verarbeitet hat (Koerner, 2011).

In der Eye-Tracking-Forschung wird diese Hypothese oft als Grundlage verwendet, um experimentelle Ergebnisse interpretieren zu können. Jedoch zeigen eine Vielzahl von Experimenten, dass die Augen-Gedanken-Hypothese nur mit Vorsicht bei der Analyse von Eye-Tracking-Daten angewendet werden sollte. Beispielsweise konnten Triesch et al. zeigen, dass in manchen Situationen die Probanden direkt auf ein aufgabenrelevantes Objekt geschaut hatten, jedoch keine Verknüpfung im Arbeitsgedächtnis dazu gefunden werden konnte (Triesch et al., 2003). Holsanova et al. zeigten, dass regelmäßige Sakkaden zwischen Text und Bildern nicht nur beweisen können, dass der Proband beide Modalitäten miteinander verknüpft, sondern auch, dass er Schwierigkeiten bei dieser Verknüpfung hat (Holsanova et al., 2008). In einem Fahrtexperiment konnten sich Probanden an nicht-fokussierte Objekte erinnern (Underwood et al., 2003).

2.7.2 Mentale Repräsentation und Modellierung von visuellem Wissen

Um die räumlichen Daten aus einem Eye-Tracking-Experiment mit der Bedeutung der, an einer bestimmten Position auf dem Stimulus dargestellten, Information verbinden zu können, wird in dieser Arbeit ein Wissensmodell über Visualisierungskonzepte und den darin gezeigten graphischen Elementen verwendet. Um dieses Modell aufbauen zu können, muss man sich zunächst mit der Repräsentation von Wissen im Gehirn und der Struktur des visuellen Arbeitsgedächtnisses beschäftigen. Wichtige Eigenschaften des visuellen Arbeitsgedächtnisses werden dann in Form einer Implementierung umgesetzt. Dazu wird in dieser Arbeit hauptsächlich die Implementierung eines Wissensgraphen mit Techniken des Semantic Webs verwendet.

Aufbau des Gedächtnis

Mit dem Gedächtnis können Erinnerungen aufgenommen und abgerufen, sowie Informationen, Begebenheiten und Ereignisse aus der Vergangenheit geordnet werden. Damit kann aus vergangenen Ereignissen eine Orientierung für das gegenwärtige und zukünftige Handeln gewonnen werden.

Das Gedächtnis wird heute nicht als monolithisch, sondern als aus unterschiedlichen Systemen aufgebaut, angesehen. So können getrennte Systeme für das episodische Gedächtnis und das prozedurale Gedächtnis gefunden werden (Tulving, 1972). Das episodische Gedächtnis gehört wiederum zum *deklarativen Gedächtnis*, das Fakten oder Ereignisse speichert. Diese Fakten können zum Beispiel die Teile eines Fahrrads und ihre semantischen Relationen zueinander sein. Das *prozedurale Gedächtnis*

beinhaltet Fertigkeiten. In diesem Teil des Gedächtnisses ist hinterlegt, wie die einzelnen Teile eines Fahrrads sinnvoll miteinander in Kombination bedient werden müssen, um damit fahren zu können.

Zusätzlich kann das Gedächtnis in ein Kurzzeit- und ein Langzeitgedächtnis unterteilt werden (Ebbinghaus, 1885). Dabei speichert das Kurzzeitgedächtnis vier bis sieben Informationseinheiten für wenige Sekunden bis Minuten (Cowan, 2001; Miller, 1956). Baddeley führte eine dritte Art des Gedächtnisses ein, das *Arbeitsgedächtnis* (Baddeley, 1981). Dieses umfasst das Kurzzeitgedächtnis, kann aber auch aus dem Langzeitgedächtnis abgerufene Information beinhalten. Als visueller Kurzzeitspeicher beinhaltet der ikonische Puffer das aktuell auf der Retina abgebildete Bild für einige wenige hundert Millisekunden (Sperling, 1960; Neisser, 1979).

Grundlagen für den Ansatz der Wissensrepräsentation in dieser Arbeit

Für diese Arbeit wird ein semantisches Netzwerk als Modell zur Repräsentation von Wissen verwendet. Quillian untersuchte, wie Menschen Informationen über verschiedene Vogelarten im Gedächtnis speichern. Als Gedächtnismodell schlug er dafür ein Netzwerk wie in Abbildung 2.11 dargestellt vor (Quillian, 1966). Diese Hierarchie zeigt, dass ein Kanarienvogel ein Vogel ist und ein Vogel ein Tier ist. Dazu können *IS-A-Verbindungen* zwischen den einzelnen Kategorien im Netzwerk gezogen werden, die aussagen: „Ein Kanarienvogel *ist* ein Vogel.“ oder „Ein Fisch *ist* ein Tier.“. Die einzelnen Kategorien geben dabei die jeweiligen Eigenschaften an. Collins und Quillian konnten auch zeigen, dass die Zeit, die ein Proband benötigt, um einen Sachverhalt als richtig oder falsch zu erkennen, mit diesem Netzwerk erklärt werden kann (Collins und Quillian, 1969). So benötigen Probanden weniger Zeit für eine Aussage, ob der Satz „Kanarienvögel können singen.“ wahr ist, als für den Satz „Kanarienvögel haben eine Haut.“. Dieser Unterschied lässt sich durch die unterschiedlichen Abstände der einzelnen Kategorien im Netzwerk erklären. Jedoch stellt dieses Netzwerk eine starke Vereinfachung der konzeptuellen Wissensrepräsentation im Gedächtnis dar und wurde aus diesem Grund seitdem erweitert. Es zeigte sich, dass neben dem Abstand von Konzepten im semantischen Netzwerk auch die Verbindungsstärke der einzelnen Tatsachen einen Einfluss auf die Abrufzeit von mental gespeichertem Wissen hat.

Durch die Nähe der Kognitionswissenschaft zur Künstlichen Intelligenz kann mit dem Ansatz, Wissen in Form eines Wissensnetzes zu speichern, die Brücke zur Informatik geschlagen werden. Semantische Wissensmodelle können mit Hilfe von Semantic-Web-Technologie implementiert und mit der objektorientierten Programmierung in Simulationsprogrammen abgebildet

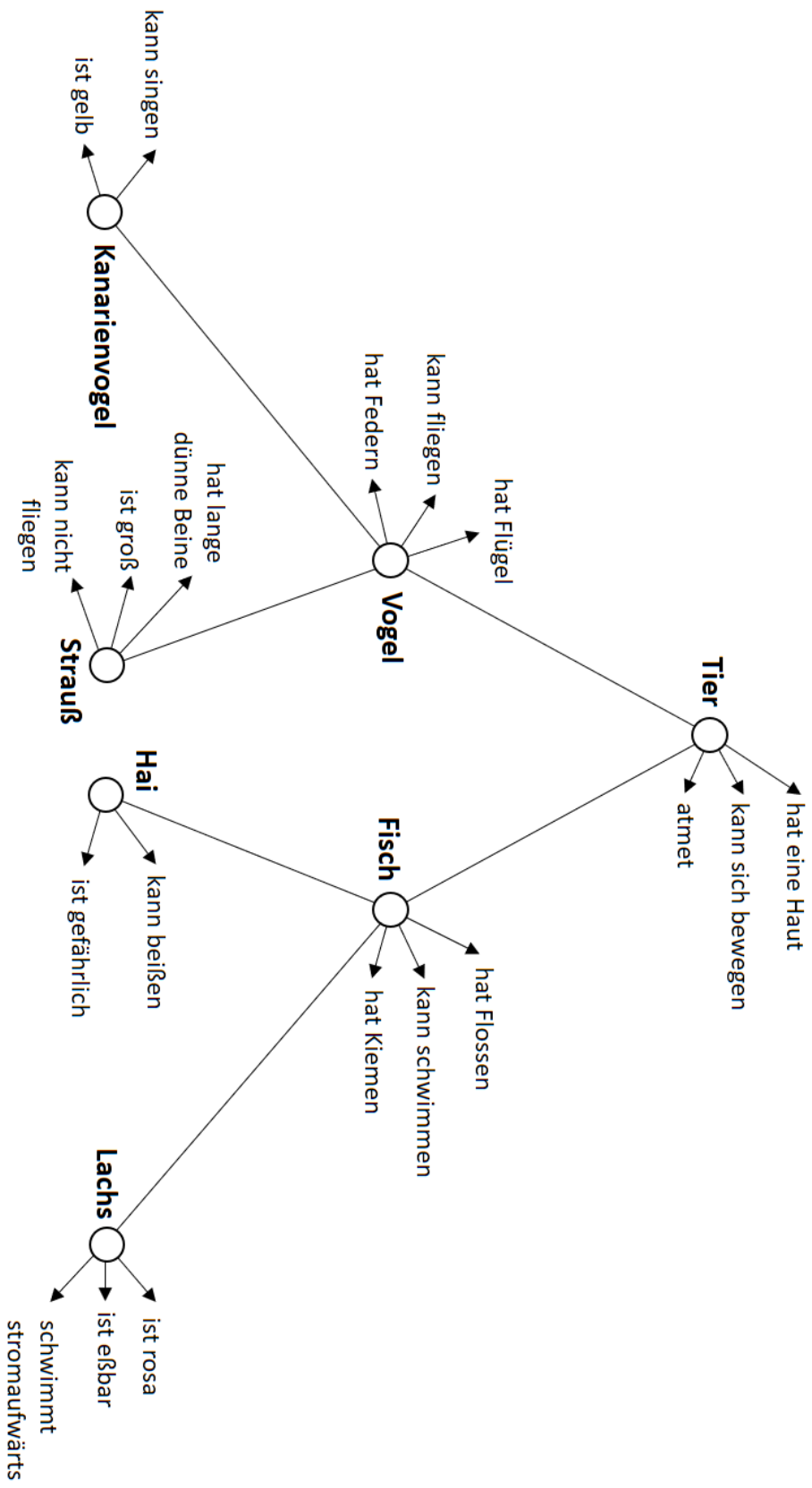


Abbildung 2.11: Wissensnetz nach (Collins und Quillian, 1969) zur Kategorisierung von Tieren.

werden. Damit wird es möglich, ein Wissensmodell mit Hilfe einer expliziten formalen Spezifikation einer Konzeptualisierung in Form einer *Ontologie* zu implementieren (Gruber, 1993).

In dieser Arbeit kommen die zwei Semantic-Web-Technologien *Resource Description Framework (RDF)* und *Web Ontology Language (OWL)* zum Einsatz. Ein RDF-Modell beschreibt einen gerichteten Graphen, bestehend aus Knoten und gerichteten Kanten. Die Kanten in diesem RDF-Graph stehen für Aussagen über sogenannte Ressourcen, welche durch die Knoten repräsentiert werden. In einem RDF-Graph sind sowohl die Kanten als auch die Knoten mit eindeutigen *Unique-Resource-Identifiern (URIs)* gekennzeichnet. Die äußeren Knoten im RDF-Graph, von denen keine gerichteten Kanten ausgehen, werden als Literale bezeichnet. Sie sind nicht durch URIs gekennzeichnet und stellen Datenwerte als Zeichenketten dar. Ontologien werden im Semantic Web in der Web Ontology Language beschrieben. OWL kann sowohl zur Kennzeichnung von Ressourcen als Klassen, als auch zur Formulierung von Ausdrücken, ähnlich der Prädikatenlogik, verwendet werden. Damit können auf einer in OWL beschriebenen Ontologie Inferenz- und Integritätsregeln angewendet werden.

Bei der semantischen Annotation von Visualisierungen werden den graphischen Bausteinen eindeutige Ressourcen aus einer Ontologie zugewiesen. XML-basierte graphische Beschreibungssprachen, wie z.B. *Scalable Vector Graphics (SVG)*, eignen sich hierfür besonders gut, da in diesen graphische Bausteine voneinander abgegrenzt definiert werden und dadurch leicht semantisch annotiert werden können (Chang und Chuang, 2003).

Die primäre Motivation für die semantische Annotation von Visualisierungen ist häufig die Verbesserung der Suche nach graphischen Darstellungen (Ferecatu et al., 2008). Nur wenige Ansätze befassen sich mit der Frage, wie die semantischen Annotationen für ein besseres Verstehen der Visualisierungen verwendet werden können. Janecek und Pu zeigen beispielsweise, wie ausgehend von annotierten Bereichen Relationen zu verwandten Themen angezeigt und deren graphische Darstellungen exploriert werden können (Janecek und Pu, 2003). Andere Ansätze nutzen die Annotationen für eine semantische Filterung (Kalogerakis et al., 2006) oder für einen intelligenten Zoom (Patel et al., 2009). Arbeiten, die Semantic-Web-Technologie zur Modellierung von Wissen über graphische Elemente in Kognitionssimulation einsetzen, wurden bisher noch nicht vorgestellt.

2.7.3 Leseregeln von Visualisierungen

Für den Aufbau eines Modells zur Repräsentation des prozeduralen Wissens, wie Visualisierungen gelesen werden können, werden graphbasierte Leseregeln von Pinker (Pinker, 1990) verwendet. Pinker stellt ein Modell über

das mentale Wissen über eine Visualisierung in Form eines *Schemas* vor. Ein Schema definiert, welche grundsätzlichen Relationen für graphische Elemente in einer bestimmten Visualisierungstechnik gelten. Die Eigenschaften und Relationen der graphischen Elemente werden in Form eines gerichteten Graphen dargestellt. Abbildung 2.12 zeigt einen Ausschnitt eines Schemas für ein Balkendiagramm.

Ein Schema kann instanziiert werden, um ein konkret vorliegendes Diagramm genau zu beschreiben. Das Ergebnis ist ein *Diagramm-Modell* mit exakten Angaben zu den einzelnen graphischen Elementen. Graphische Elemente, die bei einzelnen Diagrammen unterschiedlich sind, wie zum Beispiel die Höhe von Balken in einem Säulendiagramm, werden dazu als Objekte von Klassen im Diagramm-Modell entsprechend ihrer Anzahl mehrfach instanziiert. Klassen sind in Pinkers Graphdarstellung mit einem Stern (*) gekennzeichnet. Message-Flags, die an Knoten und Kanten im Graph angehängt sind, beschreiben den Lesevorgang des Schemas. Ein vorliegendes Diagramm wird mit im Gedächtnis hinterlegten Schemata verglichen. Das Schema, das am besten passt, wird dann für die Durchführung einer visuellen Aufgabe verwendet. Mit Hilfe der Message-Flags können dann kognitive Prozesse beschrieben werden, die ausgeführt werden müssen, um eine visuelle Aufgabe mit der im Schema abgebildeten Visualisierung erfolgreich durchzuführen.

Aufbauend auf der Unterteilung von Pinkers Wissensmodell für Leseregeln von Visualisierungen in Schemata und Modellen wird in Abschnitt 5.2 ein dreistufiges Modell zur Repräsentation von Wissen über Visualisierungen vorgestellt, das zur Analyse (Abschnitt 5.4) und Simulation (Abschnitt 6.1) von Augenbewegungen verwendet wird.

2.7.4 Operatorenbasierte Ansätze in der Mensch-Computer-Interaktion

Eines der ersten Modelle, um Benutzeraufgaben und Interaktionsprozesse zu simulieren, wurde von Card et al. im Jahr 1980 mit dem GOMS Modell (**G**oals, **O**perators, **M**ethod, **S**election rules) vorgestellt (Card et al., 1980, 2000). Basierend auf dem GOMS sind zahlreiche weitere Modelle im Umfeld der Usability-Forschung entstanden. GOMS reduziert die Interaktion zwischen einem Benutzer und einem Computer auf elementare Aktionen. Diese können entweder physikalischer, perzeptiver oder kognitiver Natur sein. Das GOMS-Modell beschreibt den Interaktionsprozess bezüglich des Ziels (Goal), das die Benutzer mit ihrer Interaktion erreichen wollen. Eine Sequenz von Operatoren beschreibt dann die einzelnen Schritte, die durchgeführt werden müssen, um dieses Ziel zu erreichen. Wenn mehrere

Sequenzen der Operatoren zu einem gleichen Ziel führen, helfen Regeln, die beschreiben, wann ein Benutzer eine bestimmte Sequenz wählt. GOMS geht davon aus, dass der Benutzer des Systems als Experte bezeichnet werden kann. Empirische Studien haben gezeigt, dass der Ansatz und die Methodik des GOMS verwendet werden können, um verschiedene Strategien zum Lösen eines Problems mit einem System vergleichen zu können (siehe beispielsweise Gong und Elkerton (1990), Gray et al. (1992) und Gong und Kieras (1994)). Der große Vorteil von GOMS liegt in dem relativ geringen Aufwand für seine Anwendung. Wichtig bei der Anwendung von GOMS ist es, eine möglichst präzise Messung für jeden einzelnen Operator durchzuführen. Die Berechnung der Gesamtdurchführungszeit einer Methode erfolgt dann durch die Addition der Einzelzeiten der Operatoren.

Card et al. entwickelten die Idee des GOMS-Modells mit dem Keystroke-Level Model (*KLM*) weiter (Card et al., 2000). Das KLM konzentriert sich auf Interaktionsaufgaben mit Computersystemen. Die Interaktion wird im KLM durch folgende Operatoren beschrieben (in Klammern sind die jeweils von Card et al. ermittelten Zeiten angegeben):

- **K**: Tastendruck (zwischen 0,08 s bis 1,2 s)
- **P**: Die Maus auf ein bestimmtes Element auf dem Bildschirm bewegen (1,1 s)
- **B**: Maustaste drücken oder Maustaste loslassen (0,1 s)
- **H**: Hand von Maus zu Tastatur oder von Tastatur zu Maus bewegen (0,4 s)
- **M**: Gedankliche Vorbereitung des nächsten Schritts (1,2 s)
- **T(n) = n·K**: Text mit n Buchstaben über die Tastatur eingeben
- **W(t)**: Warten auf Reaktion des Computersystems in Sekunden
- **D(n0, l0)**: Eine gerade Linie $n0$ mit Länge $l0$ in *cm* mit der Maus zeichnen

In ihrer Arbeit präsentieren Card et al. umfangreiche Evaluationsexperimente der verwendeten zeitlichen Parameter. Viele Arbeiten in der Mensch-Computer-Interaktion haben erfolgreich das KLM in verschiedenen Anwendungsfeldern validiert (siehe beispielsweise (Haunold und Kuhn, 1993; Bälter, 2000; Teo und John, 2006)). Andere Arbeiten passten die Operatoren an oder fügten neue hinzu (für Mensch-Fahrzeug-Interaktion siehe (Pettitt et al., 2007; Schneegaß et al., 2011), für Interaktionen mit

dem Handy siehe (Holleis et al., 2007b; Li et al., 2010) und für haptische Benutzerschnittstellen (Holleis et al., 2007a)).

Der Erfolg des KLM in der Mensch-Computer-Interaktion führte zur Frage, ob der Ansatz des KLM auch für die Modellierung von visuellen Suchstrategien und zur Berechnung von Durchführungszeiten für visuelle Aufgaben verwendet werden kann. Dazu stellt diese Arbeit ein einfaches Modell zur Vorhersage von Durchführungszeiten von visuellen Aufgaben vor (Kapitel 6.2) und diskutiert die Ergebnisse aus zwei Experimenten.

2.7.5 Perzeptions- und Kognitionsmodelle für visuelle Suchstrategien

In der Literatur wurden verschiedene Modelle vorgestellt, um visuelle Suchstrategien vorherzusagen, Durchführungszeiten zu berechnen und den Prozess der Informationssuche in Visualisierungen zu beschreiben. In diesem Abschnitt wird ein chronologischer Überblick über die für diese Arbeit relevanten Veröffentlichungen aus den Gebieten der Visualisierungsforschung, Mensch-Computer-Interaktion, Perzeptionsforschung und Kognitionswissenschaft gegeben. Auf Arbeiten, die ACT-R zur Simulation der visuellen Suche verwenden, gehen die beiden folgenden Abschnitte ein.

Simkin und Hastie analysierten die Wahrnehmung von Diagrammen (Simkin und Hastie, 1987) basierend auf Arbeiten von Pinker (Pinker, 1990), Kosslyn (Kosslyn, 1986) und Ullman (Ullman, 1984). Sie führten dazu mehrere Experimente durch, die zeigten, dass die Fehlerrate für Diagramme am geringsten war, bei denen Probanden Längen und Abstände schätzen mussten. Die Fehlerrate war höher, wenn Probanden Winkel für das Lösen der Aufgaben schätzen mussten. Damit bestätigten Simkin und Hastie die Arbeit von Cleveland und McGill (Cleveland und McGill, 1984), welche die selben Ergebnisse zeigt. Jedoch gingen die Autoren noch einen Schritt weiter und führten für die Analyse der Lesevorgänge elementare Prozesse ein, um die Lesevorgänge im Detail analysieren zu können. Dieser Ansatz wird in Abschnitt 6.2 für die Unterteilung einer visuellen Suchstrategie in einzelne, sogenannte Visual Unit Tasks angewendet.

Lohse et al. stellten das kognitive Modell UCIE (**U**nderstanding **C**ognitive **I**nformation **E**ngineering) zur Simulation der visuellen Wahrnehmung vor, um Durchführungszeiten für gegebene Fragen zu berechnen, die mit Hilfe einer graphischen Anzeige beantwortet werden sollten (Lohse, 1993). Obwohl das Modell von Lohse et al. auf GOMS basiert, werden für die Berechnung der Durchführungszeiten zusätzliche Annahmen über die Reihenfolge der Fixationen, die Kapazität des Kurzzeitgedächtnisses und der Schwierigkeitsgrad, graphisch dargestellte Information auf einen Blick wahrzunehmen, berücksichtigt.

Wolfe stellte mit Guided Search 2.0 bis 4.0 Modelle der visuellen Wahrnehmung vor (Wolfe, 1994; Wolfe und Gancarz, 1997; Wolfe, 2007). Die Modelle unterscheiden zwischen einer präattentiven, massiv parallel arbeitenden Phase, in der grundlegende visuelle Charakteristika wie Farbe oder Bewegung verarbeitet werden, und höheren komplexeren Aufgaben, wie Lesen oder Objekterkennung, die beschränkt durch die kognitive Leistungsfähigkeit sind. Die Kernidee der Guided-Search-Theorie ist es, dass die Verwendung der kognitiven Ressourcen in der zweiten Phase durch die früher ablaufenden Prozesse der ersten Phase gesteuert werden.

Shah präsentiert ein Modell zu kognitiven und perzeptiven Prozessen für die visuelle Wahrnehmung und diskutiert das Verhältnis von Bottom-Up- und Top-Down-Prozessen der Wahrnehmung und kognitiven Verarbeitung (Shah, 1997). Ein weiteres Beispiel ist das visuelle Suchmodell „Active Vision“ von Halverson und Hornof (Halverson und Hornof, 2011) für Aufgaben der Mensch-Computer-Interaktion. Die Autoren präsentieren ein Modell, das auf dem Kognitionssimulationsframework EPIC (Meyer und Kieras, 1997) basiert, und formulieren vier Fragen der „Active Vision“, die sie in ihrer Arbeit beantworten.

Liu und Stasko diskutieren das Wechselspiel zwischen internen mentalen Modellen und externen Repräsentationen aus Sicht der Visualisierungsforschung und motivieren in ihrer Arbeit zu einem interdisziplinären Ansatz zwischen Kognitionswissenschaft und Informationsvisualisierung (Liu und Stasko, 2010). Die vorliegende Arbeit folgt diesem Aufruf und stellt in Kapitel 3 einen interdisziplinären Ansatz vor, auf dessen Basis neue Methoden und Techniken zur Evaluation von Visualisierungen entwickelt worden sind.

2.7.6 Das Kognitionssimulationsframework ACT-R

In dieser Arbeit wird für die Simulation von kognitiven Prozessen der visuellen Suche das weit verbreitete Kognitionssimulationsframework ACT-R (**A**daptive **C**ontrol of **T**hought – **R**ational) eingesetzt (Anderson, 2007; Anderson et al., 2004). ACT-R gehört zu der Gruppe der Produktionsregelsysteme und basiert auf der von Anderson entwickelten Theorien ACT (Adaptive Control of Thought) (Anderson, 1976), ACT* (Anderson, 1983b) und ACT-R (Adaptive Control of Thought – Rational) (Anderson und Lebiere, 1998). Das Kognitionssimulationsframework ACT-R (im Folgenden werden sowohl die Theorie „ACT-R“ von Anderson als auch das Kognitionssimulationsframework „ACT-R“ als „ACT-R“ bezeichnet) umfasst dabei symbolische und subsymbolische Verarbeitungsmechanismen.

Das in der Version 7.0 vorliegende Framework ACT-R² besteht aus einer

²Das ACT-R-Framework kann frei unter <http://act-r.psy.cmu.edu> heruntergeladen werden.

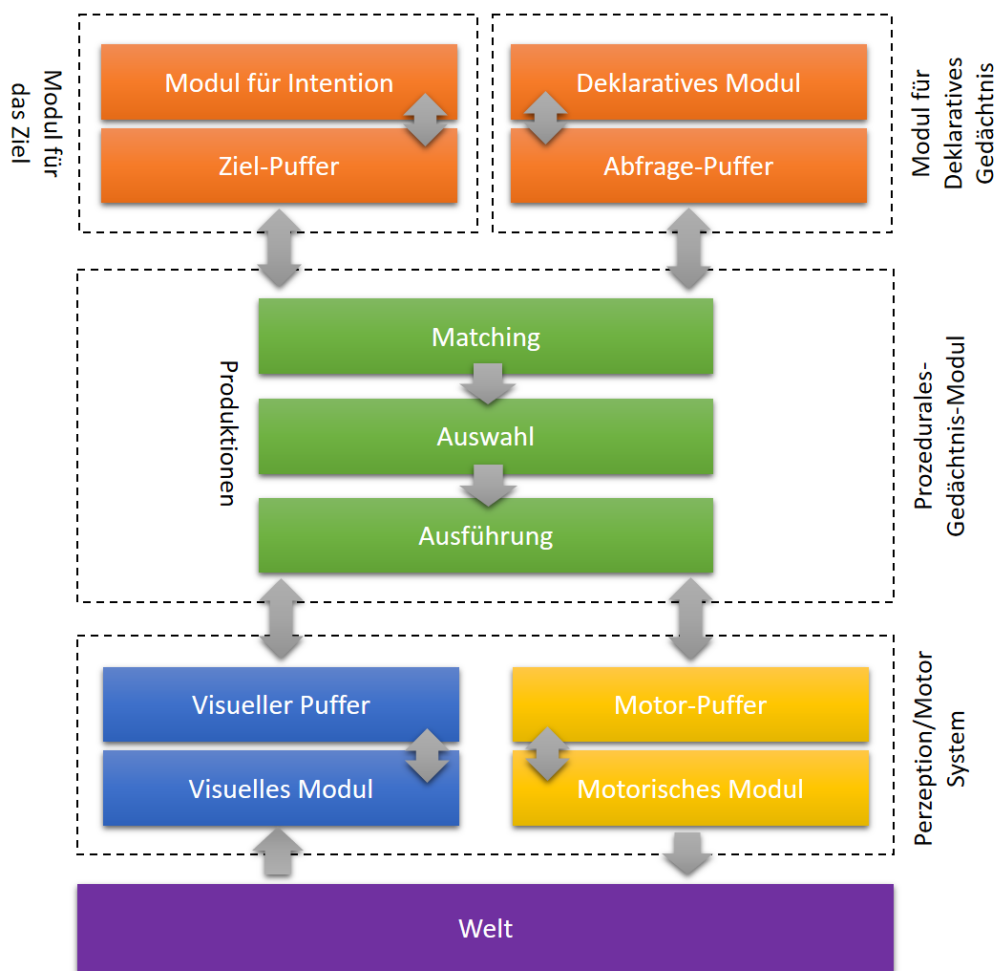


Abbildung 2.13: Das weit verbreitete Kognitionssimulationsframework ACT-R (**A**daptive **C**ontrol of **T**hought-**R**ational) simuliert kognitive Prozesse mit Hilfe von Produktionsregeln und einzelnen Modulen, die nach (Anderson et al., 2004) verschiedene Gehirnareale repräsentieren.

modularen Architektur (Abbildung 2.13). Die einzelnen Module sind dabei psychologisch fundiert und können mit Gehirnarealen identifiziert werden. Die Module umfassen grundlegende Funktionen für die Wahrnehmung (für diese Arbeit ist das visuelle Modul wichtig) sowie für die Ausführung von motorischen Reaktionen (Motorisches Modul). ACT-R besitzt die Fähigkeit, sowohl prozedurales als auch deklaratives Wissen abzubilden. Deklaratives Wissen wird in Form von sogenannten *Chunks* gespeichert, prozedurales Wissen in Form von *Produktionen*.

Unter dieser Webadresse finden sich auch eine umfangreiche Datenbank von ACT-R-relevanten Publikationen sowie ausführliche Anleitungen zur Entwicklung von ACT-R-Programmen.

Im Zentrum des Kognitionssimulationsframeworks steht das Produktionsregelsystem. Produktionsregeln sind ähnlich wie Wenn-Dann-Anweisungen aufgebaut. Eine Produktionsregel wird dann ausgeführt, wenn sich passende Chunks in den Puffern der Module befinden. Im Aktionsteil einer Produktionsregel wird definiert, wie Chunks entfernt, verändert, hinzugefügt oder angefragt werden sollen. ACT-R erlaubt es, jeder Produktionsregel Eigenschaften zu ihrer Nützlichkeit oder Informationen über die Kosten, die zur Ausführung notwendig sind, hinzuzufügen. Diese Werte können dann während des Ablaufs einer Simulation dynamisch angepasst werden. So kann zum Beispiel ein Lerneffekt simuliert werden, in dem die Kosten für die Ausführung einer Produktionsregel reduziert werden, wenn die Produktionsregel öfters während des Simulationslaufs ausgeführt wird. Zusätzlich verwendet ACT-R das Prinzip der Aktivationsausbreitung (Spreading Activation) zum Abfragen von Chunks aus dem deklarativen Gedächtnis (Anderson, 1983a). Die Zeit, die zum Abrufen von Chunks benötigt wird, die häufiger abgefragt werden, wird dadurch reduziert.

Da die einzelnen Module von ACT-R eine Entsprechung im menschlichen Gehirn haben, besitzt das Framework die Möglichkeit, die Intensität der Aktivität von Gehirnbereichen zu berechnen, um diese mit bildgebenden Verfahren, wie der funktionellen Magnet-Resonanz-Tomographie, zu vergleichen. Dies bietet, neben der direkten Überprüfung der Simulationsergebnisse, eine zusätzliche Möglichkeit kognitive Modelle zu validieren.

2.7.7 Simulation der visuellen Suche mit ACT-R

Peebles und Cheng stellen Ergebnisse der Simulation von verschiedenen Aufgaben mit Liniendiagrammen vor (Peebles und Cheng, 2002). Dazu verwendeten sie als Stimuli zwei verschiedene Arten von Liniendiagrammen, welche die gleiche Information beinhalten. Mit einer Simulation der kognitiven Prozesse ist es möglich, neben der reinen Wahrnehmung von graphischer Information, wie sie beispielsweise durch das UCIE-Modell von Lohse et al. modelliert wird, auch die für das Verständnis der Information notwendigen kognitiven Prozesse zu simulieren. Die Autoren entwickelten dazu ein auf ACT-R basierendes Programm. Neben der Darstellung der Diagramme im Stimulus beinhaltete das Programm auch die Frage als Text sowie Knöpfe, die zum Beantworten der Frage angeklickt werden mussten. Zur Simulation der Durchführungszeiten umfasst das ACT-R-Programm Produktionsregeln, um die gegebene Frage zu lesen, weitere Produktionsregeln, um die passenden Werte im Diagramm zu finden und zu lesen und weiterhin Produktionsregeln, mit dem die Mausbewegung auf dem Bildschirm simuliert werden kann, um die Antwortknöpfe zu drücken. Die

Analyse der Simulationsergebnisse zeigte, dass das ACT-R-Programm die aufgenommenen Augenbewegungsmuster reproduzieren kann.

Basierend auf seiner Arbeit von 2002 stellt Peebles eine Erweiterung seines ACT-R-Programms vor (Peebles, 2012). Diese erweiterte Version modelliert auch das grundsätzliche Wissen über den Aufbau eines Graphen. Aus diesem Wissen kann dann beispielsweise abgeleitet werden, wie einzelne Beschriftungen aufgrund ihrer Position im Graphen korrekt interpretiert werden müssen. Dazu muss das Wissensmodell auch die Relationen der einzelnen graphischen Elemente in einem Graphen abbilden. Das Modell muss dann aus diesen Relationen auch auf die räumlichen Positionen der graphischen Elemente Rückschlüsse ziehen können. Zusätzlich besitzt das Modell auch die Fähigkeit, Zusammenhänge, die während des Leseprozesses der Visualisierung gefunden werden, im deklarativen Wissensmodul von ACT-R abzuspeichern. In seiner Arbeit motiviert Peebles, dass der Hintergrund für die Vorstellung seiner Ergebnisse auch darin besteht, umfangreichere Modelle zur Simulation des Verständnisses für ein breiteres Spektrum an Klassen von Graphen zu entwickeln, wie zum Beispiel Säulendiagramme, Kuchendiagramme und andere Visualisierungstechniken. Diese Motivation wurde für die vorliegende Arbeit aufgegriffen.

2.7.8 CogTool

Abschließend soll in diesem Kapitel kurz die Software CogTool vorgestellt werden, die als Inspiration für die in Kapitel 6.1 vorgestellte Simulation der visuellen Suche diente. Die graphische Oberfläche von CogTool erlaubt es, relativ einfach Simulationen für die Mensch-Computer-Interaktion zu entwickeln (John et al., 2004). CogTool unterteilt dazu die Modellierung der Mensch-Computer-Interaktion in einen Ablaufplan (Storyboard) und in Aufgaben (Tasks), mit denen Interaktionsprozesse beschrieben werden. CogTool berechnet auf Basis dieses Modells mit Hilfe des KLMs und einer ACT-R-Kognitionssimulation die Zeit, die ein erfahrener Benutzer für die Durchführung der vorgegebenen Aufgabe benötigt. Die Ergebnisse zeigt CogTool in einer Zeitleistenvisualisierung (Abbildung 2.14) an. Anwendungsbeispiele von CogTool sind die Evaluation des Layouts von Webseiten (Teo und John, 2008), der Vergleich von verschiedenen Programmierwerkzeugen (Bellamy et al., 2010), die Simulation der Interaktion mit Mobiltelefonen (John und Suzuki, 2009) oder der Interaktion von sehbehinderten Benutzern mit Screen-Readern (Trewin et al., 2010).

Ein vergleichbares Werkzeug existiert in der Visualisierungsforschung bisher nicht. Ein möglicher Grund dafür könnte sein, dass der Aufwand zur Erstellung von Modellen zur visuellen Suche sowie der Aufbau eines geeigneten Wissensmodells als zu hoch und zu komplex wahrgenommen

2 Grundlagen und Stand der Technik

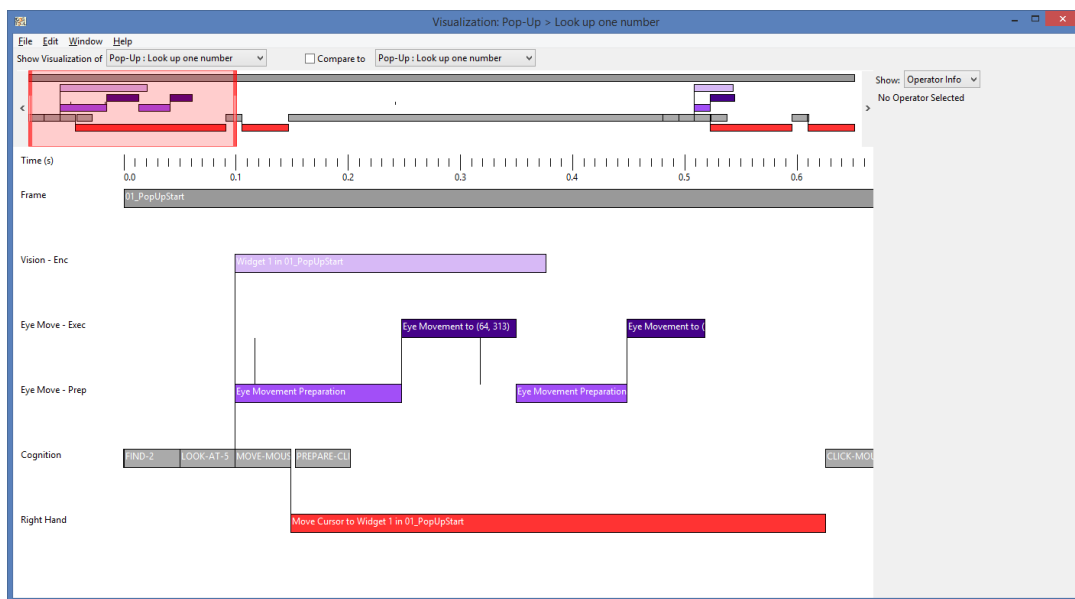


Abbildung 2.14: CogTool erlaubt es durch seine graphische Oberfläche mit relativ geringem Aufwand, Simulationen in der Mensch-Computer-Interaktion durchzuführen.

wird. Ein wesentliches Ziel der vorliegenden Arbeit war es, diesen Aufwand zu reduzieren. Damit soll es auch in der Visualisierungsforschung möglich sein, Simulationen der visuellen Suche einzusetzen, um den Aufwand für die Durchführung von Benutzerexperimenten reduzieren zu können und zusätzlich Einblick in kognitive Prozesse bei der Arbeit mit Visualisierungen zu erhalten.

Interdisziplinärer Ansatz für die Entwicklung von neuen Methoden und Techniken zur Evaluation von Visualisierungen

Für diese Arbeit wurde ein interdisziplinärer Ansatz entwickelt (Abbildung 3.1), um

- die experimentelle Methode des Eye-Trackings
- mit Modellen zur Wissenrepräsentation
- und Methoden und Techniken aus den Kognitionswissenschaften

zu verbinden.

Diese Verbindung erlaubte es, neue Methoden und Techniken für die Evaluation von Visualisierungen zu entwickeln. Im Folgenden wird eine kurze Übersicht über Ansätze, Modelle und Techniken aus dem Eye-Tracking, der Wissensmodellierung und den Kognitionswissenschaften gegeben, die für diese Arbeit verwendet wurden. Der interdisziplinäre Ansatz wird in Abschnitt 3.1 vorgestellt. Das Kapitel schließt mit einer Zusammenfassung des Beitrags der vorliegenden Arbeit zur Visualisierungsforschung (Abschnitt 3.2).

Eye-Tracking

Zu den beiden wesentlichen Gründen, sich im Rahmen der Visualisierungsforschung mit Eye-Tracking auseinanderzusetzen, gehört zum einen

Teile dieses Kapitels wurden veröffentlicht in:

Raschke, Michael; Blascheck, Tanja; Ertl, Thomas: Cognitive Ergonomics in Visualization. In: *Ebert, Achim and van der Veer, Gerrit C. and Domik, Gitta and Gershon, Nahum D. and Scheler, Inga: Building Bridges: HCI, Visualization, and Non-formal Modeling. Berlin/Heidelberg: Springer-Verlag, 2014.*

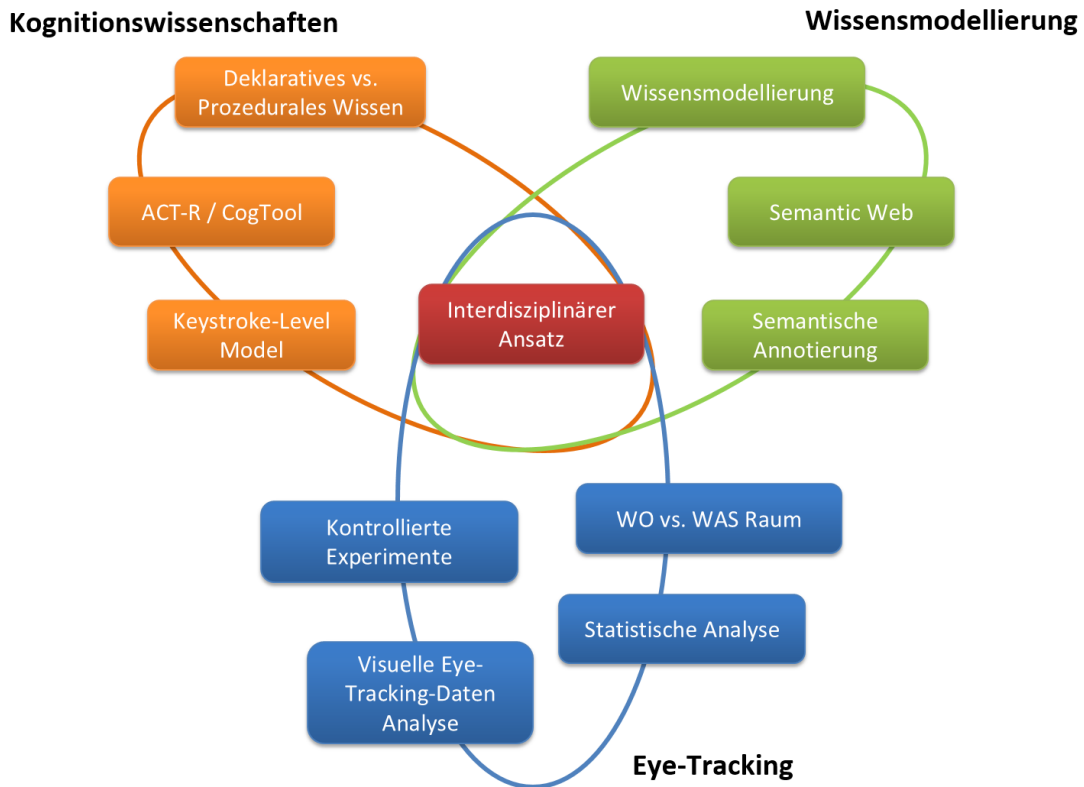


Abbildung 3.1: Auf Basis eines interdisziplinären Ansatzes wurden neue Methoden und Techniken für die Evaluation von Visualisierungen entwickelt. Dabei wurde die experimentelle Methode des Eye-Trackings mit Wissensmodellen und Methoden und Techniken aus den Kognitionswissenschaften verbunden.

die Möglichkeit, Visualisierungstechniken mit Hilfe von Eye-Tracking-Experimenten zu evaluieren, zum anderen besteht auch die Notwendigkeit für die Entwicklung von weiteren, neuen Visualisierungstechniken für die Analyse von Eye-Tracking-Experimenten.

Eine Vielzahl von Experimenten wurde in der Visualisierungsforschung durchgeführt, um die Wahrnehmung und das Verständnis von Visualisierungen zu untersuchen (Abschnitt 2.3 und 2.4). Auch im Rahmen dieser Arbeit wurden Eye-Tracking-Experimente durchgeführt. Dabei wurde ebenfalls das Paradigma des kontrollierten Experiments verwendet (Abschnitt 2.4.1 und 2.6.2), um signifikante, quantitative Ergebnisse zu erhalten.

Eine Recherche zu existierenden Visualisierungstechniken für Eye-Tracking-Daten ergab, dass die meisten Visualisierungstechniken der Eye-Tracking-Daten bei einer größeren Anzahl von Probanden zu einer unüber-

sichtlichen Darstellung, dem sogenannten Visual Clutter (Rosenholtz et al., 2005), führen (Abschnitt 2.4.3). Die Recherche zeigte auch, dass bisher noch kein Visualisierungskonzept für Eye-Tracking-Daten vorgestellt wurde, mit dem direkt in der Visualisierung Rückschlüsse auf kognitive und perzeptive Prozesse gezogen werden können. Dazu wird in dieser Arbeit die parallele Scan-Path-Visualisierungstechnik vorgestellt. Als Grundlage für die Entwicklung dieser Visualisierungstechnik wurde ein Visualisierungskonzept verwendet, das in den Arbeiten von Itoh et al. (Itoh et al., 2000) und Bednarik und Tukiainen (Bednarik und Tukiainen, 2008) verwendet wird. Jedoch besitzt die parallele Scan-Path-Visualisierungstechnik eine deutlich umfangreichere Funktionalität und erlaubt neben der Analyse kognitiver und perzeptiver Prozesse auch eine Identifikation von ähnlichen Scan-Paths. Dazu können, motiviert durch die Arbeiten von Andrienko et al. (Andrienko et al., 2012), mit Hilfe der visuellen Analytik ähnliche Scan-Paths gruppiert werden.

Modelle zur Wissensrepräsentation

Das Hauptaugenmerk bei der Entwicklung von Visualisierungen lag bisher meist auf der Frage, wie möglichst viele Daten in möglichst kurzer Zeit in einer möglichst hohen Auflösung dargestellt werden können. In den letzten Jahren gewann in der Visualisierungsforschung auch die Fragestellung immer mehr an Bedeutung, ob eine visuelle Darstellung von Informationen nützlich und leicht lesbar ist (Abschnitt 2.1.2). Um zur Beantwortung dieser Fragestellung beizutragen, verwendet diese Arbeit Ansätze und Techniken aus dem Forschungsgebiet der Wissensrepräsentation (Abschnitt 2.7.2). Das zentrale Werkzeug zur Modellierung von Wissen über Visualisierungen ist dabei die Speicherung des Wissens über Visualisierungskonzepte, graphische Elemente und deren Beziehungen zueinander in einer Ontologie (Abschnitt 2.7.2). Ein Vorteil der formalen Repräsentation von Wissen über Visualisierungen ist es, dass damit Computerprogrammen ermöglicht wird, automatisch Rückschlüsse auf Basis der Ontologie berechnen zu können, jedoch die verwendeten Modelle auch für Menschen lesbar sind. Zusätzlich wird durch die Verwendung einer Ontologie die Brücke zur Psychologie und den Kognitionswissenschaften geschlagen, da diese beiden Disziplinen ebenfalls graphenbasierte Konzepte zur Modellierung von Wissen und Wissensverarbeitungsprozessen einsetzen, wie beispielsweise zur formellen Beschreibung von Leseregeln von Visualisierungen (Abschnitt 2.7.3).

Durch die Annotierung graphischer Elemente in Visualisierungen kann die Information aus Scan-Paths verwendet werden, um Verstehensprozesse zu analysieren und diese miteinander zu vergleichen. Diese Annahme ba-

siert auf der Augen-Gedanken-Hypothese (Abschnitt 2.7.1), die besagt, dass das, was wir betrachten, maßgeblich auch kognitive Prozesse beeinflusst. Diese grundlegende Annahme stellt zugleich auch die Grenzen der hier vorgestellten Ansätze und Techniken dar. Eine ausführliche Diskussion dieser Grenzen findet sich in Kapitel 5.6.

Kognitionswissenschaften

Meist werden Evaluationsexperimente in der Visualisierungsforschung inkrementell durchgeführt (Abschnitt 2.4.1). Zunächst wird ein Konzept der Benutzerstudie, basierend auf gegebenen Forschungsfragen, erstellt. Dann werden die Stimuli entworfen, die Anzahl der Stimuli im Experiment festgelegt, Aufgabenbeschreibungen und Übungsaufgaben formuliert. Anschließend wird eine Pilotstudie durchgeführt. Ergebnisse aus dieser Pilotstudie werden analysiert, um den experimentellen Aufbau weiter zu optimieren. Erst dann wird das eigentliche Hauptexperiment durchgeführt, dessen Ergebnisse abschließend analysiert werden, um eingangs formulierte Hypothesen zu beweisen oder zu falsifizieren (Abschnitt 2.4.1).

Dieser inkrementelle Vorgang ist sehr zeitaufwändig. Im Gegensatz dazu ist die Modellierung von visuellen Suchstrategien (Abschnitte 2.7.5 und 2.7.7) in der Visualisierungsforschung bisher nicht weit verbreitet. Analog zu Argumenten in der Mensch-Computer-Interaktionsforschung von John et al. (John et al., 2004) kann davon ausgegangen werden, dass der Aufwand für die Erstellung von Modellen zur Simulation von visuellen Aufgaben, sogar von einfachen, als zu hoch wahrgenommen wird, um die Vorteile der Simulation von Augenbewegungen zu rechtfertigen. Um die Erstellung von Modellen zur Simulation von visuellen Aufgaben zu unterstützen, werden in dieser Arbeit zwei verschiedene Ansätze zur Modellierung der visuellen Suche vorgestellt.

Der erste Ansatz besteht darin, auf Basis des in den Kognitionswissenschaften weit verbreiteten ACT-R-Frameworks (Abschnitt 2.7.6), ähnlich dem in der Mensch-Computer-Interaktionsforschung eingesetzten CogTool (Abschnitt 2.7.8), eine visuelle Suche zu simulieren. Für eine technische Realisierung des Simulationswerkzeuges werden Ergebnisse aus kognitionswissenschaftlichen Studien verwendet, die zeigen, dass das Gehirn Informationen über visuelle Elemente auf einem Stimulus bezüglich ihrer Position (WO-System) und ihrer Eigenschaft (WAS-System) erfasst (Abschnitt 2.2.2 und 2.7.2). Beide Informationen stehen durch die semantische Annotierung von graphischen Elementen in einer Visualisierung für die Kognitionssimulation bereit. Als Ergebnis liefert die Simulation eine Reihenfolge von Fixationen. Diese können anschließend wieder mit Er-

gebnissen aus Eye-Tracking-Experimenten verglichen werden. Damit folgt diese Arbeit der Methode der kognitionswissenschaftlichen Arbeitsweise (Abschnitt 2.6.2). Der zweite Ansatz erlaubt es, Durchführungszeiten von visuellen Aufgaben mit Hilfe eines operatorenbasierten Modells ähnlich dem Keystroke-Level-Model (Abschnitt 2.7.4) aus der Mensch-Computer-Interaktionsforschung zu berechnen.

3.1 Interdisziplinärer Ansatz

Abbildung 3.2 zeigt den dieser Arbeit zugrunde liegenden interdisziplinären Ansatz. In dieser Arbeit werden Eye-Tracking-Experimente dazu eingesetzt, um experimentelle Daten zu liefern, mit denen das Leseverhalten von Visualisierungen bei der Durchführung von gegebenen visuellen Aufgaben analysiert werden kann. Aus dem Leseverhalten abgeleitete Leseregeln werden dann in Modellen und Simulationen der visuellen Suche verwendet. Ergebnisse aus Eye-Tracking-Experimenten werden aber auch mit vorhergesagten Ergebnissen der Modelle und Simulationen verglichen. Durch diesen Vergleich können die erstellten Modelle und Simulationen weiter optimiert werden. Die aufgenommenen Eye-Tracking-Daten werden dazu

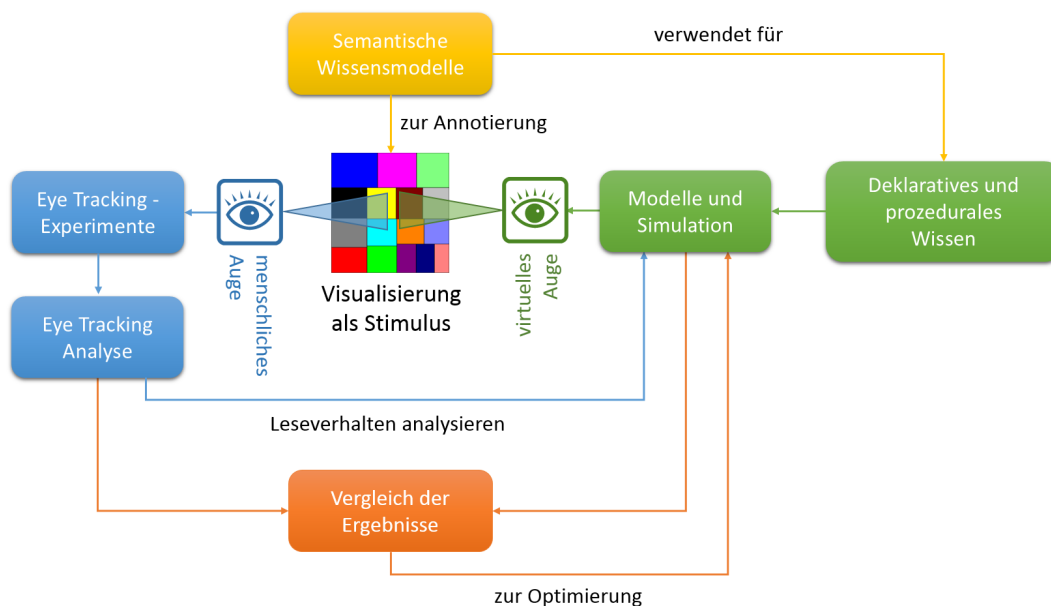


Abbildung 3.2: In dieser Arbeit werden Modelle und Techniken aus den drei Gebieten Eye-Tracking-Visualisierung, semantische Wissensmodelle und den Kognitionswissenschaften kombiniert, um kognitive und perceptiv Prozesse bei der Arbeit mit Visualisierungen zu untersuchen.

hinsichtlich ihrer räumlichen und zeitlichen Eigenschaften (*WO-Raum*) und semantischen Struktur (*WAS-Raum*) mit visuellen und statistischen Analysemethoden sowie mit der visuellen Analytik analysiert. Für die Analyse der Augenbewegungen im *WAS-Raum* wird ein semantisches Wissensmodell verwendet. Dieses wird auch für die Modellierung und Simulation visueller Suchstrategien herangezogen. Damit basiert sowohl die Analyse der Eye-Tracking-Daten als auch die Modellierung von Augenbewegungen auf dem gleichen Wissensmodell.

3.2 Beitrag dieser Arbeit

Ein Hauptinteresse bei der Auswertung von Augenbewegungen, bei der Durchführung einer Aufgabe mit einer gegebenen Visualisierung, gilt der Analyse des Blickverhaltens. Eine interessante Fragestellung aus Sicht der Visualisierungsforschung ist es, ob ähnliche Augenbewegungsmuster über mehrere Probanden hinweg existieren und ob damit verschiedene Gruppen von Probanden identifiziert werden können, die ähnliche Strategien zum Lesen der gleichen Visualisierung verwendet haben. Ergebnisse aus dieser Analyse können dafür genutzt werden, um existierende Visualisierungen an die Bedürfnisse von bestimmten Benutzergruppen besser anzupassen oder neue Visualisierungstechniken benutzerorientiert zu entwickeln.

Für die Analyse von Augenbewegungen wird in dieser Arbeit ein Konzept zur systematischen Entwicklung von neuartigen Visualisierungstechniken für Eye-Tracking-Daten beschrieben (Raschke et al., 2014a). Dieses Konzept wurde bei der Entwicklung der parallelen Scan-Path-Visualisierungstechnik (Raschke et al., 2012) verwendet. Mit einer Erweiterung der parallelen Scan-Path-Visualisierungstechnik, durch Techniken aus dem Gebiet der visuellen Analytik, können Ähnlichkeiten zwischen Scan-Path-Strukturen gefunden werden, um das Problem des Visual Clutters zu reduzieren (Raschke et al., 2014d).

Um kognitive Prozesse analysieren zu können, ist die semantische Bedeutung fokussierter einzelner Bereiche auf dem Stimulus wichtig. Da Stimuli meist aber als Ergebnis eines Bildsyntheseprozesses in Form von Rastergraphiken vorliegen, ist die semantische Bedeutung einzelner Bereiche im Stimulus nicht direkt zugänglich. Diese Arbeit legt eine Methode vor, mit der graphische Elemente in Visualisierungen automatisch schon während des Renderings annotiert werden können (Raschke et al., 2014c). Um Scan-Paths im *WAS-Raum* analysieren zu können, wurde die parallele Scan-Path-Visualisierungstechnik erweitert (Raschke et al., 2014b).

Diese Arbeit trägt zwei unterschiedliche Ansätze bei, um die visuelle Suche in Visualisierungen zu modellieren und zu simulieren. Der erste Ansatz

besteht in einer Kognitionssimulation, die Leseregeln aus Eye-Tracking-Experimenten zusammen mit einem Wissensmodell über Visualisierungen verwendet, um Scan-Paths zu berechnen (Raschke et al., 2013). Der zweite Ansatz besteht in einer operatorenbasierten Methode, mit der Durchführungszeiten für visuelle Aufgabe vorhergesagt werden können. Dazu wird in dieser Arbeit erstmalig ein einfaches Diagram-Viewing-Modell vorgestellt, das mit Hilfe eines Satzes von primitiven Operatoren und Heuristiken ähnlich dem Keystroke-Level-Modell Durchführungszeiten für eine visuelle Aufgabe berechnet.

Insgesamt können mit Hilfe des interdisziplinären Ansatzes folgende Aspekte der Augenbewegungen näher untersucht werden (siehe dazu auch Abbildung 3.3):

- Ergebnisse aus der Analyse von Eye-Tracking-Experimenten im WO- und WAS-Raum werden verwendet, um Modelle der visuellen Suche zu entwickeln. Damit kann die Durchführungszeit einer visuellen Aufgabe vorhergesagt und die dabei ausgeführten Fixationen, sowie deren Reihenfolge simuliert werden.
- Für die Analyse von kognitiven Prozessen bei der Durchführung von visuellen Aufgaben, als auch für die Entwicklung von Simulationen der visuellen Suche, werden gleiche Modelle für die Repräsentation von Wissen über Visualisierungen verwendet. Dies erlaubt einen direkten Vergleich der experimentellen Ergebnisse mit den simulierten Augenbewegungen.
- Durch die Annotation von graphischen Elementen in Visualisierungen wird eine Analyse von Scan-Paths im WAS-Raum möglich. Damit kann unter anderem untersucht werden, welche graphischen Elemente in einer Visualisierung zu welchen kognitiven und perzeptiven Prozessen führen.
- Werden Methoden und Techniken aus allen drei Teilbereichen zusammen verwendet, wird der Aufwand zur Modellierung und Simulation der visuellen Suche reduziert. Damit wird ein Beitrag geleistet, Visualisierungen effizienter zu evaluieren. Insbesondere erlaubt diese Verbindung einen iterativen Entwicklungsprozess von Visualisierungen.

3 Interdisziplinärer Ansatz für die Entwicklung von neuen Methoden und Techniken zur Evaluation von Visualisierungen



Abbildung 3.3: Einzelbeiträge der visuellen Analyse von Eye-Tracking-Experimenten (blau), der Repräsentation von Wissen über Visualisierungen (grün) sowie der Modellierung und Simulation der visuellen Suche (gelb); sowie Beiträge, die durch die Kombination der Einzelergebnisse geleistet werden (rot und hellblau).

Visualisierung von Eye-Tracking-Daten

In diesem Kapitel wird die parallele Scan-Path-Visualisierungstechnik (PSP-Visualisierung) vorgestellt (Abschnitt 4.1), die der Klasse der Informationsvisualisierungen zugeordnet werden kann. Zunächst wird ihr grundlegendes Visualisierungskonzept beschrieben. Dann werden drei Varianten dieser Visualisierungstechnik vorgestellt. In Abschnitt 4.2 wird gezeigt, wie mit visueller Analytik, bei der die PSP-Visualisierung als Visualisierungskomponente eingesetzt wird, Scan-Paths quantitativ miteinander verglichen werden können. Abschnitt 4.3 demonstriert die Anwendung der PSP-Visualisierung mit zwei Beispielen. Das Kapitel schließt mit einer Einordnung der parallelen Scan-Path-Visualisierungstechnik in existierende Visualisierungstechniken für Eye-Tracking-Daten.

4.1 Parallele Scan-Path-Visualisierung

Mit der PSP-Visualisierung können Augenbewegungen hinsichtlich der folgenden Fragestellungen untersucht werden:

1. Gibt es generelle Augenbewegungsmuster, wenn Probanden Visualisierungen betrachten?
2. Können verschiedene Nutzergruppen auf Basis ihrer Augenbewegungen unterschieden werden?
3. Wie sehen verschiedene visuelle Lösungsstrategien bei der Durchführung von Aufgaben aus?
4. Welche Rückschlüsse können auf kognitive und perzeptive Prozesse gezogen werden?

Teile dieses Abschnitts wurden veröffentlicht in:

Raschke, Michael; Chen, Xuemei; Ertl, Thomas: Parallel Scan-Path Visualization. In: *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications, ETRA '12*, S. 165-168, 2012.

In diesem Abschnitt wird beschrieben, wie die PSP-Visualisierungstechnik für die Beantwortung der ersten drei Fragen eingesetzt werden kann. Die Anwendung der PSP-Visualisierung für die vierte Frage wird in Kapitel 5 beschrieben. Um die Suche nach gleichen Scan-Path-Mustern zu unterstützen, besitzt die PSP-Visualisierungstechnik folgende Eigenschaften:

- Zur Auswertung der Augenbewegungen werden die Positionen der Fixationen auf dem Stimulus und die Fixationsdauern verwendet.
- Die Visualisierungstechnik kann einen gesamten Scan-Path oder auch nur Abschnitte davon darstellen. Zusätzlich werden Fixationen bezüglich AOIs und verschiedener Probanden gefiltert.
- Die Reihenfolge der Fixationen wird strukturiert dargestellt.
- Die Darstellung von verschiedenen Eigenschaften der Scan-Paths, wie Verweildauern und Zeitpunkte von einzelnen Fixationen, werden visualisiert.
- Eine schnelle Erkennung von gleichen Augenbewegungsmustern wird unterstützt.
- Sowohl zeitliche als auch räumliche Eigenschaften der Scan-Paths sind auf einen Blick in der Visualisierung erkennbar.
- Die Visualisierungstechnik ist skalierbar und sie erlaubt es, eine große Anzahl von Scan-Paths zu untersuchen.

Die PSP-Visualisierung basiert auf AOIs und bildet Verweildauern und Fixationen entlang von Achsen ab. Abbildung 4.1 zeigt eine Skizze der Visualisierungstechnik, bei der drei AOIs auf drei Koordinatenachsen in der Visualisierung abgebildet werden. Die Achse auf der linken Seite zeigt die Zeit an und läuft hier von unten nach oben. Die Ausrichtung der Zeitachse ist flexibel und kann nach Nutzeranforderung auch von oben nach unten oder von links nach rechts verlaufen. Im Folgenden wird immer eine vertikale Achse zur Abbildung des Zeitverlaufs von unten (Start des Experiments) nach oben (Ende des Experiments) genutzt. Die horizontale Achse zeigt alle ausgewählten AOIs. Die Reihenfolge der AOIs auf der AOI-Achse kann beliebig geändert werden.

Um im Folgenden einen gewöhnlichen Scan-Path von Scan-Paths, die über AOIs laufen, besser unterscheiden zu können, wird im Kontext der PSP-Visualisierung die darin dargestellte Augenbewegung als *Fixationspfad* bezeichnet. In Abbildung 4.1 startet der Fixationspfad in AOI 2, es folgt eine Transition nach AOI 1. Dort verweilen die Augen für eine Zeitdauer.

Anschließend erfolgt wieder eine Transition zurück nach AOI 2. Fixationen, die während des Experiments nicht in eine AOI fallen, sind in Abbildung 4.1 als gestrichelte Linien dargestellt. Das gleiche gilt für Zeiträume, in denen keine Fixationen aufgenommen worden sind. In diesem Beispiel läuft der Fixationspfad dann von AOI 2 weiter zu AOI 3.

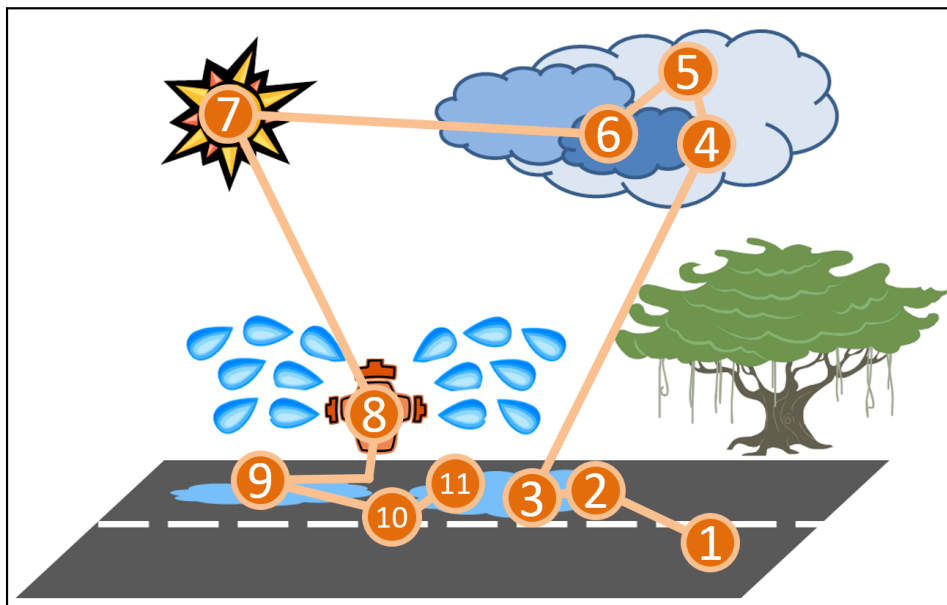
Es wurden drei Varianten der PSP-Visualisierung entwickelt (Abbildungen 4.3 bis 4.5). Mit diesen drei Varianten können verschiedene Charakteristika der Fixationspfade untersucht werden:

1. Verweildauer-Sequenz-Diagramm (Gaze Sequence Diagram): Zeigt nur Fixationspfade
2. Fixationspunkt-Diagramm (Fixation Point Diagram): Zeigt Fixationspfade zusammen mit einzelnen Fixationen innerhalb von AOIs sowie Transitionsfrequenzen
3. Verweildauer-Verteilungs-Diagramm (Gaze Duration Distribution Diagram): Zeigt Fixationspfade zusammen mit Transitionsfrequenzen und Verweildauern

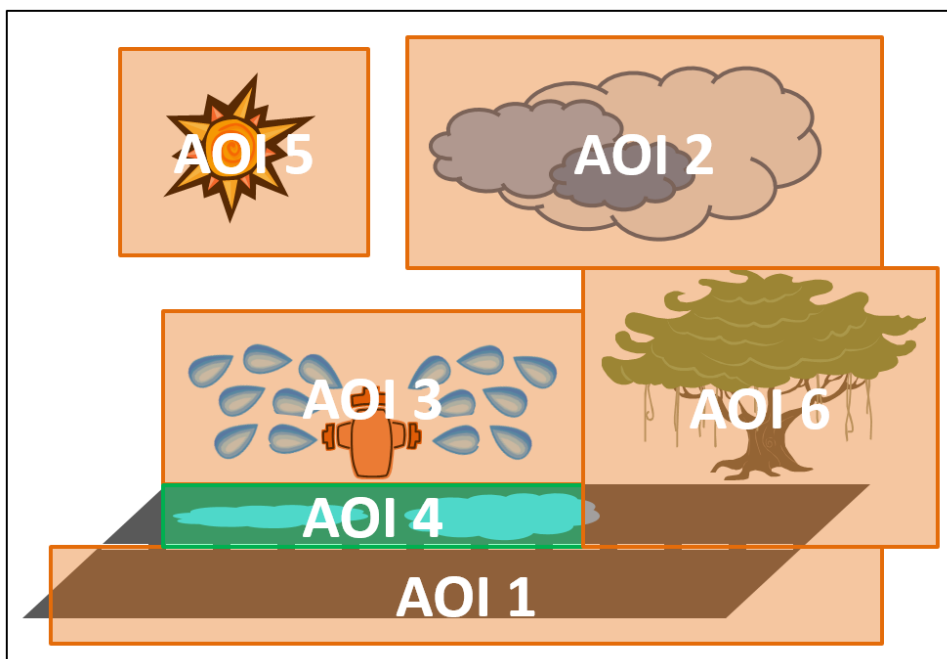
Die drei Varianten der PSP-Visualisierung können die Fixationspfade jeweils auf einem zeitabhängigen Gitter oder auf einem zeitunabhängigen Gitter darstellen. Zeitunabhängige PSP-Visualisierungen stellen dabei nur Transitionen der Fixationspfade dar, ohne die Verweildauern in der Visualisierung abzubilden.

Verweildauer-Sequenz-Diagramm

Abbildung 4.2a) zeigt ein illustratives Beispiel einer Scan-Path-Visualisierung zu der Frage „Warum ist die Straße nass?“, die mit dem dargestellten Stimulus beantwortet werden soll. Abbildung 4.2b) zeigt AOIs, die für die PSP-Visualisierungen definiert worden sind. Diese decken alle aufgabenrelevanten Bereiche auf dem Stimulus ab. Abbildung 4.3 zeigt das Verweildauer-Sequenz-Diagramm (Gaze Duration Sequence Diagram) des gleichen Blickverhaltens. Dieses Beispiel zeigt den Fixationspfad eines Probanden. Jede kontinuierliche Linie auf einer vertikalen Achse repräsentiert eine Verweildauer innerhalb einer AOI (Abbildung 4.2b). Horizontale Linien zeigen einen Wechsel der Aufmerksamkeit von einer AOI zu einer anderen. In der Abbildung zu sehen, ist ein Fixationspfad startend in AOI1, der dann zu AOI4 verläuft, dann zu AOI2, AOI5, AOI3 und schließlich bei AOI4 endet. Durch die Betrachtung der Zeitachse können Parameter wie Startzeit und Endzeit der Verweildauern innerhalb der AOIs sowie die Verweildauer pro AOI erkannt werden. Der Verlauf der Aufmerksamkeit des Probanden auf



(a)



(b)

Abbildung 4.2: a) Illustrativer Scan-Path auf einem Stimulus zur Frage "Warum ist die Straße nass?". Dieser Scan-Path wird in den folgenden Beispielen (Abbildung 4.3, 4.4, 4.5 und 4.6) der PSP-Visualisierung verwendet. b) AOIs, die auf dem Stimulus definiert wurden.

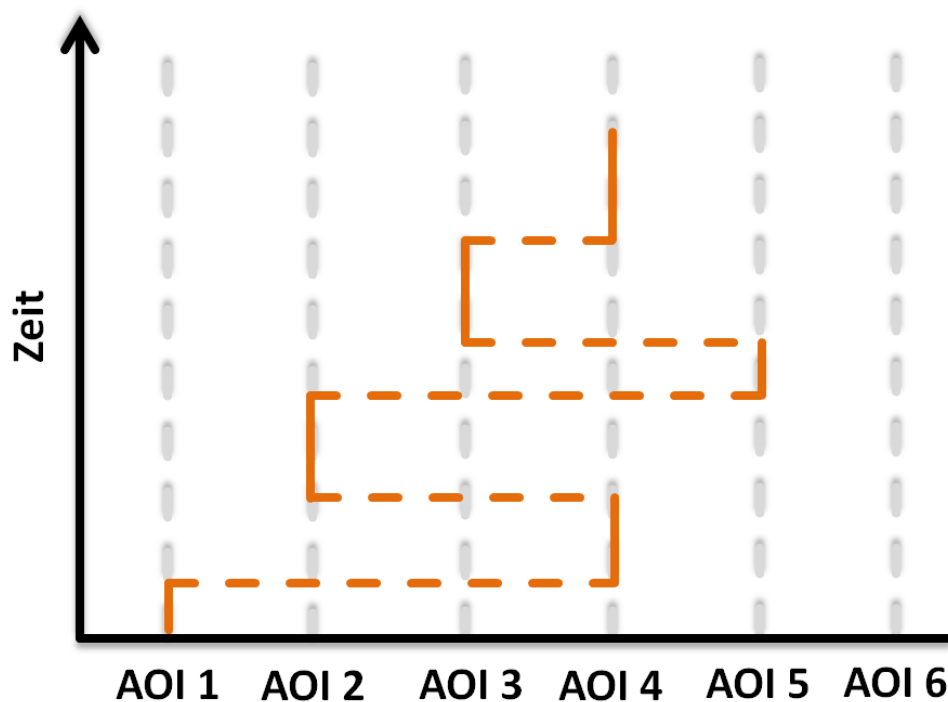


Abbildung 4.3: Verweildauer-Sequenz-Diagramm mit einem Fixationspfad, der in AOI1 beginnt, dann über AOI4, AOI2, AOI5, AOI3 zur AOI4 verläuft. Im Diagramm zu erkennen sind Start- und Endzeiten von Verweildauern in AOIs. Der Verlauf der Aufmerksamkeit des Probanden kann durch das Folgen des Fixationspfads in der Visualisierung untersucht werden.

dem Stimulus kann durch das Folgen der orange eingefärbten Linie in der Visualisierung untersucht werden.

Fixationspunkt-Diagramm

Fixationspunkt-Diagramme (siehe Abbildung 4.4) zeigen neben dem Verlauf der Aufmerksamkeit auf einem Stimulus einzelne Fixationen. In einem Fixationspunkt-Diagramm werden Fixationen innerhalb einer AOI auf die AOI-Achse als Kreise abgebildet. Die Verweildauer wird mit einem vertikalen Strich auf der AOI-Achse hervorgehoben, der beim Zeitpunkt des Eintritts des Scan-Paths in die AOI beginnt und beim Austritt des Scan-Paths aus der AOI endet. Für jede Gruppe von Fixationen werden deren zeitliche Mittelpunkte mit aufsteigenden Linien miteinander verbunden. Mit Hilfe des Fixationspunkt-Diagramms können sowohl zeitliche Eigenschaften der Fixationen, zeitliche Eigenschaften der Verweildauern, sowie Transitionseigenschaften der Fixationspfade auf einen Blick untersucht werden. In Abbildung 4.4 sind die Augenbewegungen aus Abbildung 4.2a) mit den AOIs aus Abbildung 4.2b) dargestellt.

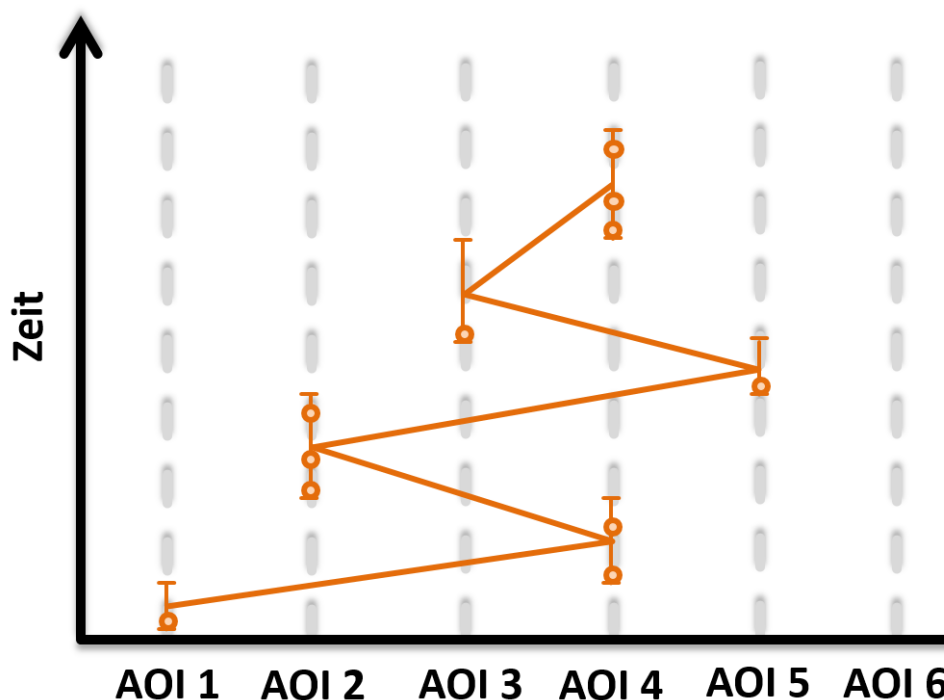


Abbildung 4.4: Mit Hilfe von Fixationspunkt-Diagrammen können zeitliche Eigenschaften der Fixationen, Verweildauern und Transitionseigenschaften der Fixationspfade untersucht werden. Die Verweildauer wird dazu mit einem vertikalen Strich auf der AOI-Achse hervorgehoben. Zusätzlich werden die Zeitpunkte einzelner Fixationen in den AOIs durch Kreise dargestellt.

Verweildauer-Verteilungs-Diagramm

Abbildung 4.5 zeigt ein Verweildauer-Verteilungs-Diagramm des Scan-Paths aus Abbildung 4.2a) über die AOIs in Abbildung 4.2b). Das Verweildauer-Verteilungs-Diagramm basiert auf einer Eye-Tracking-Visualisierung für Webseiten von Guan und Cutrell (Guan und Cutrell, 2007). Ein Verweildauer-Verteilungs-Diagramm verwendet eine kombinierte Darstellung der Fixationspfade aus einem Fixationspunkt-Diagramm und dem Verweildauer-Sequenz-Diagramm. Jedoch zeigt ein Verweildauer-Verteilungs-Diagramm nicht direkt den Zeitpunkt der Transitionen. Das Verweildauer-Verteilungs-Diagramm zeigt nur die zeitlichen Mittelpunkte der Verweildauern als Punkte auf den AOI-Achsen an. Diese Mittelpunkte sind mit Linien verbunden, die Transitionen der jeweiligen Probanden zeigen. Zusätzlich ist ein Balkendiagramm über die Darstellung der Fixationspfade gelegt. Dieses zeigt den Anteil der Verweildauern pro Proband an der gesamten Betrachtungsdauer des Stimulus. So können sowohl der Fixationspfad als auch statistische Metriken mit einem Blick erfasst werden.

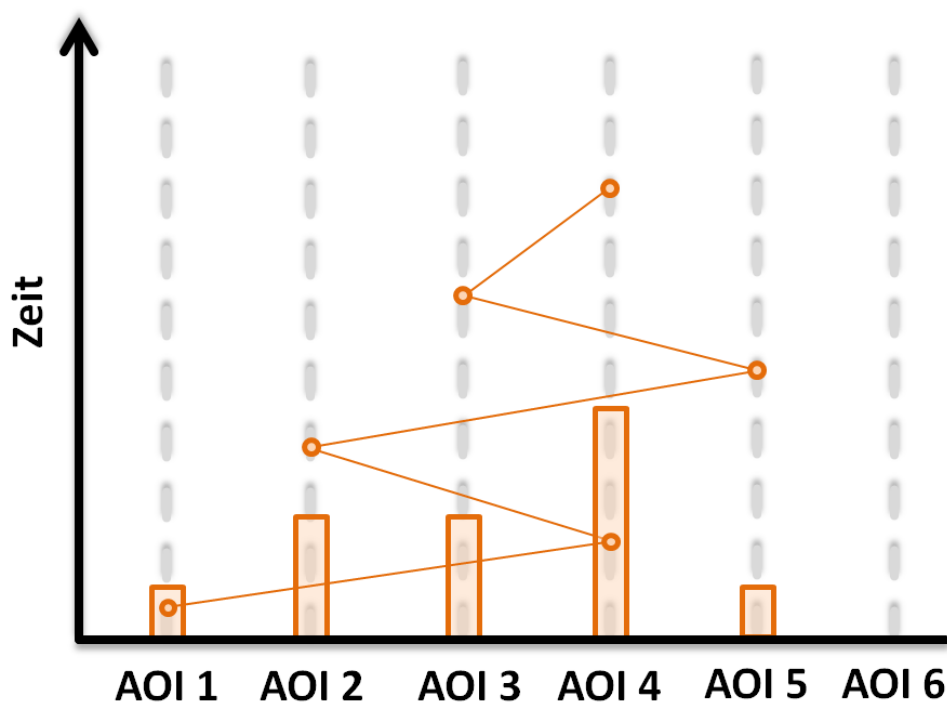


Abbildung 4.5: Verweildauer-Verteilungs-Diagramme zeigen die zeitlichen Mittelpunkte der Verweildauern als Punkte auf den AOI-Achsen an. Diese Mittelpunkte sind mit Linien verbunden, die Transitionen der jeweiligen Probanden zeigen. Zusätzlich ist ein Balkendiagramm über die Darstellung der Fixationspfade gelegt, das die Verweildauern pro Proband als Anteil an der gesamten Betrachtungsdauer des Stimulus anzeigt.

Zeitunabhängige PSP-Visualisierung

Eine zeitunabhängige PSP-Visualisierung (siehe Abbildung 4.6) zeigt nur Transitionen des Fixationspfads zwischen den AOIs auf einem zeitunabhängigen Gitter. Dabei werden zeitliche Charakteristika der Fixationspfade nicht in der Visualisierung abgebildet. Die Visualisierung fasst die Fixationspfade dahingehend zusammen, dass nur der Eintritt eines Scan-Paths in eine AOI und dessen Austritt beachtet und in der Visualisierung abgebildet wird. Diese Fixationen werden als Kreise in der zeitunabhängigen PSP-Visualisierung dargestellt. Fixationen während der Verweildauer werden nicht visualisiert. Wie im Verweildauer-Sequenz-Diagramm werden die Transitionen wieder mit horizontalen Linien dargestellt. Mit der zeitunabhängigen PSP-Visualisierung kann das grundsätzliche Leseverhalten von Visualisierungen untersucht und Fixationspfade von mehreren Probanden miteinander verglichen werden.

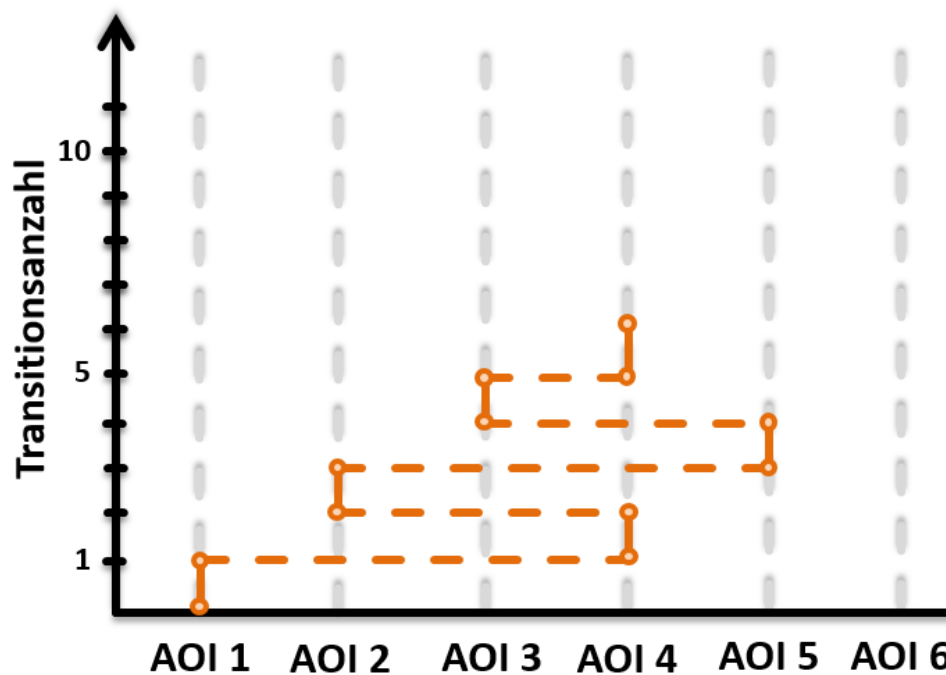


Abbildung 4.6: Die zeitunabhängige PSP-Visualisierung zeigt nur Transitionen des Fixationspfads zwischen AOIs auf einem zeitunabhängigen Gitter. Zeitliche Charakteristika der Fixationspfade werden in der Visualisierung nicht abgebildet.

Interaktion mit der PSP-Visualisierung

Komplexe Strukturen in den Fixationspfaden können durch das Hineinzoomen in die PSP-Visualisierung in einer höheren Auflösung dargestellt werden. Zeigt der Mauszeiger auf einen Fixationspfad, wird dieser hervorgehoben und es werden zusätzliche Informationen zum Namen des Probanden oder zum Fixationspfad mit einem Tool-Tip eingeblendet (siehe Abbildung 4.7 (1)). Die Farben der Fixationspfade können frei für jeden Probanden gewählt werden. Probanden und AOIs werden vom Benutzer in einem separaten Dialog für die Darstellung in der PSP-Visualisierung an- oder abgewählt.

Das AOI-Management (Abbildung 4.7 (2)) unterhalb des Visualisierungsbereichs erlaubt es, AOIs direkt für die Visualisierung an- oder abzuschalten. Zusätzlich können sich AOIs in verschiedenen Ebenen übereinander befinden. Die Einschränkung dabei ist, dass jede Unter-AOI komplett von ihrer Über-AOI umschlossen wird. Die Reihenfolge der AOI-Ebenen kann beliebig geändert werden. Die PSP-Visualisierung aktualisiert sich automatisch, wenn die AOI-Auswahl oder die Anordnung der AOIs geändert wird. Die Berechnung einer Transitionsmatrix mit anschließender Sortie-

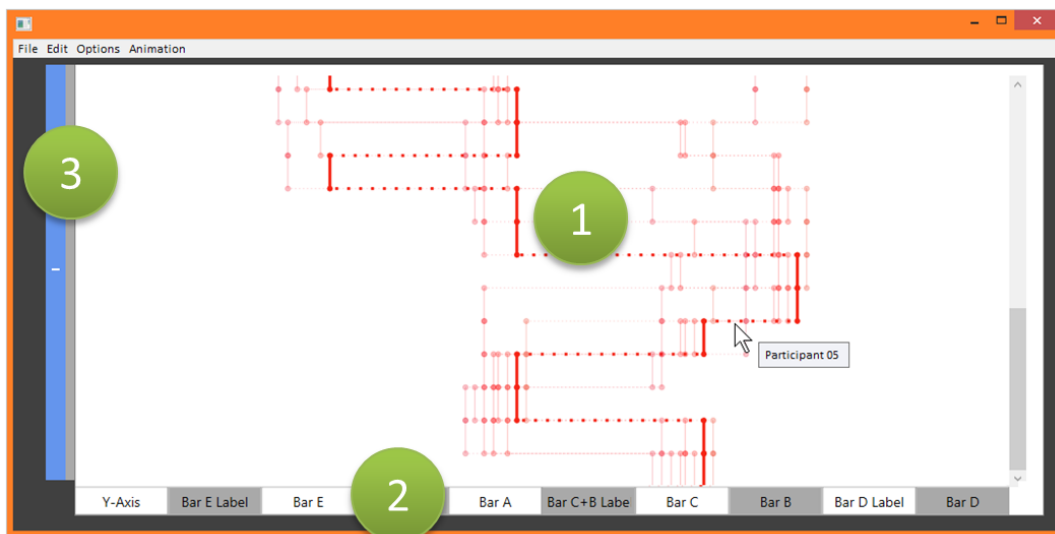


Abbildung 4.7: Die PSP-Visualisierung bietet die Möglichkeit, Fixationspfade in der Visualisierung hervorzuheben, mit Tool-Tips Informationen einzublenden und in die Visualisierung hineinzuzoomen (1). Die Reihenfolge der AOIs kann beliebig verändert werden (2). Zusätzlich können AOIs aktiviert oder deaktiviert werden. Mit einer Szenenleiste kann ein längeres Eye-Tracking-Experiment in einzelne Szenen unterteilt werden (3).

Die Reihenfolge der Transitions nach ihrer Häufigkeit kann dazu eingesetzt werden, eine optimale Reihenfolge der AOIs zu berechnen. Damit wird erreicht, dass die Fixationspfade in der PSP-Visualisierung sich möglichst wenig überschneiden.

Wenn Augenbewegungen über einen längeren Zeitraum hinweg aufgenommen wurden, kann es hilfreich sein, das Eye-Tracking-Experiment in einzelne Abschnitte aufzuteilen. Dies wird durch die Unterteilung des Experiments in Szenen ermöglicht (Abbildung 4.7 3). Beispielsweise kann jede Szene eine bestimmte Aufgabe darstellen, die vom Probanden durchgeführt werden musste. Ähnlich wie beim AOI-Management können Szenen aktiviert oder deaktiviert werden. Dies hat wie bei der Änderung von AOIs eine direkte Erneuerung der PSP-Visualisierung zur Folge. Zusätzlich unterstützen Vorschau-Bilder die Navigation innerhalb der Szenen. Die Unterteilung in einzelne Szenen kann aber nur verwendet werden, wenn alle Probanden zu jedem Zeitpunkt des Experiments immer den gleichen Stimulus betrachteten und dabei die gleiche Aufgabe durchgeführt haben.

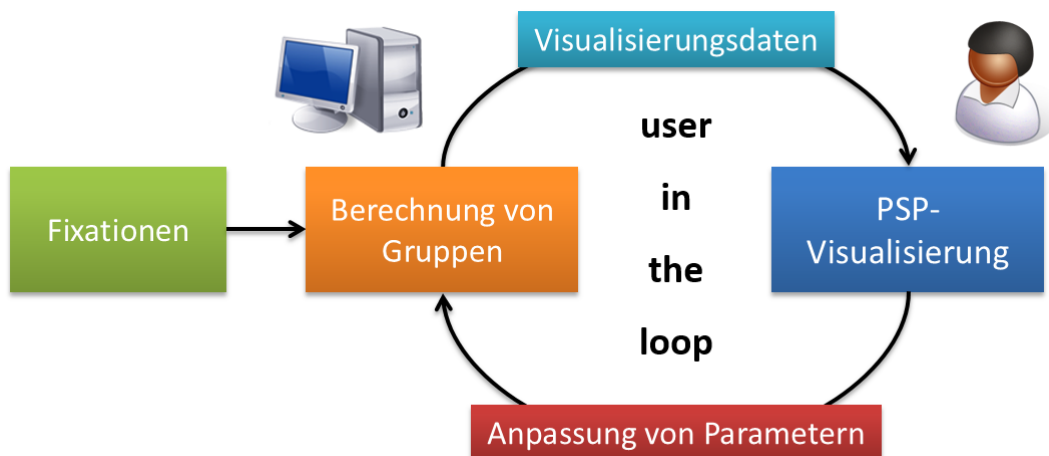


Abbildung 4.8: Zum Vergleich von Augenbewegungen wird nach dem Ansatz der visuellen Analytik die Leistungsfähigkeit von rechnergestützten Methoden verwendet und die menschliche Fähigkeit genutzt, visuelle Muster zu erkennen.

4.2 Visueller Vergleich von Scan-Paths

In diesem Abschnitt wird eine Methode zum Vergleich von Augenbewegungsmustern vorgestellt. Ein Vergleich von Augenbewegungsmustern spielt eine wichtige Rolle zur Identifikation von ähnlichen Lesestrategien bei Visualisierungen. Durch einen Vergleich der Augenbewegungen können auch verschiedene Benutzergruppen erkannt werden (Abschnitt 4.3) und Schwachpunkte im Visualisierungskonzept identifiziert werden (Abschnitt 5.4). Für die Simulation kognitiver Prozesse wird zu jeder Benutzergruppe dann ein repräsentativer Fixationspfad ausgewählt, mit dem Leseregeln für Visualisierungen formuliert werden (Abschnitt 5.5).

Für einen Vergleich der Augenbewegungen müssen die aufgenommenen Scan-Paths bezüglich ihrer Ähnlichkeit hin untersucht werden. In der Literatur werden dazu mehrere Lösungen, basierend auf String-Editing-Techniken, vorgestellt (siehe Kapitel 2.5.2). Die in dieser Arbeit präsentierte Methode verwendet ebenfalls AOIs und String-Editing-Methoden, geht jedoch einen Schritt weiter und erlaubt es, mit Hilfe der visuellen Analytik Gruppen von ähnlichen Augenbewegungen zu finden. Mit dem sogenannten „User in the Loop“-Ansatz der visuellen Analytik werden Fixationspfade mit

Teile dieses Abschnitts basieren auf:

Raschke, Michael; Herr, Dominik; Blascheck, Tanja; Burch, Michael; Schrauf, Michael; Willmann, Sven; Ertl, Thomas: A Visual Approach for Scan Path Comparison. In: *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '14*, S. 135-142, 2014.

Hilfe eines hierarchischen agglomerativen Clustering Algorithmus (HAC) in Gruppen aufgeteilt. Dieser Ansatz erlaubt es, sowohl die Leistungsfähigkeit von rechnergestützten Methoden einzusetzen, als auch die menschliche Fähigkeit zu nutzen, visuelle Muster zu erkennen (Abbildung 4.8). Als ein Distanzmaß zur Bildung der Gruppen wird die Levenshtein-Distanz (Kapitel 2.5.2) verwendet. Ähnliche Fixationspfade werden anschließend in der PSP-Visualisierung mit einem ähnlichen Farbton dargestellt. Parameter des Clustering-Algorithmus können interaktiv weiter verändert werden, bis eindeutige Lesestrategien von Visualisierungen gefunden wurden.

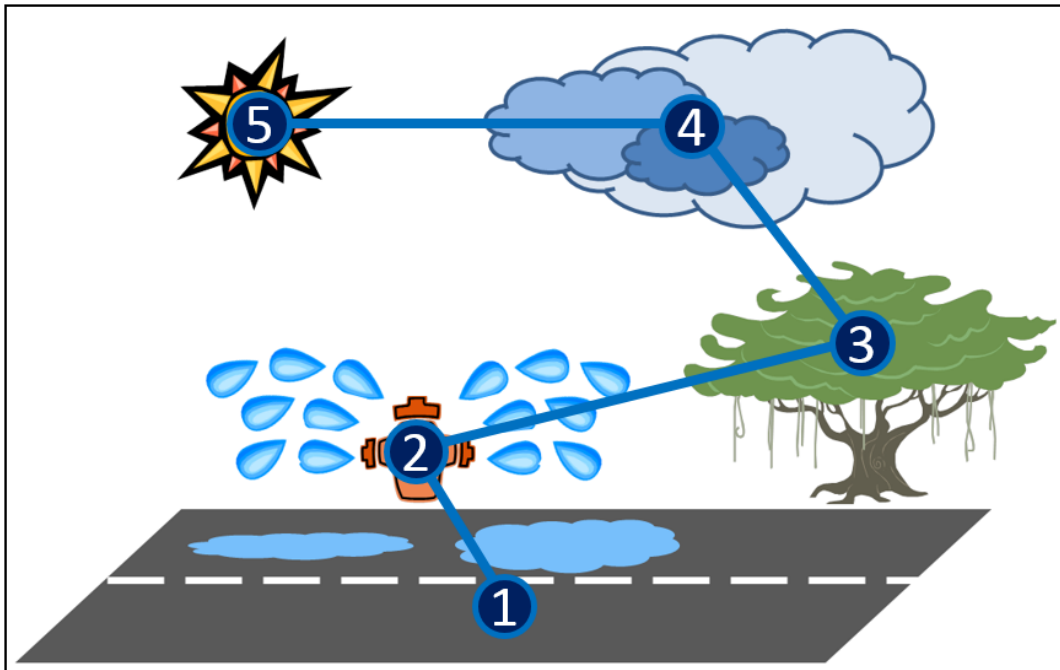
Im Folgenden wird zunächst im Detail beschrieben, wie die Erkennung von ähnlichen Scan-Paths funktioniert und wie ähnliche Scan-Paths mit der PSP-Visualisierung dargestellt werden (Abschnitt 4.2.1). Anschließend wird das technische System hinter der Ähnlichkeitsberechnung von Fixationspfaden, sowie der verwendete Clustering-Algorithmus vorgestellt (Abschnitt 4.2.2).

4.2.1 Konzept der Ähnlichkeitsbestimmung

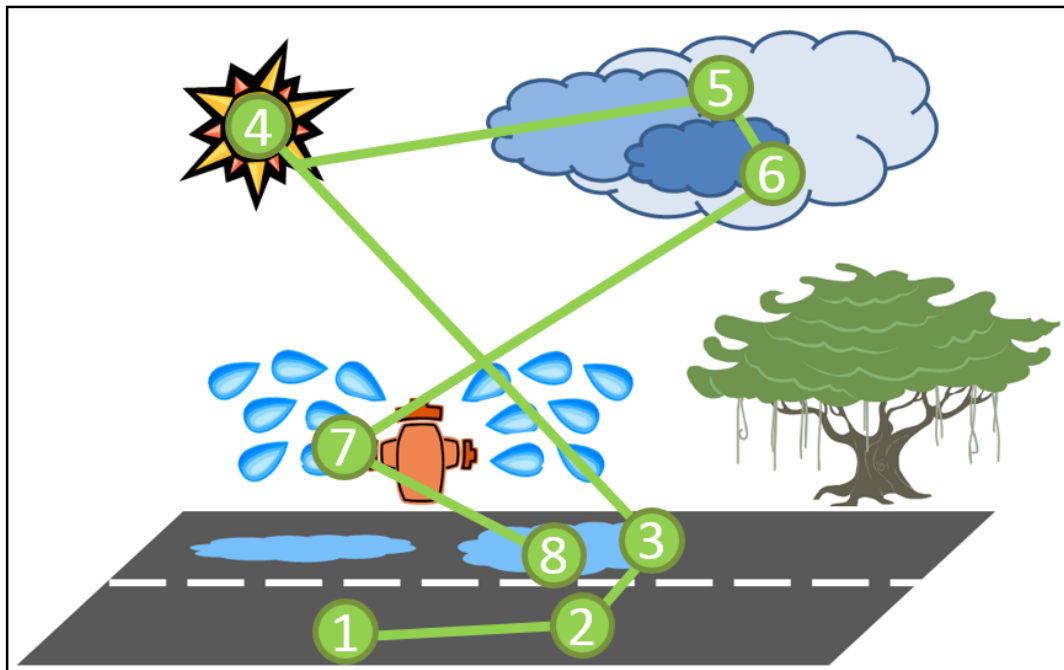
Der Scan-Path in Abbildung 4.2a) und die beiden in den Abbildung 4.9a) und b) gezeigten Scan-Paths werden im Folgenden verglichen. Die Methode zum Vergleich der Scan-Paths verwendet AOIs. Diese sind in Abbildung 4.2b) dargestellt. Abbildung 4.10a) zeigt die Visualisierung der drei Scan-Paths in einer PSP-Visualisierung ohne die Anwendung einer Ähnlichkeitsbestimmung. Der Analyst hat nun zwei Möglichkeiten, die Fixationspfade miteinander zu vergleichen.

Zum einen kann er die PSP-Visualisierung ohne eine Datenaufbereitung und eine Unterstützung durch Clustering-Algorithmen verwenden. Dazu können die Fixationspfade umfangreich mit Hilfe von verschiedenen Interaktionstechniken analysiert werden, die in Abschnitt 4.1 beschrieben wurden. Dabei kann er zwischen einem zeitabhängigen Gitter, auf das die Fixationen abgebildet werden, und einem zeitunabhängigen Gitter hin- und herwechseln, um das Leseverhalten zu analysieren und ähnliche Augenbewegungen über mehrere Probanden hinweg zu finden.

Zum anderen kann er eine automatische Berechnung von ähnlichen Gruppen von Fixationspfaden einsetzen. Dazu wird vom Analysten ein Ähnlichkeitswert zwischen 0.0 und 1.0 vorgegeben, der bestimmt, ab wann Fixationspfade als ähnlich angesehen werden sollen. Ein Ähnlichkeitswert von 1.0 bedeutet, dass alle Fixationen der zu vergleichenden Fixationspfade in gleicher Reihenfolge den gleichen AOIs zugewiesen werden müssen. Das bedeutet, dass alle Fixationspfade räumlich exakt übereinstimmen und in gleicher Reihenfolge durch die AOIs hindurch laufen. Ein Ähnlichkeitswert von 0.0 bedeutet, dass keine der Fixationen, die AOIs zugewiesen worden

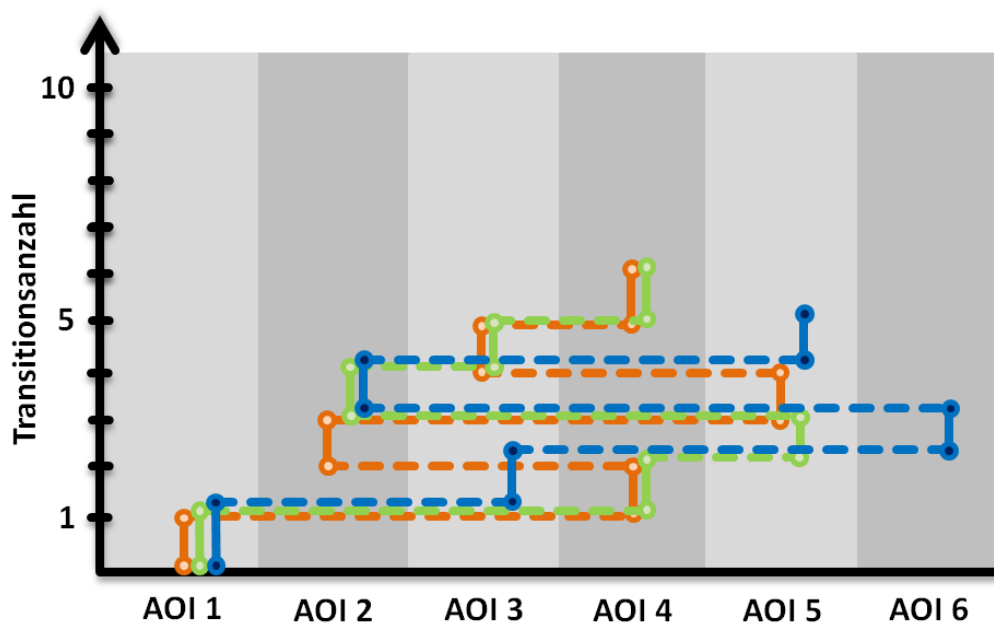


(a)

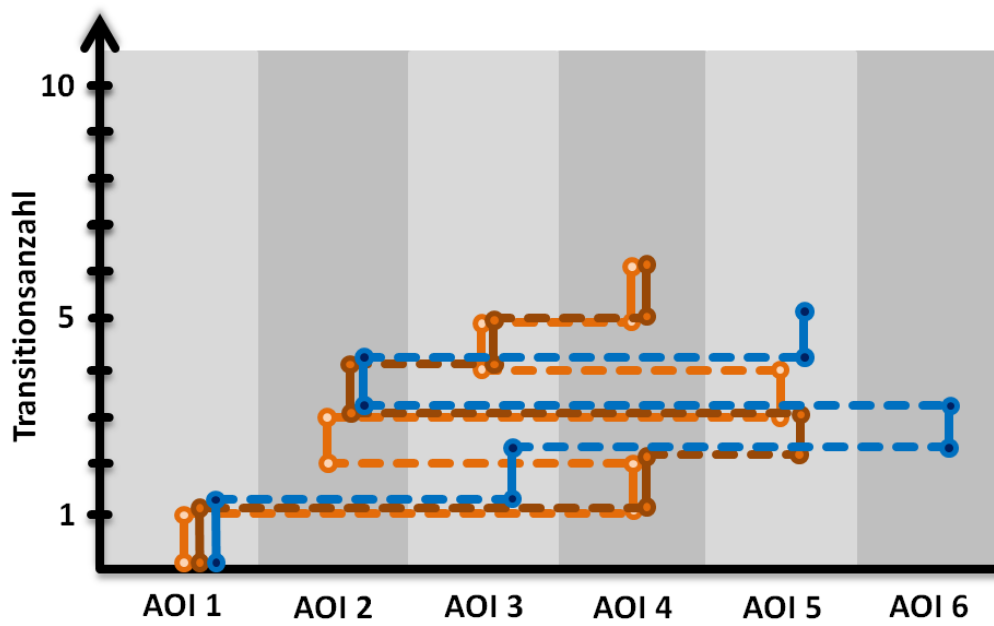


(b)

Abbildung 4.9: Zwei alternative Scan-Paths zur Lösung der Frage „Warum ist die Straße nass?“. In diesem Kapitel wird eine Methode vorgestellt, mit der mehrere Scan-Paths mit Hilfe der visuellen Analytik verglichen werden.



(a)



(b)

Abbildung 4.10: a) PSP-Visualisierung der Fixationspfade aus den Abbildungen 4.2 und 4.9 ohne Ähnlichkeitsberechnung. b) PSP-Visualisierung der Fixationspfade aus den Abbildungen 4.2 und 4.9 mit einer gefundenen Gruppe von ähnlichen Fixationspfaden (in rotem Farbtönen eingefärbt).

sind, räumlich oder zeitlich übereinstimmen müssen. Dies bedeutet, dass alle Fixationspfade als ähnlich angesehen werden, auch wenn sie sich in allen Positionen der Fixationen auf dem Stimulus voneinander unterscheiden. Alle Fixationspfade, für die eine Ähnlichkeit über diesem vorgegebenen Ähnlichkeitswert errechnet wird, werden in der PSP-Visualisierung in einem ähnlichen Farbton eingefärbt. Beispielsweise wurden in Abbildung 4.10b) zwei Fixationspfade als ähnlich erkannt und in einem rötlichen Farbton eingefärbt. Der dritte Fixationspfad, in der Abbildung in blau dargestellt, unterscheidet sich hingegen bei dem vorgegebenen Ähnlichkeitswert zu stark von den beiden anderen und wird deshalb in einer anderen Farbe dargestellt.

Zur Berechnung der Gruppen ähnlicher Fixationspfade wird das hierarchisch agglomerative Clustering eingesetzt (Manning et al., 2008), das in drei Schritten ähnliche Fixationspfade gruppiert:

1. Jede Fixation wird einem eigenen Cluster zugewiesen.
2. Die zwei Cluster mit der größten Ähnlichkeit werden zusammengefasst.
3. Schritt 2 wird solange wiederholt bis eine vorgegebene Unähnlichkeit der Cluster erreicht wird oder eine Mindestanzahl von Clustern erkannt wurde.

Fixationspfade, die in gleiche Cluster gruppiert wurden, werden in einem ähnlichen Farbton in der PSP-Visualisierung eingefärbt.

Der Benutzer kann die Parameter des Clustering-Algorithmus interaktiv in Abhängigkeit der visualisierten Fixationspfade verändern. Das Ergebnis der erneuten Berechnung von Clustern wird ihm dann anschließend in Echtzeit präsentiert. Dieser „User in the Loop“-Prozess wird solange durchgeführt, bis zufriedenstellende Ergebnisse explorativ gefunden werden oder eine gegebene Hypothese über zu erwartende Ähnlichkeiten des Blickverhaltens der Probanden unterstützt werden kann.

Der gleiche Algorithmus zum Identifizieren von Clustern mit ähnlichen Augenbewegungsmustern wird auch dafür eingesetzt, um innerhalb der einzelnen identifizierten Gruppen von Fixationspfaden *Repräsentanten* dieser Gruppen auszuwählen. Ein Repräsentant ist dabei der Fixationspfad innerhalb einer Gruppe ähnlicher Fixationspfade, der diese Gruppe am besten repräsentiert. Er besitzt zu allen anderen Fixationspfaden in seiner Gruppe den kleinsten Unterschied, also die größte Ähnlichkeit.

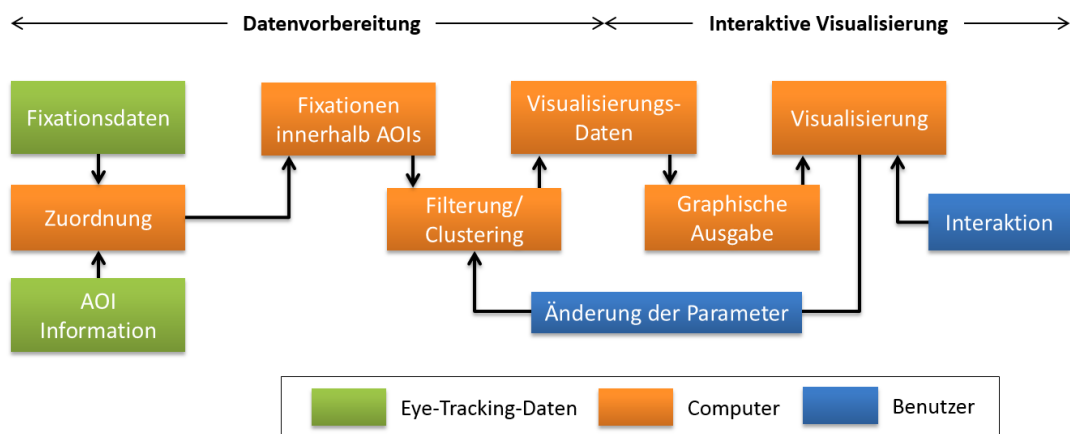


Abbildung 4.11: Analyse-Pipeline zur Vorbereitung der Eye-Tracking-Daten, zur Ermittlung ähnlicher Fixationspfade und zur Darstellung der aufbereiteten Augenbewegung mit der Möglichkeit, die Parameter des Clustering-Algorithmus interaktiv zu ändern.

4.2.2 Analyse-Pipeline zum Vergleich von Fixationspfaden

Abbildung 4.11 zeigt die Analyse-Pipeline zur Vorbereitung der Eye-Tracking-Daten, zur Ermittlung ähnlicher Fixationspfade und zur Darstellung der aufbereiteten Augenbewegung. Durch diese Pipeline ist eine flexible Ausgestaltung und Anpassung der Analyse und Visualisierung möglich. Im Folgenden werden die einzelnen Schritte „Zuordnung“, „Filterung/Clustering“, „Graphische Ausgabe“ und „Änderung der Parameter“ im Detail beschrieben.

Zuordnung der Fixationen

Der erste Schritt der Analyse-Pipeline ist die Zuordnung der Fixationen auf AOIs. Im Folgenden nehmen wir an, dass die AOIs schon definiert wurden. Dies wurde entweder manuell mit Hilfe eines AOI-Editors durchgeführt oder mit Hilfe der Anwendung von automatischen Clustering-Algorithmen auf Fixationsebene wie von Duchowski et al. (Duchowski et al., 2010) vorgestellt. Bei der Zuordnung wird überprüft, ob eine Fixation innerhalb einer der gegebenen AOIs liegt, und man weist diese Fixation, falls das Ergebnis positiv ist, der AOI zu. Falls die Fixation nicht innerhalb einer AOI liegt, wird sie der „no hit“-Gruppe hinzugefügt, um sie später gesondert darstellen zu können. Diese Zuordnung von Fixationen zu AOIs wird nur einmal pro Datensatz zu Beginn der Datenanalyse durchgeführt. Das Ergebnis wird aus Performanzgründen in einer SQL-Datenbank gespeichert. Diese Vorprozessierung der Daten erlaubt ein schnelles, späteres Clustering und eine

schnelle Visualisierung auch von großen Eye-Tracking-Datensätzen. Da die Eye-Tracking-Daten im Vorfeld des Clusterings und der Visualisierung vorbereitet wurden, ist es ohne Weiteres möglich, mehrere hunderttausend Fixationen in Echtzeit interaktiv in der Visualisierung zu analysieren. Wenn sich die Definition der AOIs geändert hat, muss allerdings dieser erste Schritt erneut durchgeführt werden.

Filterung/Clustering

Basierend auf den vorbereiteten Fixationsdaten innerhalb von AOIs und durch die vom Nutzer vorgegebenen Filterparameter werden im zweiten Schritt Fixationsdaten für die Analyse ausgewählt. Beispielsweise können Fixationspfade von verschiedenen Probandengruppen für die spätere Analyse ausgewählt werden. Dazu kann nach Geschlecht, Alter oder Vorkenntnissen gefiltert werden.

Für den Vergleich von Fixationspfaden wird jede AOI mit einem Symbol α aus einem Satz ν von Symbolen repräsentiert. Alle Symbole bilden ein Alphabet Σ . Im Folgenden werden für die Veranschaulichung der Ähnlichkeitsberechnung lateinische Buchstaben verwendet. Die Fixationspfade werden auf eine Zeichenkette abgebildet, bei der das erste Zeichen die erste Fixation des Fixationspfads darstellt, das zweite die zweite Fixation und so fort. Als nächstes werden diese Zeichenketten komprimiert, indem die Abfolge von gleichen Symbolen durch ein Symbol aus diesen gleichen Symbolen ersetzt wird. Zum Beispiel wird die Zeichenkette „ABCABCCCAD“ zu „ABCABCAD“ verkürzt.

Den AOIs aus Abbildung 4.2b) werden folgende Symbole zugewiesen:

AOI1 \rightarrow A	AOI3 \rightarrow C	AOI5 \rightarrow E
AOI2 \rightarrow B	AOI4 \rightarrow D	AOI6 \rightarrow F

Für die in den Abbildungen dargestellten Fixationspfade ergeben sich dann die Zeichenketten:

- Abbildung 4.2a) \rightarrow ADBECD
- Abbildung 4.9a) \rightarrow ACFBE
- Abbildung 4.9b) \rightarrow AADEBBCD
- Abbildung 4.9b) \rightarrow ADEBCD

Sowohl nicht gekürzte als auch verkürzte Zeichenketten werden für ein hierarchisches agglomeratives Clustering verwendet. Die Vorgehensweise

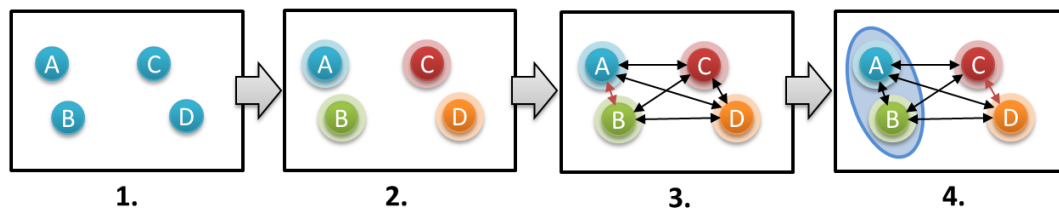


Abbildung 4.12: Hierarchisches agglomeratives Clustering zur Identifizierung von Gruppen ähnlicher Fixationspfade: 1.) Alle Fixationspfade werden dem Clustering-Algorithmus hinzugefügt 2.) und als Zeichenketten in einzelne Gruppen einsortiert. 3.) Die Levenshtein-Distanz unter allen Zeichenketten über alle Clustern hinweg wird berechnet. 4.) Alle Fixationspfade, die unter dem vorgegebenen Ähnlichkeitswert liegen, werden einem gleichen Cluster hinzugefügt.

ist dabei in Abbildung 4.12 skizziert. Zunächst werden alle Fixationspfade als einzelne Scan-Paths hinzugefügt (Schritt 1) und als eigene Cluster identifiziert (Schritt 2). Dann wird die Levenshtein-Distanz unter allen Zeichenketten über alle Cluster berechnet (Schritt 3) und die zwei Cluster mit der kleinsten Levenshtein-Distanz werden zusammengefügt. Dabei kann die Länge der Fixationspfade begrenzt werden, die für die Berechnung der Levenshtein-Distanz verwendet werden. Die Berechnung der Levenshtein-Distanz kann entweder vom Anfang der Zeichenketten aus erfolgen oder Abschnitte aus den Fixationspfaden verwenden. Dann werden Cluster, die über dem vorgegebenen Ähnlichkeitswert liegen, in weitere Cluster zusammengefügt (Schritt 4). Zur Vereinigung der Gruppen wird das „Single-Linkage“-Kriterium verwendet. Dies bedeutet, dass die zwei Elemente aus den zwei Clustern mit der minimalen Distanz jeweils für die Berechnung der Gesamtdistanz der Cluster berücksichtigt werden. Die Normierung der Ähnlichkeitswerte in einem Intervall zwischen 0.0 und 1.0 erfolgt zu Beginn des hierarchischen agglomerativen Clusterings durch die Division aller einzelnen Distanzen zwischen den einzelnen Elementen durch die größte vorliegende Distanz zwischen zwei Elementen. Der Algorithmus terminiert, wenn eine vorgegebene Unähnlichkeit der Cluster erreicht wird oder eine Mindestanzahl von Clustern erkannt wurde. Die berechneten Cluster werden als Visualisierungsdaten an die graphische Ausgabe weitergegeben.

Graphische Ausgabe

Abschließend werden die Fixationspfade mit einer PSP-Visualisierung ausgegeben. Fixationspfade der selben Gruppe werden in einem ähnlichen

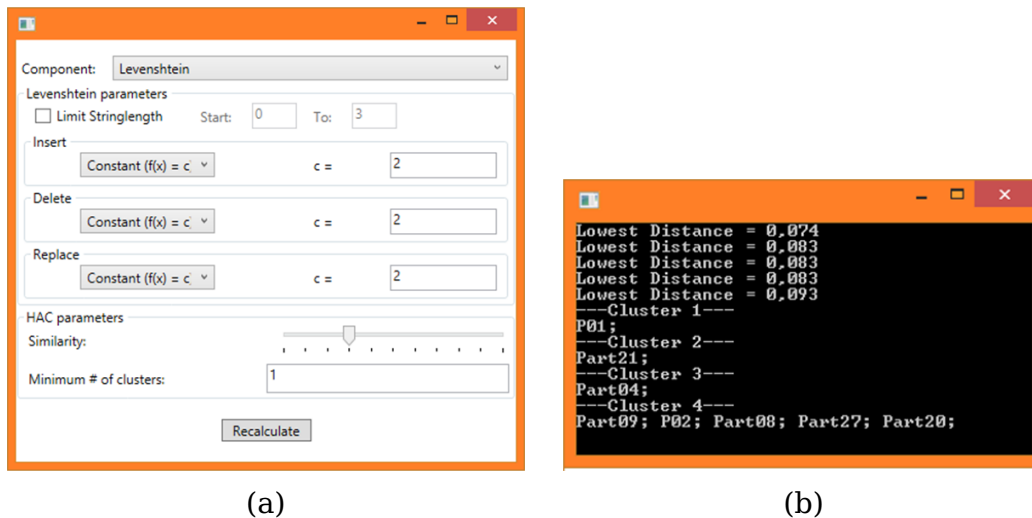


Abbildung 4.13: a) Parameter des Clusterings können in einem separaten Dialog eingestellt werden. b) Ergebnisse der Cluster-Berechnung werden in einer Konsole ausgegeben.

Farbton eingefärbt. Fixationspfade in nicht gleichen Gruppen werden in maximal unterschiedlichen Farben visualisiert.

Änderung der Parameter

Die Parameter des Clusterings können in einem separaten Dialog (Abbildung 4.13a) eingestellt werden. In diesem Dialog wird die Länge der Zeichenketten eingestellt, die pro Fixationspfad miteinander verglichen werden sollen. Falls das Häkchen vor „Limit Stringlength“, nicht gesetzt ist, werden alle Symbole der verkürzten Fixationspfade für den Vergleich verwendet. Für die drei Levenshtein-Funktionen „Einfügen“, „Löschen“ und „Ersetzen“ können unterschiedliche Kostenfunktionen ausgewählt werden und Parameter dieser Kostenfunktionen gesetzt werden. Der Ähnlichkeitswert, ab dem Fixationspfade als ähnlich angesehen werden sollen, wird mit einem Schieberegler gewählt, der Werte zwischen 0.0 und 1.0 zulässt. Alternativ kann eine minimale Anzahl von gefundenen Clustern vorgegeben werden. In einer Konsole (Abbildung 4.13b) werden Ergebnisse der Cluster-Berechnung ausgegeben.

4.3 Demonstration der PSP-Visualisierungstechnik

In diesem Abschnitt wird gezeigt, wie die in Abschnitt 4.1 vorgestellten PSP-Visualisierungen für die Untersuchung der Lesestrategie von einfa-

chen Visualisierungen (Abschnitt 4.3.1) und von Balkendiagrammen (Abschnitt 4.3.2) verwendet werden können.

4.3.1 Analyse von einfachen visuellen Aufgaben

Dieser Abschnitt demonstriert die Anwendung der PSP-Visualisierungstechnik anhand exemplarischer Scan-Paths, die speziell für die Demonstration aufgezeichnet wurden. Die Demonstration lässt sich jedoch ohne Weiteres auf Augenbewegungen aus realen Eye-Tracking-Experimenten übertragen, da hier ähnliche Fragestellungen beantwortet werden.

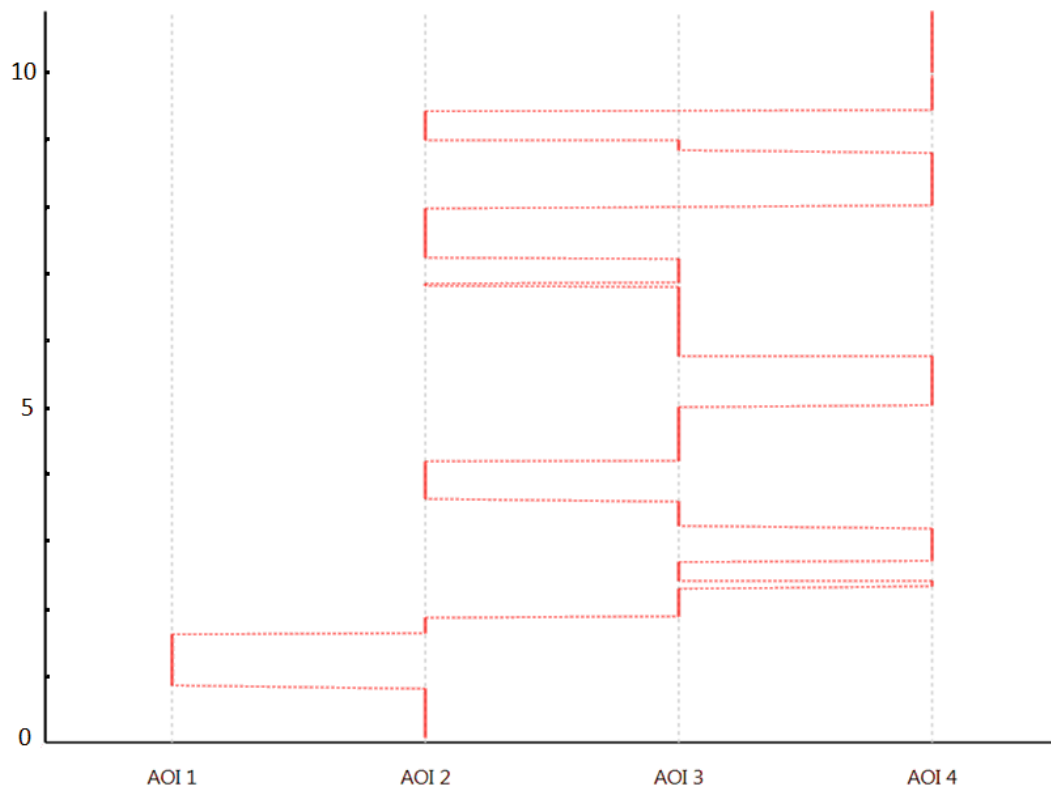
Aufgabe: Lesen eines Werts auf einem Zahlenstrahl

Abbildung 4.14b) zeigt einen Stimulus für die Aufgabe „Lesen Sie bitte den Wert des gegebenen Punktes auf dem Zahlenstrahl ab“. Um die Bewegung der Aufmerksamkeit auf dem Zahlenstrahl näher zu untersuchen, wurden die in Abbildung 4.14b) dargestellten AOIs definiert und es wird ein Verweildauer-Sequenz-Diagramm (Abbildung 4.14a) für die Analyse eines einzelnen Probanden gewählt. Das Verweildauer-Sequenz-Diagramm zeigt hier deutlich die Lesestrategie des Probanden für diese Aufgabe und den gegebenen Stimulus.

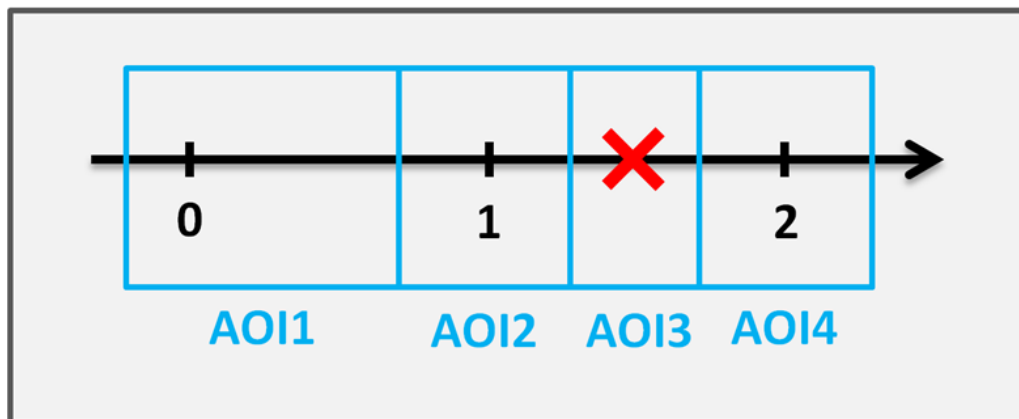
Der Proband benötigte ca. 11 s und wechselte 18 Mal die AOIs. Als erstes gewann er einen Überblick über den Stimulus (erste zwei Sekunden mit einem Wechsel der Aufmerksamkeit zwischen allen AOIs). Dann begann er die exakte Position des markierten Punktes zu lesen. Er wechselte dazu seine Aufmerksamkeit zwischen den beiden Skalenstrichen 1 und 2 und dem gegebenen Punkt. Für die Identifikation des Zahlenwertes benötigte er 9 s und 14 Transitionen. Ein direkter visueller Vergleich der Länge der vertikalen Linien des Fixationspfades zeigt, dass der Proband mehr Aufmerksamkeit darauf verwendete, zwischen dem gegebenen Punkt und dem Skalenstrich 2 hin- und herzuwechseln, als zwischen dem gegebenen Punkt und dem Skalenstrich 1.

Aufgabe: Lesen von zwei Werten in einem Kartesischen Koordinatensystem

Vier Fixationspfade zur Aufgabe „Lesen Sie bitte die Werte der beiden im Diagramm gezeigten Punkte ab.“ sind in Abbildung 4.15a) in einem Fixationspunkt-Diagramm dargestellt. Grundsätzlich wird bei allen Probanden eine wechselnde Aufmerksamkeit zwischen den Koordinatenachsen und den angezeigten Punkten gefunden.



(a)



(b)

Abbildung 4.14: a) Verweildauer-Sequenz-Diagramm eines Probanden. b) Stimulus und AOIs für die Aufgabe „Lesen Sie bitte den Wert des gegebenen Punktes auf dem Zahlenstrahl ab.“.

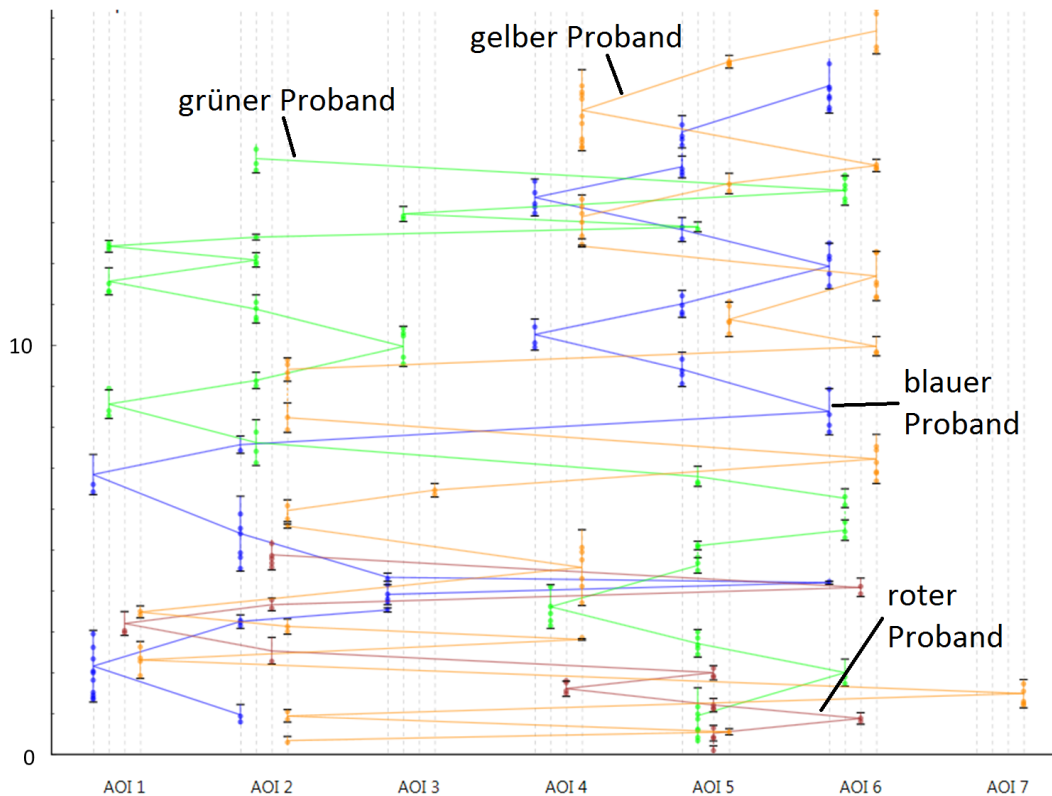
Zur Analyse der Lesestrategien wurden sieben AOIs definiert (siehe Abbildung 4.15b). AOI1 bis AOI3 decken das Leseverhalten von P1 ab, AOI4 bis AOI6 das Lesen der Koordinaten von P2. AOI7 deckt das obere Ende der Y-Achse ab. Die aufgenommenen Scan-Paths der drei Probanden sind in Abbildung 4.15c) visualisiert. Die Analyse der Scan-Path-Visualisierung lässt jedoch kein direktes Erkennen und keinen Vergleich der Lösungsstrategien der vier Probanden zu.

Mit Hilfe des Fixationspunkt-Diagramms erkennt man, dass die vier Probanden ein unterschiedliches Leseverhalten hatten. Der schnellste Proband (grüner Fixationspfad) las zuerst die Position des Punktes P2, dann die des Punktes P1. Dazu benötigte er ca. 13 s. Der blau eingefärbte Proband las die gegebenen Punkte in einer umgekehrten Reihenfolge wie der grün dargestellte Proband und benötigte dazu 17 s. Der mit einem gelben Fixationspfad dargestellte Proband zeigt ein chaotisches Verhalten am Anfang und ein zielstrebiges Lesen der Koordinaten von P2 im zweiten Teil der Aufnahme. Der rot eingefärbte Proband hingegen zeigt ein chaotisches Verhalten während der gesamten Aufgabe. Bei ihm ist keine zielgerichtete Lesestrategie im Fixationspunkt-Diagramm sichtbar. Über alle vier Probanden hinweg erkennt man in der PSP-Visualisierung kurze Verweildauern und wenige Fixationen innerhalb der AOIs.

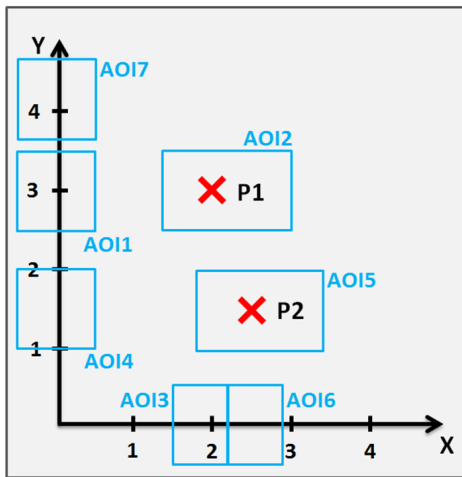
Aufgabe: Vergleich von Dreiecken

Als letztes Beispiel zeigt Abbildung 4.16a) ein Verweildauer-Verteilungs-Diagramm von sechs Probanden für die Aufgabe „Vergleichen Sie bitte die zwei gegebenen Dreiecke und geben Sie an, ob diese mathematisch kongruent sind.“ Die Heat-Map in Abbildung 4.16c) über alle Fixationen der fünf Probanden zeigt, dass die Probanden hauptsächlich die Ecken der Dreiecke betrachtet haben. Aus diesem Grund werden sechs AOIs über die Ecken gelegt (Abbildung 4.16b). Dabei bedecken die AOI1 bis AOI3 jeweils die Ecken des linken Dreiecks und die AOI4 bis AOI6 die Ecken des rechten Dreiecks.

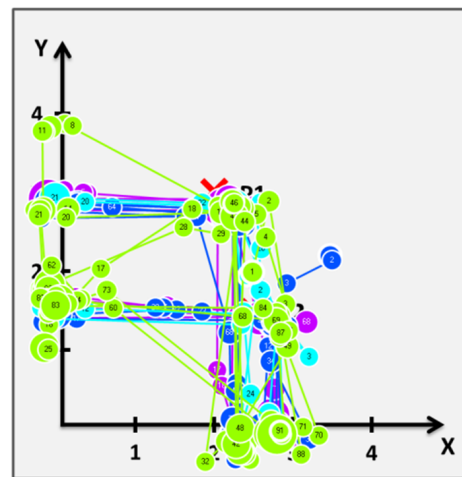
Mit einem Verweildauer-Verteilungs-Diagramm werden die Augenbewegungsstrategien der sechs Probanden miteinander verglichen. Die Probanden benötigten zwischen 3 s (lila eingefärbter Proband) und 14 s (blauer Proband), um die Aufgabe zu lösen. Das im Verweildauer-Verteilungs-Diagramm eingeblendete Balkendiagramm zeigt, dass der grüne Proband dem linken Dreieck mehr Aufmerksamkeit widmete. Dieses Ergebnis zeigt sich auch durch eine niedrigere Transitionsfrequenz zwischen den AOIs 1, 2 und 3 als zwischen den AOIs 4, 5 und 6 für diesen Probanden.



(a)

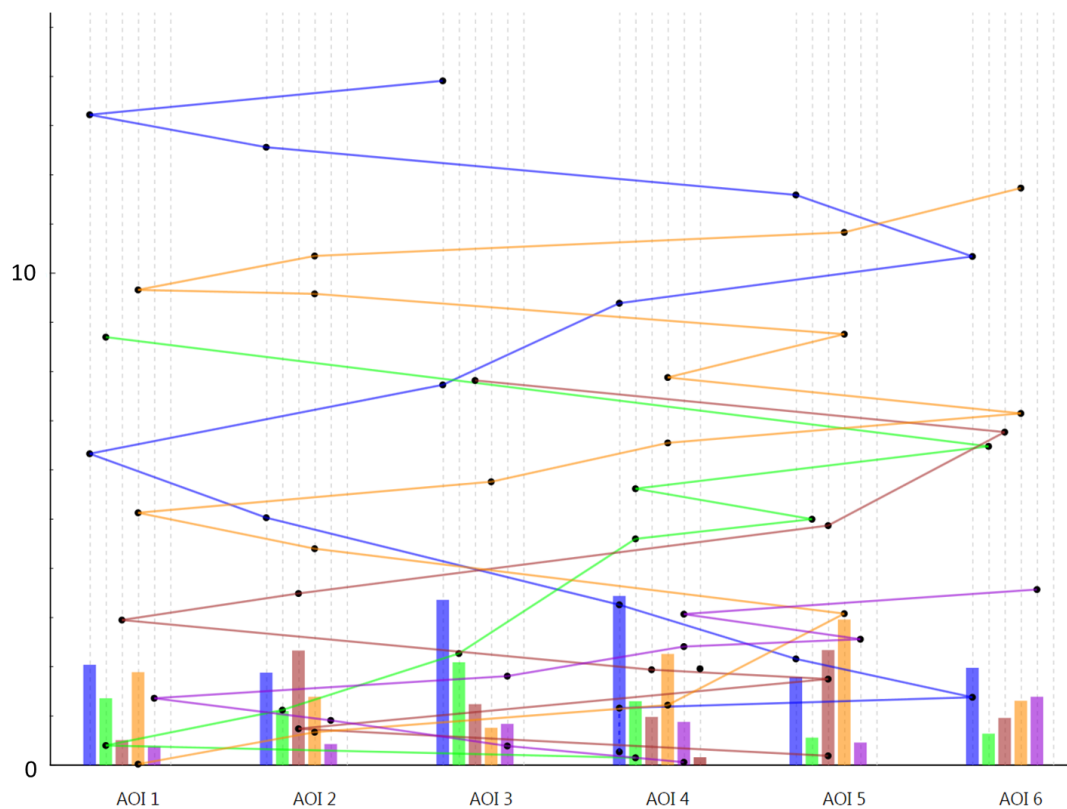


(b)

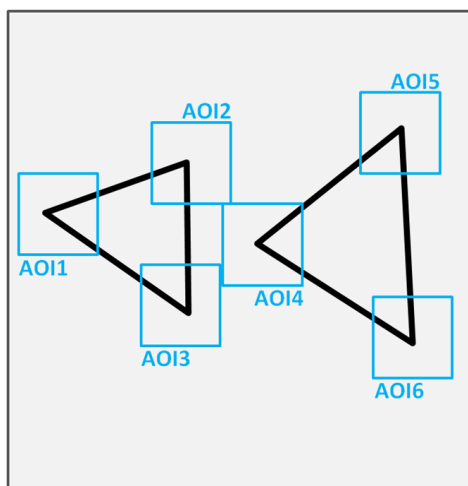


(c)

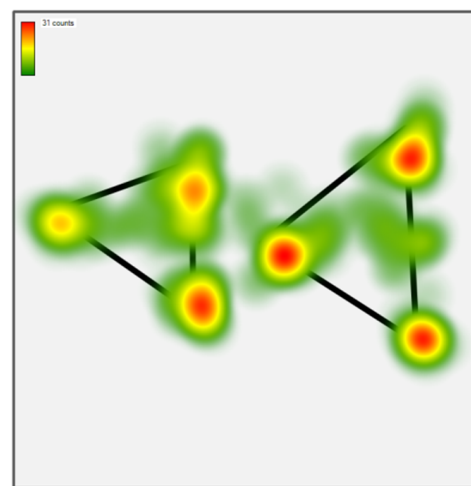
Abbildung 4.15: a) Vier Fixationspfade zur Aufgabe „Lesen Sie bitte die Werte der beiden im Diagramm gezeigten Punkte ab.“ sind in einem Fixationspunkt-Diagramm dargestellt. b) Stimulus zusammen mit AOIs: AOI1 bis AOI3 decken das Leseverhalten von P1 ab, AOI4 bis AOI6 das Lesen der Koordinaten von P2. AOI7 deckt das obere Ende der Y-Achse ab. c) Klassische Scan-Path-Visualisierungen. Diese lässt keinen detaillierten Vergleich der Lesestrategien zu.



(a)



(b)



(c)

Abbildung 4.16: a) Verweildauer-Verteilungs-Diagramm von sechs Probanden für die Aufgabe „Vergleichen Sie bitte die zwei gegebenen Dreiecke und geben Sie an, ob diese mathematisch kongruent sind.“ b) AOIs c) Heat-Map der Fixationen aller Probanden.

Grundsätzlich können mit der PSP-Visualisierung vier unterschiedliche Lösungsstrategien für diese Aufgabe erkannt werden:

1. Erst werden alle Ecken des einen Dreiecks fokussiert, anschließend alle Ecken des anderen Dreiecks (lila, grün und blau eingefärbter Proband).
2. Jede Ecke wird mit jeder Ecke des anderen Dreiecks verglichen (braun dargestellte Proband).
3. Eine Kombination von der ersten und der zweiten Strategie wird verwendet, um die Aufgabe zu lösen (gelber Proband).
4. Der sechste Proband konzentrierte sich auf die Kanten der Dreiecke, um die Dreiecke miteinander zu vergleichen. Da die Kanten der Dreiecke jedoch nicht mit AOIs abgedeckt sind, wird kein Fixationspfad dieses Probanden in der PSP-Visualisierung dargestellt. Die Fixationen des sechsten Probanden sind aber in der Heat-Map auf den Schenkeln des Dreiecks zu sehen.

4.3.2 Lesestrategien beim Lesen eines Balkendiagramms

Im Folgenden werden die Augenbewegungen von sieben Probanden aus einer Pilotstudie mit zehn Teilnehmern untersucht. Das Ziel der Pilotstudie, die in Kapitel 5.4.2 im Detail beschrieben wird, war es, den experimentellen Aufbau und die Durchführung eines größeren Eye-Tracking-Experiments mit 40 Probanden (siehe Abschnitt 6.2.4) zu testen. In diesem Abschnitt wird anhand eines Stimulus und einer Aufgabe demonstriert, wie die entwickelte Methode zum Vergleich von Scan-Paths sowie zum Finden von ähnlichen Augenbewegungen verwendet wird. Der Stimulus ist in Abbildung 4.17a) dargestellt. Die Probanden sollten die Aufgabe „Geben Sie bitte den Wert des höheren Werts der Balken A und D an.“ durchführen. Das Experiment hatte den in Abschnitt 2.4.1 vorgestellten Aufbau und wurde wie dort beschrieben durchgeführt.

Zum Vergleich der Scan-Paths wurden die in Abbildung 4.17b) gezeigten AOIs definiert. Jeder Balken im Balkendiagramm wurde mit einer eigenen AOI umrandet (Bar E, Bar A, Bar C, Bar B, Bar D), ebenso die Y-Achse des Balkendiagramms (Y-Axis). Die Beschriftungen der X-Achse wurden ebenfalls mit AOIs gekennzeichnet (Bar E Label, Bar A Label, Bar B+C Label, Bar D Label). Dabei wurden die Beschriftungen der Balken C und B in einer AOI zusammengefasst, da eine Unterteilung der beiden Balken für die Analyse der Aufgabendurchführung nicht wichtig war.

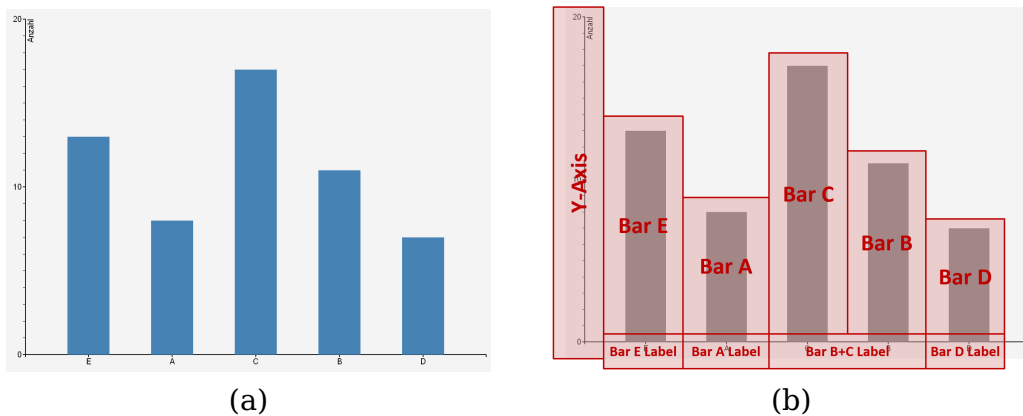


Abbildung 4.17: In einer Benutzerstudie sollten Probanden den höheren Wert der beiden Balken „A“ und „D“ angeben. a) zeigt den Stimulus b) AOIs für die PSP-Visualisierung.

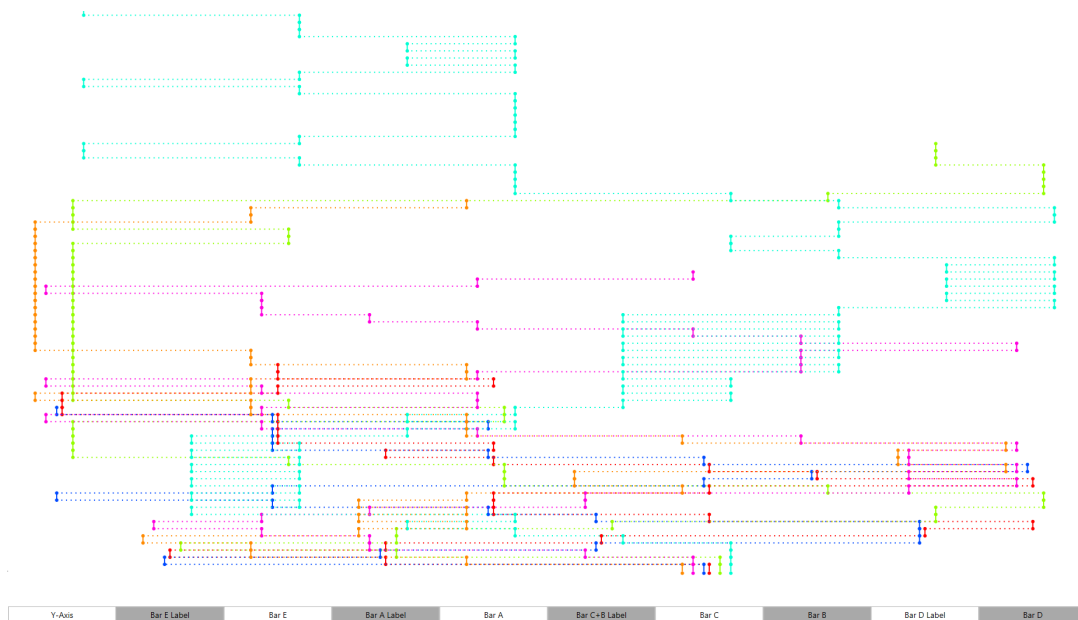


Abbildung 4.18: PSP-Visualisierung von sieben Probanden zur Aufgabe „Geben Sie bitte den Wert des höheren Wertes der Balken „A“ und „D“ an.“. Ein direktes Finden von ähnlichen Augenbewegungsmustern ist nicht möglich.

Abbildung 4.18 zeigt die PSP-Visualisierung von allen sieben Probanden der Pilotstudie bei einem Ähnlichkeitswert von 1.0. Keine ähnlichen Fixationspfade wurden gefunden, da sich alle Fixationspfade in der räumlichen Reihenfolge komplett unterscheiden. In der PSP-Visualisierung ist zu erkennen, dass alle Probanden das Lesen des Balkendiagramms mit dem Balken

„C“ begonnen haben. Die Ursache dafür ist, dass jeder Proband das Kreuz in der Mitte des Bildschirms fokussiert hat und beim Wechsel der Darstellung an dieser Stelle auf dem Stimulus verblieben ist, welche in die AOI „Bar C“ fällt. Ausgehend von dieser AOI begannen die Probanden dann, mit Hilfe unterschiedlicher Augenbewegungsstrategien, die Aufgabe zu lösen. Alle AOIs wurden dabei mindestens von einem Probanden betrachtet.

Wird der Ähnlichkeitswert jedoch schrittweise reduziert, können bei einem Ähnlichkeitswert von 0.73 zwei Gruppen von Fixationspfaden erkannt werden. Die Fixationspfade dieser beiden Gruppen können dann auf einem zeitunabhängigen Gitter visualisiert werden, siehe Abbildung 4.19. Die beiden rot eingefärbten Fixationspfade zeigen Probanden, die zuerst den Wert des Balkens „D“ gelesen haben und anschließend den Wert des Balkens „A“. Die blaue Gruppe an Fixationspfaden zeigt ein inverses Verhalten. Die Repräsentanten aus beiden Fixationspfadgruppen zeigen dieses Verhalten ebenfalls, siehe Abbildung 4.20.

4.4 Zusammenfassung

Dieses Kapitel stellte die PSP-Visualisierung vor, die in Kombination mit existierenden Visualisierungstechniken für die Analyse von Eye-Tracking-Daten eingesetzt werden kann. Die PSP-Visualisierungstechnik besitzt die besondere Eigenschaft, dass sie eine größere Anzahl von Fixationspfaden übersichtlich darstellt. Mit ihr können Lesestrategien der Probanden erkannt werden.

Wie auch die klassische Scan-Path-Visualisierung tendiert die PSP-Visualisierung zu Visual Clutter (Rosenholtz et al., 2005), wenn auch erst bei einer größeren Anzahl von dargestellten Blickpfaden. Um dieses Problem zu beheben, wurde die PSP-Visualisierung als Visualisierungskomponente in einem visuellen Analytik-Ansatz verwendet. Durch die Anwendung von String-Editing-Verfahren erhöht sich jedoch auch der Anspruch an den Benutzer der Visualisierung, da dieser für die Einschätzung der Ähnlichkeit des Blickverhaltens Wissen über den verwendeten String-Editing-Algorithmus besitzen sollte.

Um die PSP-Visualisierung mit klassischen Visualisierungstechniken, wie Scan-Path-Visualisierung und Heat-Map-Darstellung, vergleichen zu können, zeigt Abbildung 4.21 ein Entscheidungsdiagramm, das zur Auswahl einer geeigneten Methode zur Analyse eines Eye-Tracking-Experiments verwendet werden kann. Neben Visualisierungstechniken sind im Diagramm zusätzlich statistische Methoden aufgeführt.

Für einen generellen Überblick über die aufgenommenen Augenbewegungen ist eine Heat-Map-Darstellung besonders geeignet. Falls keine Vi-

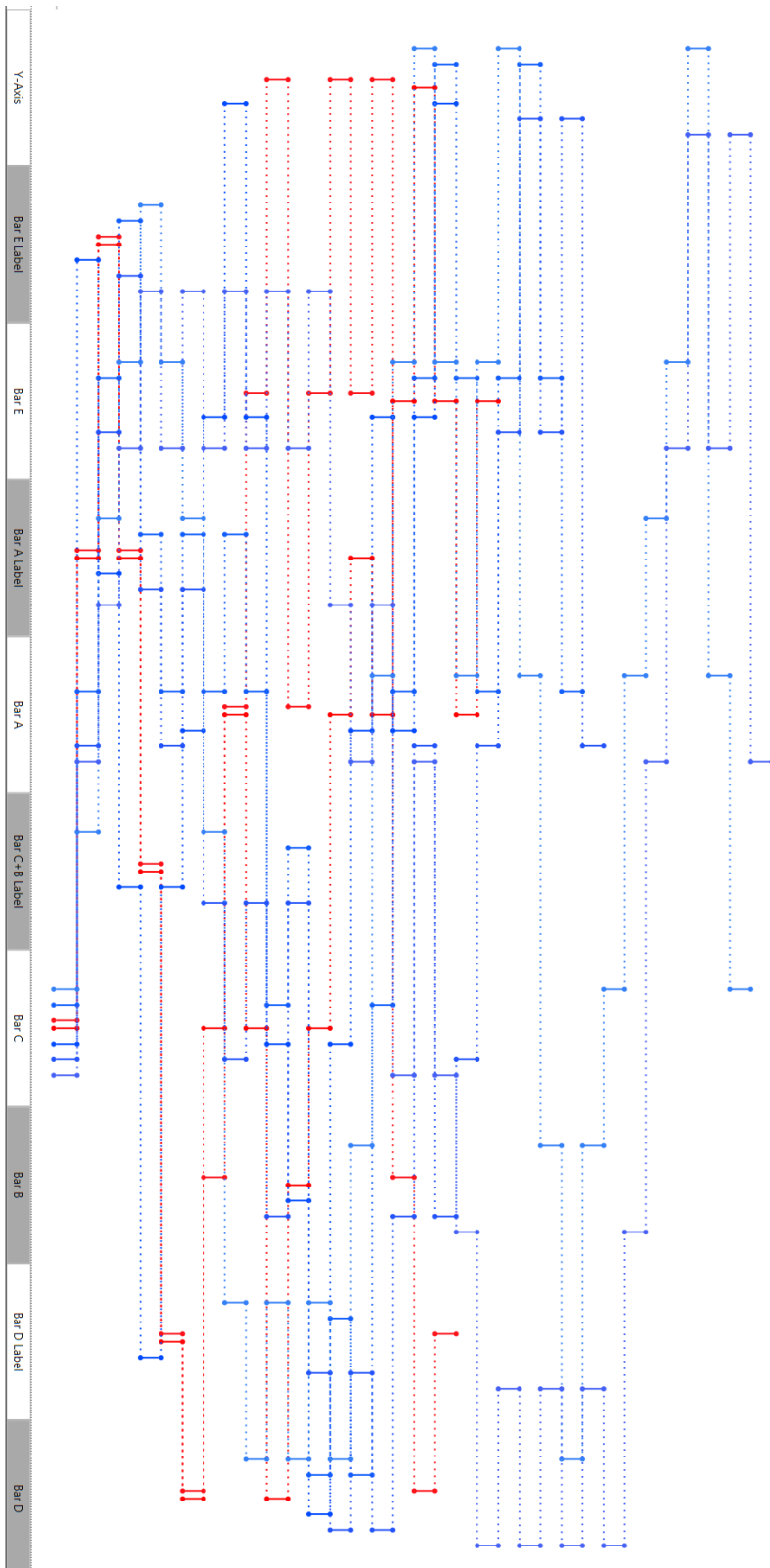


Abbildung 4.19: Bei einem Ähnlichkeitswert von 0.73 werden zwei Gruppen von ähnlichen Fixationspfaden erkannt.

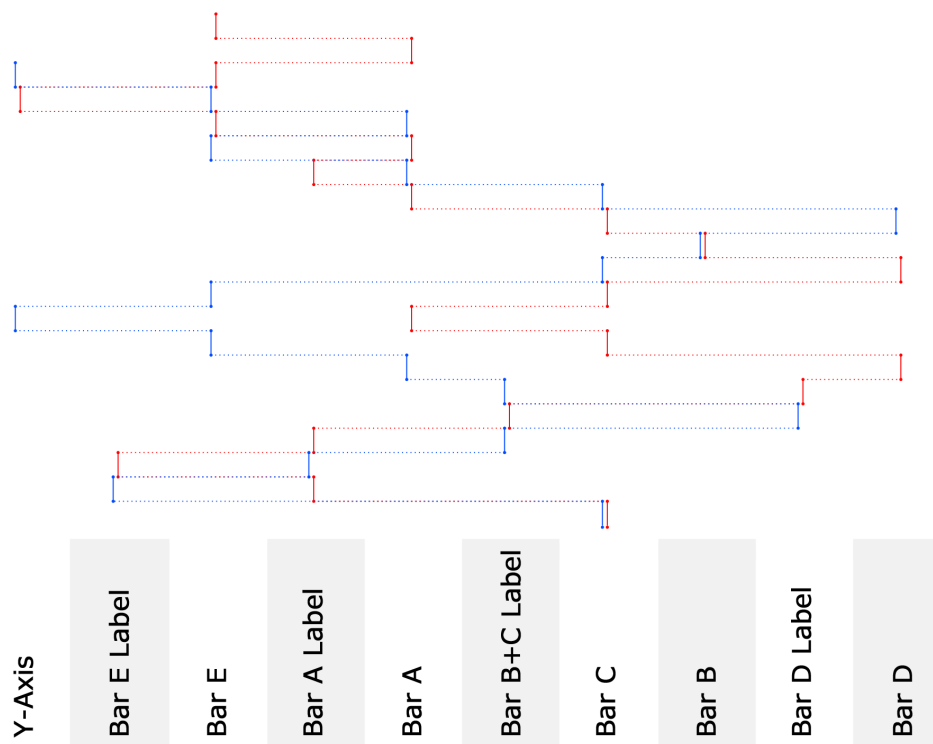


Abbildung 4.20: Auf Basis der beiden erkannten Gruppen von ähnlichen Fixationspfaden können zwei Repräsentanten aus diesen beiden Gruppen ausgewählt werden: Probanden, die zuerst den Wert des Balkens „D“ gelesen haben und anschließend den Wert des Balkens „A“ und Probanden, die eine umgekehrte Reihenfolge gewählt haben.

sualisierungstechniken eingesetzt werden sollen, können die Eye-Tracking-Daten auch rein statistisch untersucht werden und die Ergebnisse der Berechnungen werden entweder numerisch oder in Form von einfachen Diagrammen wie Balkendiagrammen oder Box-Plots ausgegeben. Ein genereller Überblick über das Blickverhalten ist zwar mit der PSP-Visualisierung grundsätzlich auch möglich, jedoch bietet die PSP-Visualisierung nur einen sehr eingeschränkten Übersichtscharakter, im Gegensatz zu einer Heat-Map, aufgrund der Reduktion des Scan-Paths auf AOIs.

Wird die PSP-Visualisierung mit der klassischen Scan-Path-Visualisierung verglichen, zeigen die verschiedenen Anwendungsbeispiele, dass Visual Clutter bei der PSP-Visualisierung erst bei größeren Probandenzahlen auftritt. Durch eine geeignete Anwendung von Interaktionstechniken, wie das Ein- und Ausblenden des Blickverhaltens von Probanden, können, wie bei einer klassischen Scan-Path-Visualisierung, Gruppen von Probanden mit einem ähnlichen Blickverhalten gefunden werden.

Einen deutlichen Vorteil besitzt die PSP-Visualisierungstechnik bei der

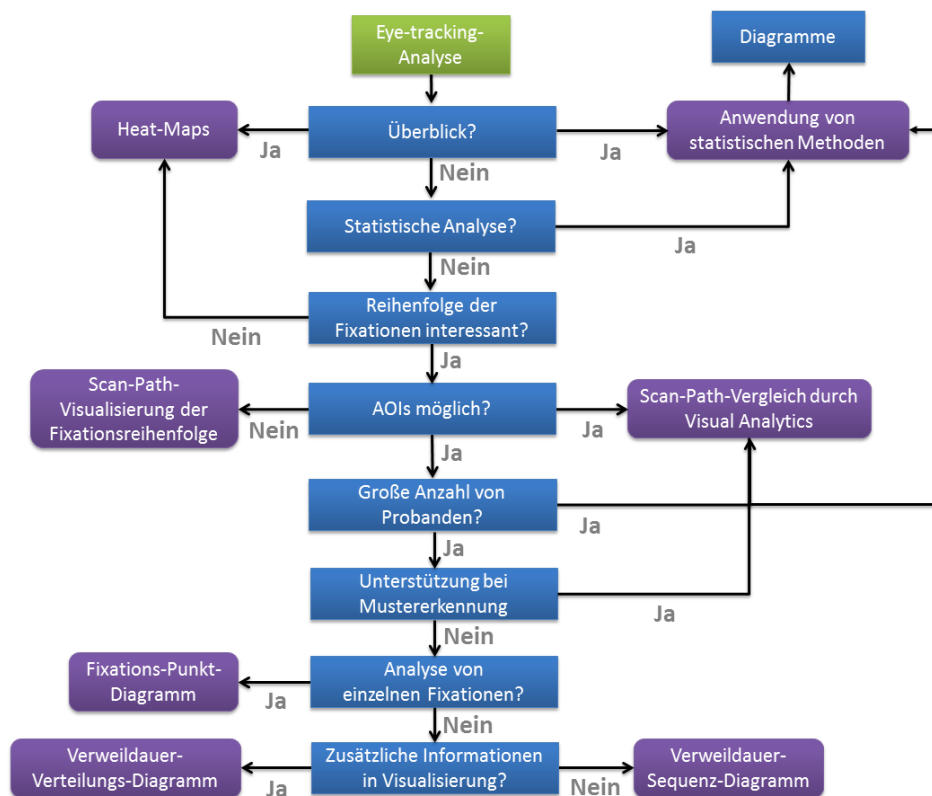


Abbildung 4.21: Überblick über die in diesem Abschnitt präsentierten visualisierungsbasierten Analysetechniken für Eye-Tracking-Daten zusammen mit klassischer Scan-Path-Visualisierung und Heat-Map-Visualisierung. Dieser Plan soll dem Analysten bei der Auswahl einer geeigneten Visualisierungstechnik helfen.

Anwendung von AOIs während der Analyse. Durch die Abstraktion des Scan-Paths zu einem Fixationspfad, kann eine deutlich gesteigerte Übersichtlichkeit der Visualisierung erreicht werden. Da die PSP-Visualisierung zusätzlich jedoch wichtige Eye-Tracking-Metriken wie Verweildauern, Transitionen und Transitionsfrequenzen abbildet, kann die PSP-Visualisierung für eine Vielzahl von verschiedenen Anwendungsfällen eingesetzt werden. Der Analyst muss dazu nicht die Visualisierungstechnik bei unterschiedlichen Fragestellungen wechseln, sondern kann im Idealfall alle wichtigen Studienfragen mit der gleichen Visualisierungstechnik beantworten.

Hierzu stehen verschiedene Varianten der PSP-Visualisierung zur Verfügung. Ein Fixationspunkt-Diagramm kann eingesetzt werden, wenn das Interesse an einzelnen Fixationen auf dem Scan-Path besteht. Ist eine Darstellung der Fixationen während der Analyse nicht notwendig, so können wahlweise das Verweildauer-Sequenz-Diagramm oder das Verweildauer-Verteilungs-Diagramm verwendet werden.

Visuelle Analyse der Wissensverarbeitung

Mit Scan-Paths kann analysiert werden, wie eine visuelle Aufgabe Schritt für Schritt durchgeführt wurde (Koerner, 2011; Rayner, 1998). Damit können Scan-Paths verwendet werden, um Verstehensprozesse zu identifizieren oder diese miteinander zu vergleichen (Grant und Spivey, 2002). Diese Annahme basiert auf der Augen-Gedanken-Hypothese (Eye-Mind-Hypothesis), die besagt, dass das, was wir betrachten, maßgeblich auch die kognitiven Prozesse beeinflusst, die dabei ausgeführt werden (Kapitel 2.7.1). Diese grundlegende Annahme stellt zugleich aber auch die Einschränkung der hier vorgestellten Ansätze und Techniken dar. Die Einschränkungen werden in diesem Kapitel in einem separaten Abschnitt im Detail diskutiert.

Um kognitive und perzeptive Prozesse zu analysieren, ist die semantische Bedeutung der Bereiche, die auf einem Stimulus betrachtet werden, von großer Bedeutung. Eine Methode zur visuellen Analyse von Wissensverarbeitungsprozessen, welche die semantische Bedeutung von graphischen Elementen direkt während der Analyse verwendet, wurde in der Visualisierungsforschung bisher noch nicht vorgestellt. Mit der hier präsentierten Methode können nicht nur die räumlichen Eigenschaften (WO-Raum) von Scan-Paths untersucht werden (siehe dazu Abschnitt 4), sondern es können auch semantischen Eigenschaften der Augenbewegungen (WAS-Raum) analysiert werden. Der Übergang vom WO-Raum in den WAS-Raum wird in dieser Arbeit als *WAS-Raumübergang* oder *inverser WAS-Raumübergang* bezeichnet und wird mit Hilfe der Annotation von graphischen Elementen mit semantischer Information erreicht (Abbildung 5.1).

Die Methode verwendet dazu ein Wissensmodell über graphische Elemente in Visualisierungen (Abschnitt 5.2), ein Verfahren zur automatischen Annotierung von graphischen Elementen mit semantischer Information (Abschnitt 5.3) und eine Erweiterung der PSP-Visualisierungstechnik zur visuellen Analyse der Wissensverarbeitungsprozessen (Abschnitt 5.4). Mit Hilfe der visuellen Analyse der Wissensverarbeitungsprozesse können Leseregeln von Visualisierungen formuliert werden (Abschnitt 5.5), die für die

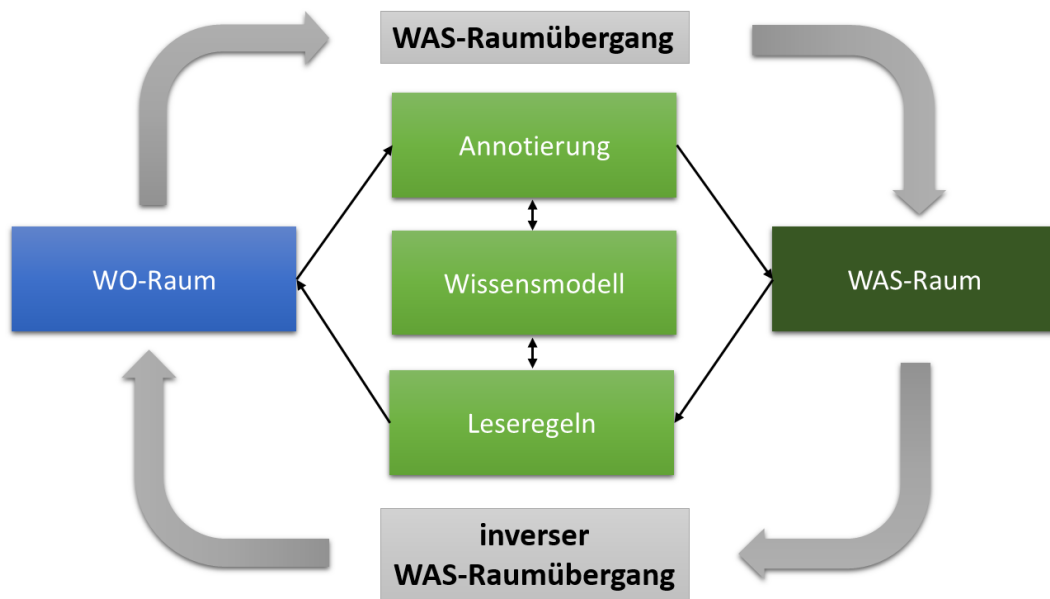


Abbildung 5.1: Durch die Annotierung von Bereichen in Visualisierung mit semantischer Information aus einem Wissensmodell können nicht nur räumliche Eigenschaften (WO-Raum) von Scan-Paths untersucht werden, sondern auch ihre semantischen Eigenschaften (WAS-Raum). Diese Möglichkeit wird in dieser Arbeit durch den Begriff *WAS-Raumübergang* oder *inverser WAS-Raumübergang* bezeichnet.

Simulation der visuellen Suche verwendet werden (Kapitel 6). Das Kapitel schließt mit einer Diskussion der vorgestellten Methode (Abschnitt 5.6).

5.1 Konzept der visuellen Analyse von Wissensverarbeitungsprozessen

Die in diesem Kapitel vorgestellte Methode zur visuellen Analyse von Wissensverarbeitungsprozessen basiert auf den folgenden Annahmen und Methoden aus Eye-Tracking-Experimenten und der Literatur:

1. Es gilt die Augen-Gedanken-Hypothese (Kapitel 2.7.1).
2. Während der Durchführung einer Aufgabe mit einer gegebenen Visualisierung werden die in einer Visualisierung relevanten Bereiche in einer seriellen Reihenfolge fokussiert. Diese Reihenfolge an Fixationen auf dem Stimulus werden in dieser Arbeit nach (Koerner, 2011; Rayner, 1998; Grant und Spivey, 2002) für die Untersuchung von kognitiven Prozessen verwendet.

3. Für die Analyse der Wahrnehmung einer Visualisierung (Perzeption) und der Verarbeitung der graphisch repräsentierten Information (Kognition) wird Wissen über Visualisierungen, wie in Kapitel 2.7.2 beschrieben in die zwei Klassen *prozedurales Wissen* und *deklaratives Wissen* aufgeteilt.
4. Bei der Informationssuche in einer Visualisierung werden zur visuellen Analyse in dieser Arbeit grundsätzlich zwei Arten von kognitiven Prozessen unterschieden:
 - a) Die visuelle Suche und das Lesen von visuell enkodierter Information, ohne eine Schlussfolgerung dabei zu ziehen. Dabei ist das deklarative Wissen über graphische Elemente von Bedeutung.
 - b) Kognitive Prozesse, bei denen eine Schlussfolgerung gezogen wird. Die syntaktische Beziehung der graphischen Elemente ist von Bedeutung. Dabei spielt sowohl prozedurales als auch deklaratives Wissen eine wichtige Rolle beim Verständnis der graphisch dargestellten Information.

Die Methode geht dann in drei Schritten vor (Abbildung 5.2):

1. Bereiche auf dem Stimulus, die für eine gegebene Aufgabe von Bedeutung sind, werden durch den Analysten identifiziert. Dieser Schritt kann mit Hilfe von geeigneten Visualisierungstechniken, wie beispielsweise mit einer Heat-Map, analysiert werden (Abbildung 5.2 Mitte).
2. Die identifizierten Bereiche auf dem Stimulus werden mit einer semantischen Bedeutung annotiert (Abbildung 5.2 oben und Abschnitt 5.3). Die Annotierung kann entweder mit Hilfe von einfachen Wortlisten durchgeführt werden oder durch das Verknüpfen mit Elementen eines Wissensmodells (Abschnitt 5.2). Eine passende Bezeichnung oder Auswahl von Elementen im Wissensmodell ist dabei sehr wichtig für eine spätere, zielführende Auswertung (Perini, 2005). Durch diese Annotierung wird der WAS-Raumübergang vorbereitet.
3. Im letzten Schritt erfolgt der WAS-Raumübergang durch die Abbildung der Scan-Paths aus dem WO-Raum auf diese semantisch annotierten Bereiche (Abbildung 5.2 unten). Diese *WAS-Raum-Fixationspfade* werden anschließend mit der PSP-Visualisierung visuell analysiert (Abschnitt 5.4).

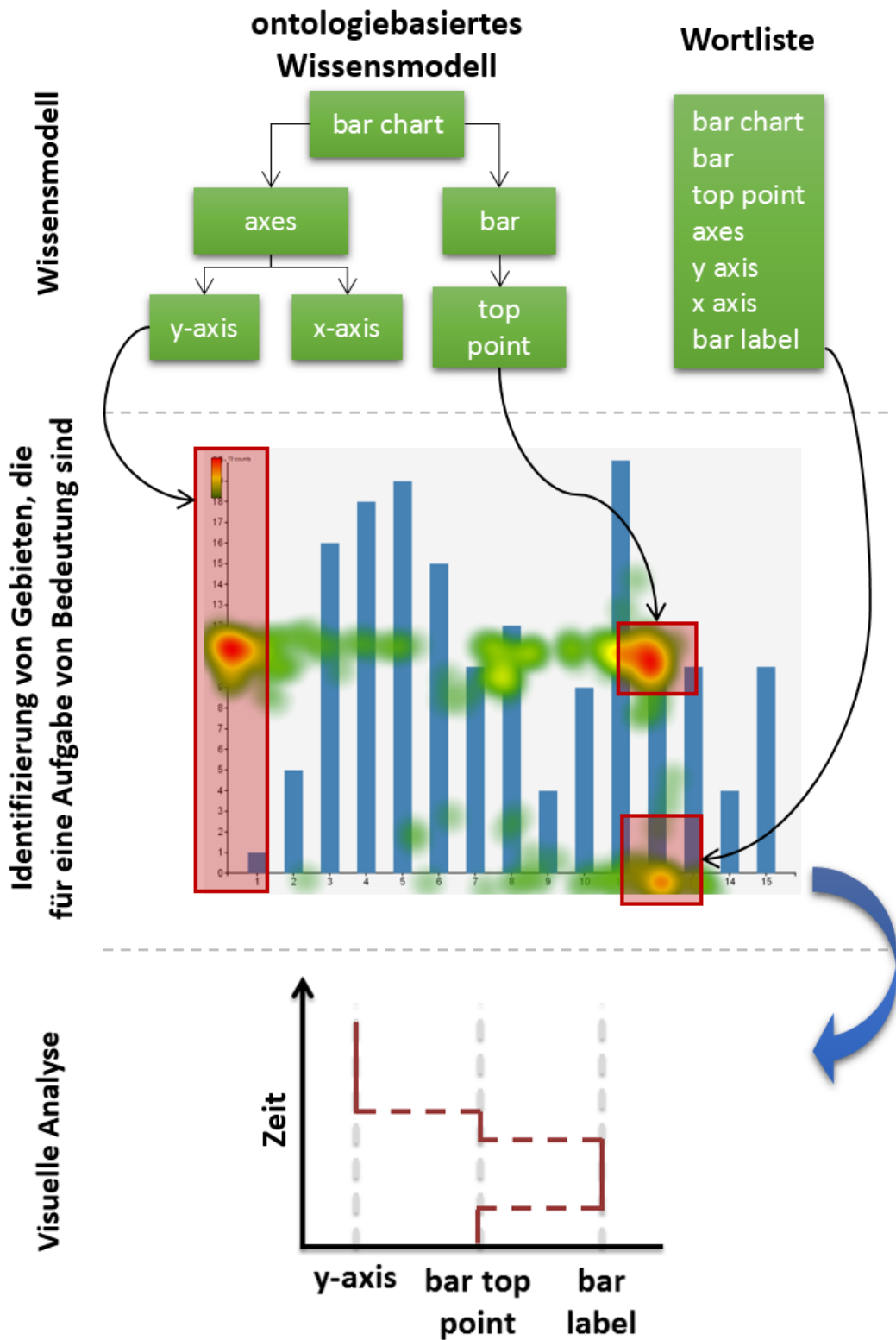


Abbildung 5.2: Bereiche auf dem Stimulus, die für die gegebene Aufgabe von Bedeutung sind, werden durch den Analysten identifiziert (Mitte) und mit semantischer Information aus einem Wissensmodell (oben) annotiert. WAS-Raum-Fixationspfade werden anschließend visuell untersucht (unten).

Basierend auf den Ergebnissen der Analyse können anschließend Lese-regeln für Visualisierungen formuliert werden (Abschnitt 5.5).

5.2 Wissenmodell auf drei Ebenen

Basierend auf der Arbeit von Pinker (Pinker, 1990) wurde für die Ana-lyse von kognitiven und perzeptiven Prozessen ein Wissensmodell mit drei Ebenen entwickelt (siehe auch Abschnitt 2.7.3). Diese drei Ebenen entsprechen unterschiedlichen Abstraktionsebenen von Wissen über Vi-sualisierungen, der graphischen Elemente, welche die Visualisierungen verwenden, sowie den Relationen der graphischen Elemente zueinander.

Die drei Ebenen des Wissensmodells sind (Abbildung 5.3):

1. *die Visualisierungsontologie*
2. *das Visualisierungsschema*
3. *und das Visualisierungsmodell.*

Neben der Abbildung von unterschiedlichen Abstraktionsebenen von Wissen über Visualisierungen nach Pinker ist diese Unterteilung auch tech-nisch von Vorteil, da auf der obersten Ebene die grundsätzlich technisch programmierbaren, primitiven graphischen Elemente einer Visualisierung abgebildet werden wie Pixel, Linien und verschiedene Arten von Polygonen, die dann auf der mittleren Ebene zu Klassen zusammengefasst und auf der untersten Ebene mit Hilfe von Informationen über einen gegeben Stimu-lus instanziiert werden. Abschnitt 5.3.2 zeigt dazu eine Implementierung in RDF/OWL auf Basis von SVG-Graphikelementen (Scalable Vector Gra-pherics) und Kapitel 6.1 die Umsetzung in einer in Lisp und C# entwickelten Kognitionssimulation für die visuelle Suche.

Visualisierungsontologie

Diese Ebene repräsentiert alle graphischen Elemente, die in Visualisierun-gen verwendet werden. Dazu wird die Beziehung zwischen graphischen Elementen über „ist ein/eine“-Eigenschaften und Vererbungsrelationen de-finiert. Beispielsweise muss ein Wissensmodells über Säulendiagramme auf

Teile dieses Kapitels wurden veröffentlicht in:

Raschke, Michael; Engelhardt, Stephan; Ertl, Thomas: A Framework for Simulating Visual Search Strategies. In: *Proceedings of the 11th International Conference on Cognitive Modeling*, 2013.

der Ebene der Visualisierungsontologie mindestens die Elemente Koordinatenachse („coordinate axis“), Linie („line“), Rechteck („rectangle“) und ein Textelement zur Beschriftung besitzen („label“), die alle ein graphisches Element sind (Abbildung 5.3 oben).

Visualisierungsschema

Im Visualisierungsschema werden den graphischen Elementen der Visualisierungsontologie weitere Klassen, Relationen und Literale hinzugefügt. Typisches Merkmal dieser Klassen, Relationen und Literale ist es, dass sie unscharfe Eigenschaften wie z. B. „Element A liegt rechts von Element B“ oder „Element A hat die Farbe grün“ beschreiben. Diese Ebene entspricht dem „Schema“ bei Pinkers Modellen und beschreibt damit die grundsätzlich in einem Visualisierungskonzept verwendeten graphischen Elemente und ihre Anordnung. Somit kann ein Visualisierungsschema als prototypisches Modell eines gegebenen Visualisierungskonzepts gesehen werden. Auf Basis eines gegebenen Visualisierungsschemas können Leseregeln für das beschriebene Visualisierungskonzept formuliert werden. Mit Hilfe der Leseregeln wird es auch möglich, eine Rücktransformation aus dem WAS-Raum in den WO-Raum durchzuführen (*inverser WAS-Raumübergang*). Dazu müssen die graphischen Elemente einer Visualisierung in einer Reihenfolge der Leseregeln, die im Visualisierungsschema vorgegebenen sind, nacheinander fokussiert werden.

Für ein Säulendiagramm gilt zum Beispiel, dass mehrere Rechtecke, die nebeneinander liegen, höhenproportional eine Menge von Werten darstellen. Im Visualisierungsschema werden Elemente, die mehrfach in einer Visualisierung vorkommen, mit einem Stern (*) gekennzeichnet. Dazu müssen die Säulen („bar“) mit einer Höhe („height“) und eine Breite („width“) beschrieben werden, die im Visualisierungsmodell durch die Angabe einer Zahl („number“) instanziiert werden (Abbildung 5.3 Mitte). Das erste Element aus der Menge von Werten wird als erste Säule links dargestellt, dann folgt die Säule für den zweiten Wert und so weiter. Die Reihe von Säulen wird durch ein Koordinatensystem mit einer Y-Achse („y-axis“) und einer X-Achse („x-axis“) eingerahmt, die senkrecht aufeinander stehen („orthogonal“). Die Y-Achse zeigt dabei die Einheiten der Werte der Säulen. Die X-Achse wird mit den Namen der einzelnen Werte beschriftet. Wie viele Werte im Säulendiagramm genau dargestellt werden oder welche genauen Werte die Säulen graphisch abbilden, wird im Visualisierungsschema noch nicht festgelegt.

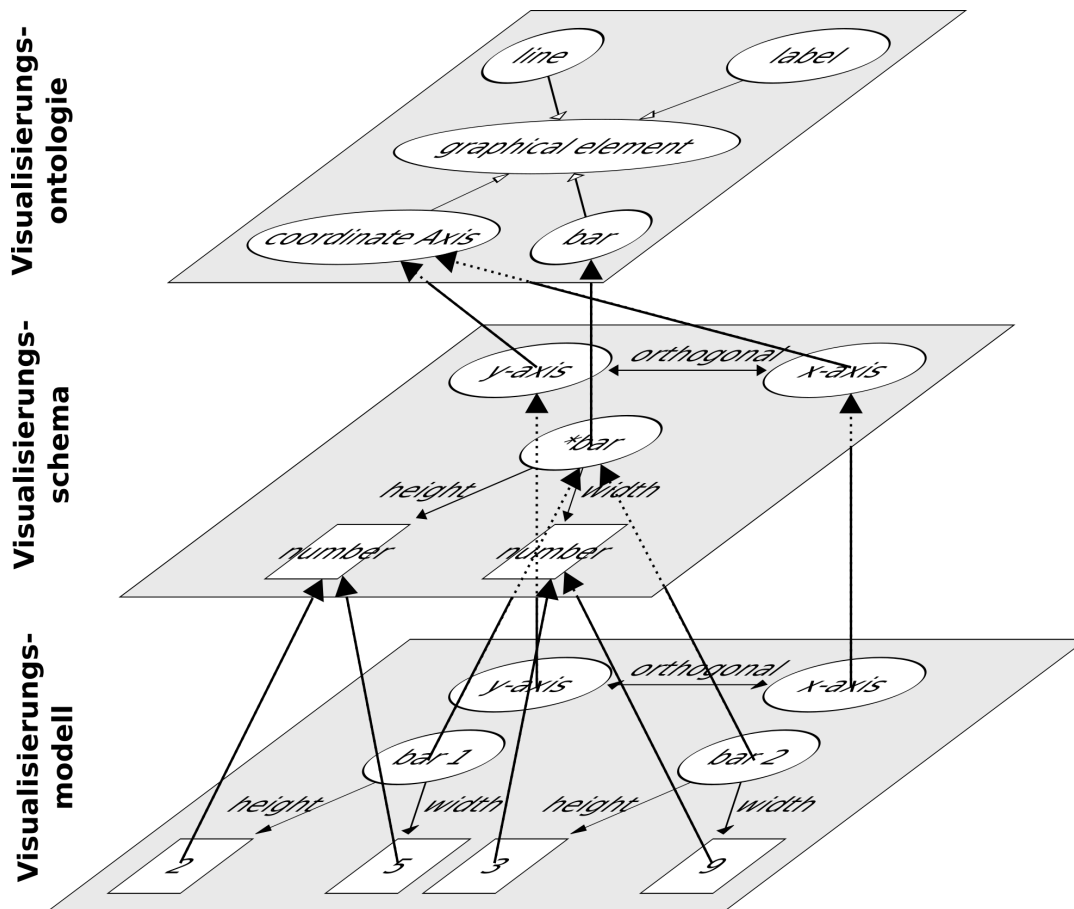


Abbildung 5.3: Wissensmodell über Visualisierungskonzepte und graphische Elemente unterteilt in die drei Ebenen *Visualisierungsontologie*, *Visualisierungsschema* und *Visualisierungsmodell*.

Visualisierungsmodell

Visualisierungsschemata werden in Form von Visualisierungsmodellen instanziiert, um eine vorliegende Visualisierung im Wissensmodell exakt abzubilden. Dazu werden Instanzen der im Visualisierungsschema definierten Klassen und Relationen erzeugt. Eigenschaften zu den einzelnen graphischen Elementen wie exakte Positionen, Textureigenschaften und Farben der graphischen Elemente werden dabei angegeben. Dies kann entweder während des Renderingprozesses automatisch geschehen (Abschnitt 5.3) oder mit Hilfe eines speziell angepassten AOI-Editors (Abbildung 5.4).

Je exakter und umfangreicher die Eigenschaften eines Visualisierungskonzeptes in einem Visualisierungsschema repräsentiert werden, um so detaillierter können kognitive und perzeptive Prozesse untersucht oder in einer Kognitionssimulation simuliert werden.

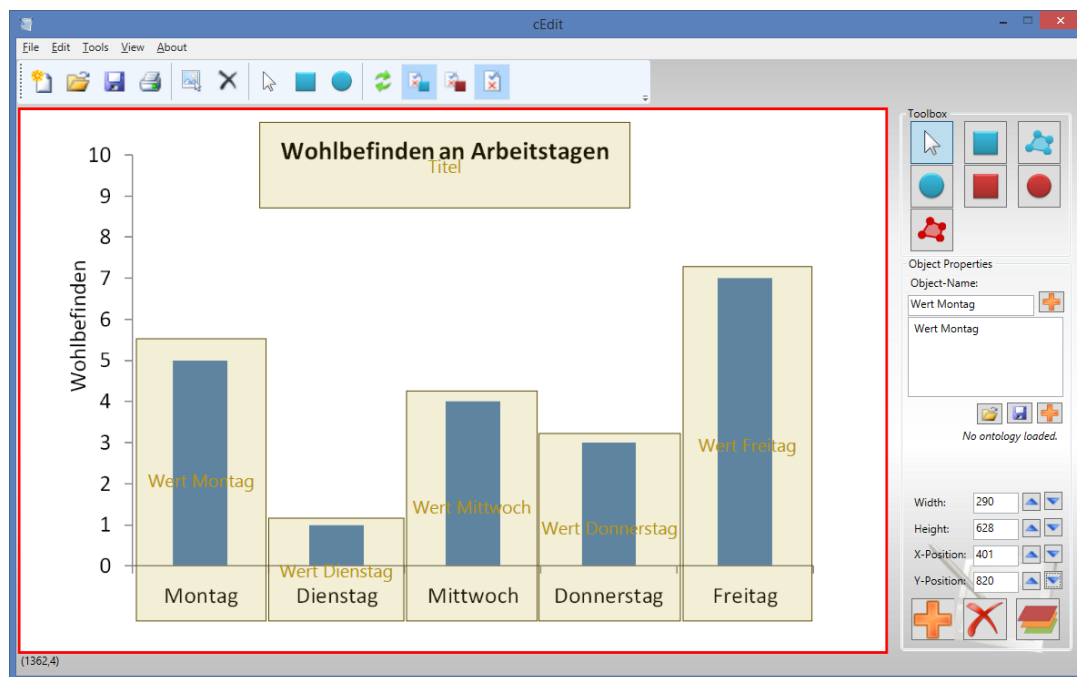


Abbildung 5.4: Normalerweise werden AOIs für die Analyse von Eye-Tracking-Experimenten manuell mit einem AOI-Editor definiert. Dazu werden geometrische Formen auf einem Stimulus gezeichnet und diesen eindeutige Namen zugewiesen (hier „Wert Montag“, „Wert Dienstag“ usw.).

5.3 Automatische Annotation von Visualisierungen

In vielen Fällen werden AOIs für die Analyse von Eye-Tracking-Experimenten manuell definiert. Dazu werden mit einem geeigneten Editor (Abbildung 5.4) Bereiche auf dem Stimulus markiert und diesen eindeutige Bezeichner zugewiesen. Meist sind es einfache Namen wie „AOI1“, „AOI2“ oder Begriffe, welche die im Bereich der AOI sich befindenden graphischen Elemente des Stimulus näher beschreiben, z.B. „Titel“ oder „Wert Montag“, „Wert Dienstag“ und so weiter. Neben frei wählbaren Begriffen können auch Namen aus Wortlisten verwendet werden oder AOIs mit Elementen einer Ontologie verbunden werden (Abbildung 5.2). Die manuelle Annotation hat allerdings den Nachteil, dass die AOIs angepasst werden müssen, wenn sich das Layout einer Visualisierung ändert, da dadurch graphische Elemente in der Visualisierung verschoben oder ihre Form oder Größe verändert wird. Dadurch müssen die Positionen der AOIs und ihre Formen wieder mit

Teile dieses Kapitels wurden veröffentlicht in:

Raschke, Michael; Blascheck, Tanja; Strohmaier, Stefan; Ertl, Thomas: Annotation of Graphical Elements in Visualizations for an Efficient Analysis of Visual Tasks. In: *CHI 2013 Extended Abstracts*, S. 1387-1398, 2014.

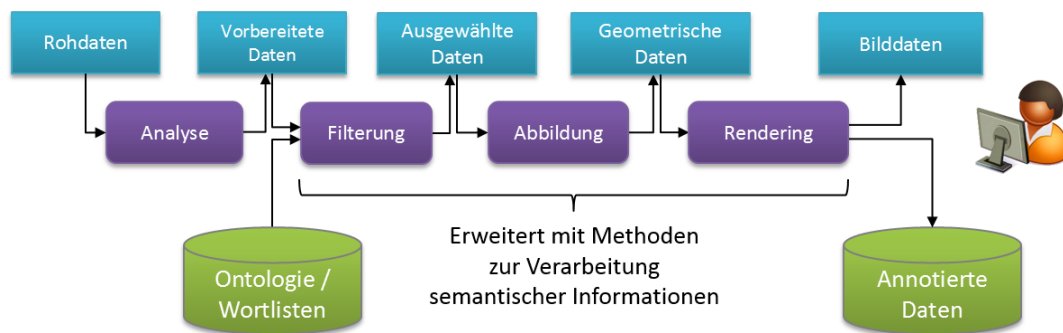


Abbildung 5.5: Semantische Informationen aus einer Ontologie oder aus Wortlisten werden entlang der Visualisierungs-Pipeline mitgeführt, um beim Rendering neben den Bilddaten auch die räumlichen und semantischen Eigenschaften der graphischen Elemente in einer Visualisierung in einer separaten Datei mit auszugeben.

einem AOI-Editor angepasst werden. Da diese manuelle Anpassung zeitaufwändig ist, wird in diesem Abschnitt ein Verfahren vorgestellt, mit dem automatisch während der Generierung einer Visualisierung die Positionen und Formen der AOIs automatisch ermittelt werden. Parallel dazu werden die AOIs semantisch annotiert.

Im Folgenden wird zunächst das Konzept der semantischen Annotation vorgestellt (Abschnitt 5.3.1). Anschließend wird gezeigt, wie das Konzept implementiert wurde (Abschnitt 5.3.2).

5.3.1 Konzept der semantischen Annotation

Die Definition von AOIs ist eine komplexe Aufgabe. Im Gegensatz zu automatischen Clustering-Verfahren, wie von Privitera und Stark (Privitera und Stark, 2000) sowie von Duchowski et al. (Duchowski et al., 2010) vorgestellt, erlaubt das hier vorgestellte Verfahren eine manuelle Einflussnahme, da der Analyst entscheiden kann, welche graphischen Elemente er in einer Visualisierung wie annotieren möchte. Zusätzlich kann er gezielt festlegen, mit welchen semantischen Informationen aus einer Ontologie die graphischen Elemente verbunden werden sollen.

Das Verfahren der automatischen Annotation von graphischen Elementen arbeitet in drei Schritten:

1. Graphische Elemente in einer Visualisierung werden mit Informationen aus einer Ontologie oder aus Wortlisten annotiert.
2. Während der Berechnung der Visualisierung wird die Annotationsinformation entlang der Visualisierungs-Pipeline transportiert (Ab-

bildung 5.5). Im letzten Schritt der Visualisierungs-Pipeline, beim eigentlichen Rendering, wird neben der Visualisierung als Bitmap auch eine Datei mit zusätzlichen Daten zur Position und Größe der annotierten, graphischen Elemente, sowie zu ihrer semantischen Information mit ausgegeben. Siehe dazu auch Abschnitt 2.1.1.

3. Diese zusätzlich ausgegebenen räumlichen und semantischen Informationen werden für die Definition von AOIs verwendet. Dazu werden graphische Elemente in einer Visualisierung zu AOIs zusammengefasst.

5.3.2 Implementierung

Um das entwickelte Konzept zu testen, wurde ein Prototyp implementiert und eine in OWL/XML entwickelte Ontologie graphischer Elemente eingesetzt (Abschnitt 2.7.2). Zur Visualisierung von Daten wird das Visualisierungsframework D3 (Data-Driven Documents)¹ eingesetzt. D3 gibt Visualisierungen in Form von SVG aus. Der gesamte Quellcode des Visualisierungsframeworks ist frei zugänglich. Zusätzlich besteht der Prototyp aus einem „Hit-Test“. Ein JavaScript-Programm simuliert Fixationen auf der Visualisierung in einem Browser mit Hilfe der Emulation von Mausclicks und gibt die semantische Information der graphischen Elemente aus, die virtuell angeklickt worden sind. Die drei Schritte des Verfahrens werden im Folgenden beschrieben.

Ontologie graphischer Elemente

Die graphischen Elemente werden in einer Ontologie beschrieben, die zwei Klassen von Elementen besitzt:

1. **Primitive graphische Elemente**, die in SVG verwendet werden. Diese sind: Pfad, Kreis, Rechteck, Linie, Polygonzug, Text und Bild. Diese Ebene entspricht der Visualisierungsontologie.
2. **Graphische Elemente einer Visualisierung** wie Achsen, Balken, Beschriftungen und Datenpunkte. Diese Ebene entspricht dem Visualisierungsschema.

Die Instanziierung des Visualisierungsschemas in Form eines Visualisierungsmodells erfolgt bei der Erstellung der eigentlichen Visualisierung im

¹<http://d3js.org/>

letzten Schritt der Visualisierungs-Pipeline. Die Ontologie wurde in Protégé 4.3² entwickelt und in OWL/XML gespeichert, um für den D3-Prototyp und den Hit-Test zur Verfügung zu stehen.

Modifizierung von D3

Informationen aus der Ontologie werden in das D3-Framework mit Funktionen aus der jOWL-Bibliothek³ geladen. D3 verwaltet SVG-Primitive und graphische Elemente als separate Klassen. Wenn eine Visualisierung generiert wird, werden die dafür notwendigen Ontologieklassen instanziiert und mit den dazu gehörenden XML-Knoten der SVG-Graphik verbunden. Zusätzliche Informationen werden mit Hilfe von `<semantic>`-Tags an die graphischen Elemente angehängt. Diese `<semantic>`-Tags verweisen auf Eigenschaften aus der Ontologie. Ein Ausschnitt aus einem D3-Programmcode zeigt, wie ein Balken in einem Balkendiagramm als ein Objekt des Typs „Bar“ aus der Ontologie annotiert wurde und mit der Eigenschaft „originDataPoint“ näher spezifiziert wird (Listing 5.1). Der damit erzeugte SVG-Code ist im Listing 5.2 dargestellt. Neben dem eigentlichen Balken, der als Typ „Bar“ aus der Ontologie annotiert wurde, werden in einem `<semantic>`-Tag Informationen zur Eigenschaft des Balkens mitgeliefert. Der `<semantic>`-Tag wird dabei vom Browser beim Rendering der SVG-Graphik ignoriert, kann jedoch für die Generierung einer AOI-Karte weiterverwendet werden.

Listing 5.1: Ein Balken in einem Balkendiagramm wird annotiert.

```
var barSemantics = svg.selectAll(".bar")
    .data(data)
    // Appending a rectangle tag for each data element
    .enter().append("rect")
    // Creating an instance of the OWL class "bar"
    .attr("typeof", jOWL("Bar").URI)
    .attr("x",function (d) { return x(d.name); })
    ...

    // Annotation with additional semantic information
barSemantics.append("semantic") // Nested semantics tag
    // Adding the OWL property attribute "originDataPoint"
    .attr("property", jOWL("originDataPoint").URI)
    .attr("resource", function (d) { return d.name + " " + d.value; });
    ...
```

²<http://protege.stanford.edu/>

³<http://jowl.ontologyonline.org>

Listing 5.2: Ausgabe von D3 in SVG.

```
<!-- Definition of a bar in a bar chart -->
<rect height="120" y="280" width="49" X="50" class="bar" typeof="Bar">
  <!-- Semantic annotation with own property "originDataPoint" -->
  <semantics property="originDataPoint" resource="1 6"></semantics>
  ...
</rect>
```

Erzeugung von AOIs und Hit-Test

Ein „Hit-Test“ in Form eines JavaScript-Programms lädt einen Scan-Path mit seinen einzelnen Fixationen. An jeder Position einer Fixation wird auf der Fläche der SVG-Graphik ein Mausklick emuliert, mit dessen Hilfe die an dieser Position sich befindenden SVG-Elemente ermittelt werden. Da diesen SVG-Elementen auch Informationen zu ihrem Typ und ihren Eigenschaften zugewiesen worden sind, können diese mit ausgegeben werden (Abbildung 5.6).

5.3.3 Diskussion

Eine technische Einschränkung des beschriebenen Verfahrens besteht darin, dass der Prototyp nicht zeigen kann, welches graphische Element von einem Probanden genau betrachtet wurde, wenn eine Fixation auf ein Gebiet fällt, in dem sich mehrere graphische Elemente übereinander befinden. Dieses Problem wurde im Prototypen so gelöst, dass alle sich an dieser Position befindenden graphischen Elemente und ihre semantischen Eigenschaften ausgegeben werden. Falls eine Fixation nicht direkt auf ein graphisches Element fällt, sondern wenige Pixel daneben, gibt der Prototyp kein betrachtetes graphisches Element aus. Dieser Fall kann aber eintreten, falls ein Eye-Tracker unzureichend kalibriert wurde oder die graphischen Elemente sehr geringe Ausdehnungen besitzen, wie es beispielsweise bei Linien der Fall ist. Eine Lösung dieses Problems wäre es, graphische Elemente in einem vorgegebenen Umkreis um die Fixation zu beachten und Informationen zu diesen auszugeben.

Für die Umsetzung des Konzepts in Form eines Prototypen wurde eine einfache Ontologie an graphischen Elementen aufgebaut. Eine umfangreiche Visualisierungsontologie fehlt jedoch bisher.

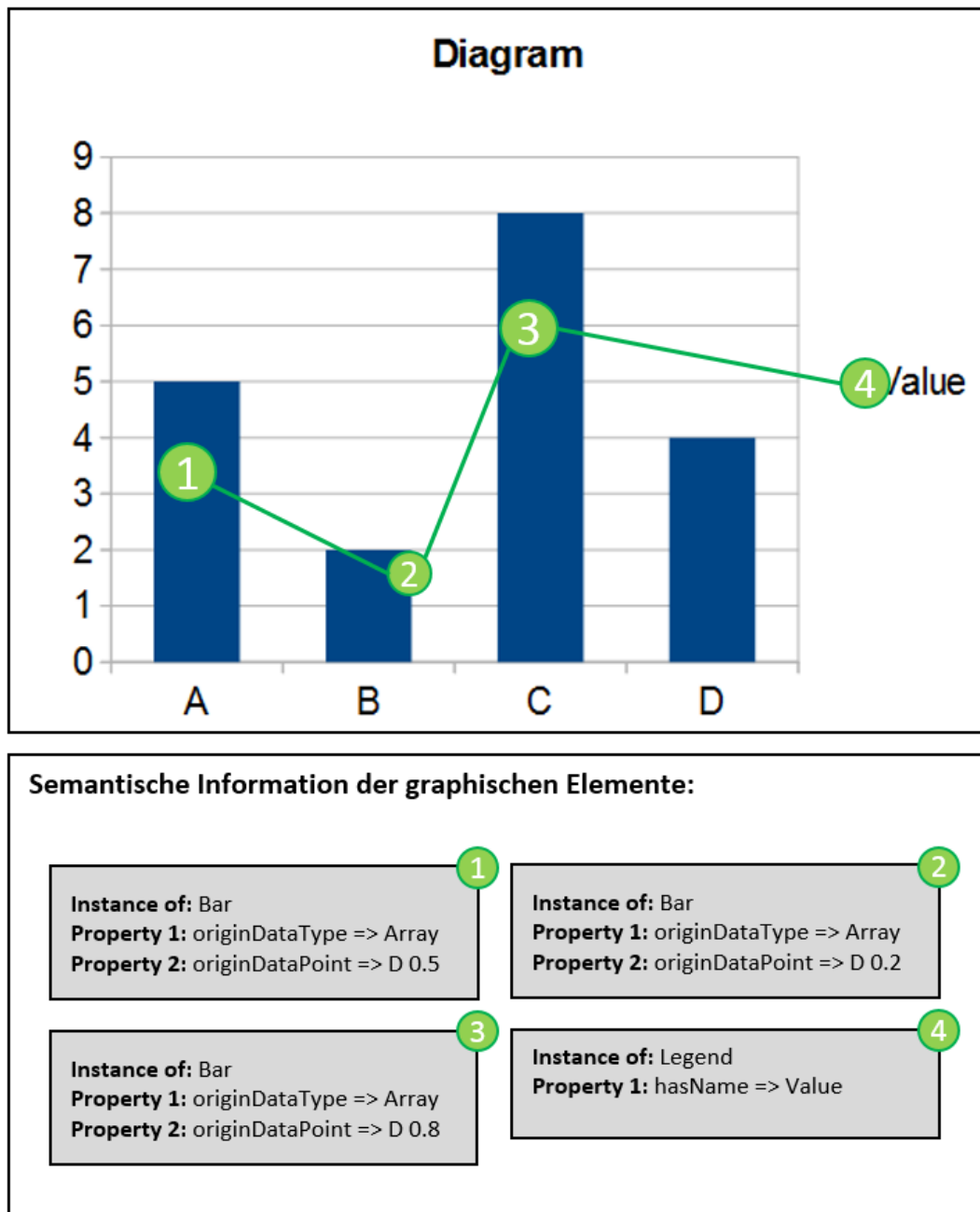


Abbildung 5.6: Illustration des Hit-Tests: Ein JavaScript-Programm simuliert Fixationen auf der Visualisierung mit Hilfe der Emulation von Mausklicks. Als Ergebnis wird eine Liste der graphischen Elemente an dieser Position mit ihrer semantischen Information ausgegeben.

5.4 Visuelle Analyse von kognitiven Prozessen

In Abschnitt 5.1 wurde ein Konzept zur visuellen Analyse von Wissensverarbeitungsprozessen vorgestellt, das in dieser Arbeit zur Analyse von kognitiven und perceptiven Prozessen verwendet wird. Zur visuellen Analyse von kognitiven Prozessen wurde dazu die parallele Scan-Path-Visualisierungstechnik (Kapitel 4) erweitert. Die Verwendung der PSP-Visualisierungstechnik hat den Vorteil, dass ähnlich eines Sequenzdiagramms Wissensverarbeitungsprozesse auf dem Wissenmodell untersucht werden können. Damit erhält der Analyst direkten Einblick in die kognitiven Prozesse eines Probanden. Übergänge zwischen einzelnen Zuständen der mentalen Verarbeitung sowie die Zeitdauer, die ein Proband mit der Betrachtung einer graphischen Information beschäftigt ist, können direkt aus der PSP-Visualisierung abgelesen werden. Damit kann direkt ein möglicher Programmablauf für eine Simulation der visuellen Suche erkannt oder simulierte Augenbewegungen mit realen Eye-Tracking-Daten verglichen werden. Zusätzlich können die in Kapitel 4.1 beschriebenen Darstellungsformen der PSP-Visualisierung sowie die Methode zum Vergleich von Fixationspfaden (Kapitel 4.2) für die Analyse von kognitiven Prozessen eingesetzt werden kann.

In Abschnitt 5.4.1 wird die generelle Darstellungsweise von Wissensverarbeitungsprozessen in PSP-Visualisierungen beschrieben. Anschließend wird in Abschnitt 5.4.2 mit Fallbeispielen, anhand von Daten aus Benutzerexperimenten gezeigt, wie die PSP-Visualisierung während der Analyse der Augenbewegungen Rückschlüsse auf kognitive Prozesse zulässt. Abschließend wird in Abschnitt 5.4.3 gezeigt, wie Fixationspfade in PSP-Visualisierungen Prozesse abbilden, die Hinweise auf eine kognitive Verzerrung sind.

5.4.1 Darstellung der Wissensverarbeitung in PSP-Visualisierungen

Im Folgenden wird beschrieben, wie aus Mustern an Augenbewegungen Rückschlüsse auf die drei folgenden dabei ablaufenden kognitiven Prozesse gezogen werden können:

- Lesen von Informationen in einer Visualisierung
- Ziehen von Schlussfolgerungen

Teile dieses Abschnitts basieren auf:

Raschke, Michael; Blascheck, Tanja; Richter, Marianne; Agapkin, Tanja; Ertl, Thomas: Visual Analysis of Perceptual and Cognitive Processes. In: *Proceedings of the International Joint Proceedings of the 5th International Conference on Information Visualization Theory and Applications, IVAPP 2014*, S. 284-291, 2014.

- Kombiniertes Lesen von Informationen mit Ziehen einer Schlussfolgerung

Abbildung 5.7 zeigt dazu drei exemplarische Beispiele. Mit Hilfe dieser drei Muster von Augenbewegungen können auf Basis der Augen-Gedanken-Hypothese Rückschlüsse auf kognitive Prozesse gezogen werden.

Lesen von Information in einer Visualisierung

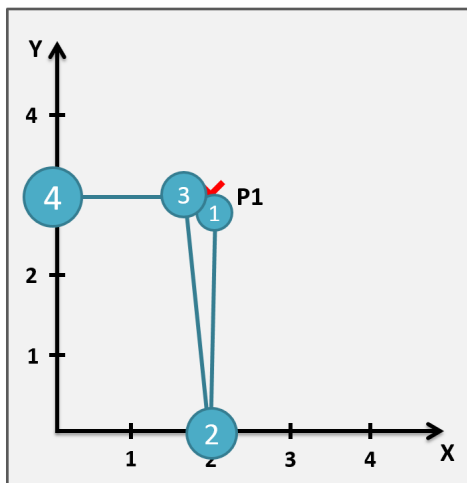
Abbildung 5.7a) zeigt einen exemplarischen Stimulus für die Aufgabe „Lesen Sie die X- und Y-Koordinate des Punktes ab.“, zusammen mit einem idealen Scan-Path (Abbildung 5.7b). Ein idealer Scan-Path für diese Aufgabe könnte die Form haben, dass der Proband erst den Datenpunkt im Kartesischen Koordinatensystem fokussiert, dann mit seinen Augen auf die X-Achse geht, von dort aus abermals den Datenpunkt fokussiert und dann seine Aufmerksamkeit auf die Y-Achse bewegt. Sobald er diese vier Fixationen durchgeführt hat, ist er prinzipiell in der Lage, die Aufgabe richtig zu lösen, da er alle aufgabenrelevanten Bereiche auf dem Stimulus betrachtet hat.

Ziehen von Schlussfolgerungen

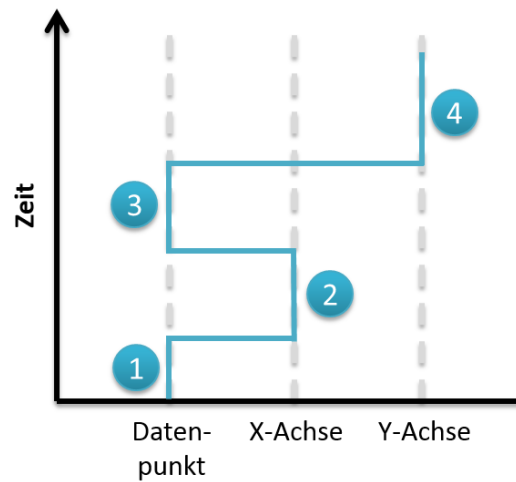
Abbildung 5.7c) zeigt einen exemplarischen Stimulus für die Frage „Warum ist die Straße nass?“ zusammen mit einem idealen Scan-Path auf dem Stimulus (Abbildung 5.7d). Ein idealer Scan-Path könnte folgende Fixationsreihenfolge haben: Zunächst wird die Straße fokussiert, dann die Pfütze (Bestätigung für die Angabe, dass die Straße nass ist), auf die Wolke (die Wolke regnet nicht), die Sonne, den Wasserhydranten (aus dem Wasser spritzt) und abschließend wieder auf die Pfütze. Diese Fixationskette bildet die Basis für den mentalen Prozess zum Ziehen der Schlussfolgerung, dass die Straße nass ist, weil der Hydrant spritzt (wahr) und nicht weil die Wolke regnet (Regen aus Wolke = falsch).

Kombination aus Lesen von Information und Ziehen einer Schlussfolgerung

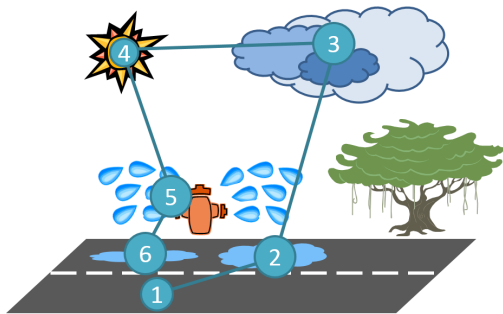
Schließlich zeigt Abbildung 5.7e) einen exemplarischen Stimulus für die Aufgabe „Ist die tiefste Temperatur aller Temperaturmessungen im Süden Frankreichs tiefer als die Temperatur im Norden Frankreichs?“ zusammen mit einem idealen Scan-Path auf dem Stimulus (Abbildung 5.7f). Der ideale Scan-Path zeigt hier das Leseverhalten der Informationen in der Visualisierung, mentale Prozesse zur Wissensspeicherung und zum Wissensabruf sowie logische Verarbeitungsprozesse des mental gespeicherten Wissens.



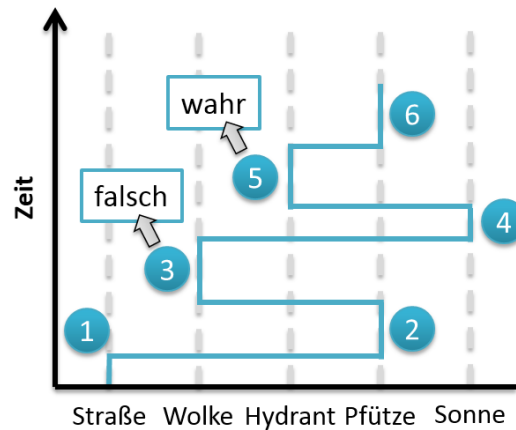
(a)



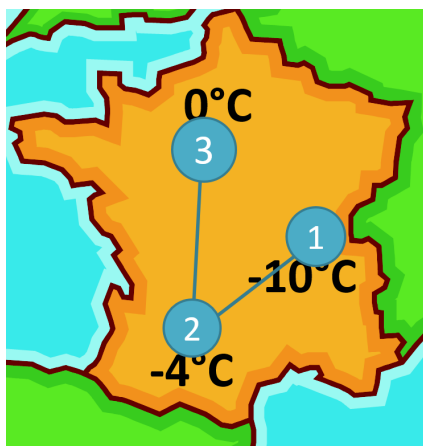
(b)



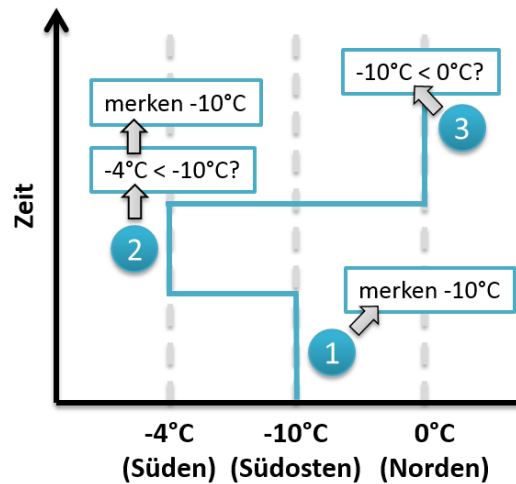
(c)



(d)



(e)



(f)

Abbildung 5.7: Visualisierung eines Lesevorgangs (a, b), dem Ziehen einer Schlussfolgerung (c, d) sowie die Kombination aus beiden (e, f). Jeweils gezeigt sind exemplarische Stimuli (linke Spalte) zusammen mit idealen Scan-Paths (rechte Spalte).

Beispielsweise könnte ein Proband zunächst den -10°C -Wert fokussiert und ihn sich behalten haben, anschließend den zweiten Wert im Süden von Frankreich (-4°C) fokussiert und beide miteinander verglichen haben ($-4^{\circ}\text{C} < -10^{\circ}\text{C} = \text{falsch}$). Den tieferen von beiden Werten (-10°C) merkte sich der Proband, las dann den Temperaturwert im Norden Frankreichs (0°C) und verglich diesen mit dem zuvor behaltenen tiefsten Wert aus dem Süden Frankreichs ($-10^{\circ}\text{C} < 0^{\circ}\text{C} = \text{wahr}$). Dieser Vergleich führte den Probanden dann zur Antwort „Die tiefste Temperatur im Süden Frankreichs ist tiefer als die Temperatur im Norden Frankreichs.“.

5.4.2 Anwendungsbeispiele

Um die Anwendbarkeit der PSP-Visualisierungstechnik für die Analyse von perzeptiven und kognitiven Prozessen zu evaluieren, wurde eine Pilotstudie mit zehn Probanden durchgeführt. Das Eye-Tracking-Experiment bestand aus drei Aufgabenteilen und hatte einen Aufbau wie in Kapitel 2.4.1 beschrieben. Im Folgenden wird eine Auswahl von Fixationspfaden von Probanden gezeigt und demonstriert, wie die in Abschnitt 5.4.1 beschriebenen Muster für eine Analyse der kognitiven Prozesse verwendet werden können.

Fixationschleifen

Fixationsschleifen zeigen, wie oft ein Proband gleiche Bereiche in einer Visualisierung hintereinander fokussiert, bevor er eine Aussage zu der an diesen Stellen dargestellten Information gegeben hat. Das Auftreten von Fixationsschleifen ist ein Hinweis auf nicht eindeutig dargestellte graphische Information in der Visualisierung, sowie nicht optimal gewählte Visualisierungsparameter, die ein schnelles Erkennen behindern.

Abbildung 5.8a) zeigt dazu einen der Stimuli aus dem Experiment. In diesem Beispiel wurde eine Beschriftung der X-Achse in numerisch, aufsteigender Reihenfolge verwendet. Die Probanden sollten den Wert der Kategorie „7“ angeben. Die annotierten Gebiete im Stimulus sind in Abbildung 5.8b) dargestellt. Es wurden vier AOIs definiert: eine für die Beschriftung der Kategorie „7“ auf der X-Achse („bar label“), eine am oberen Ende des Balkens der Kategorie „7“ („bar top“), eine auf dem dazu gehörenden Werteabschnitt auf der Y-Achse („bar value“) und eine vierte AOI im Bereich des schwarzen Kreuzes („black cross“), das vor dem eigentlichen Stimulus präsentiert wurde. Diese AOIs decken alle für die Durchführung der Aufgabe relevanten Bereiche in der Visualisierung zusammen mit den darin enthaltenen graphischen Elementen ab. Die Heat-Map aller Probanden ist in Abbildung 5.8c) abgebildet. Zu sehen ist eine

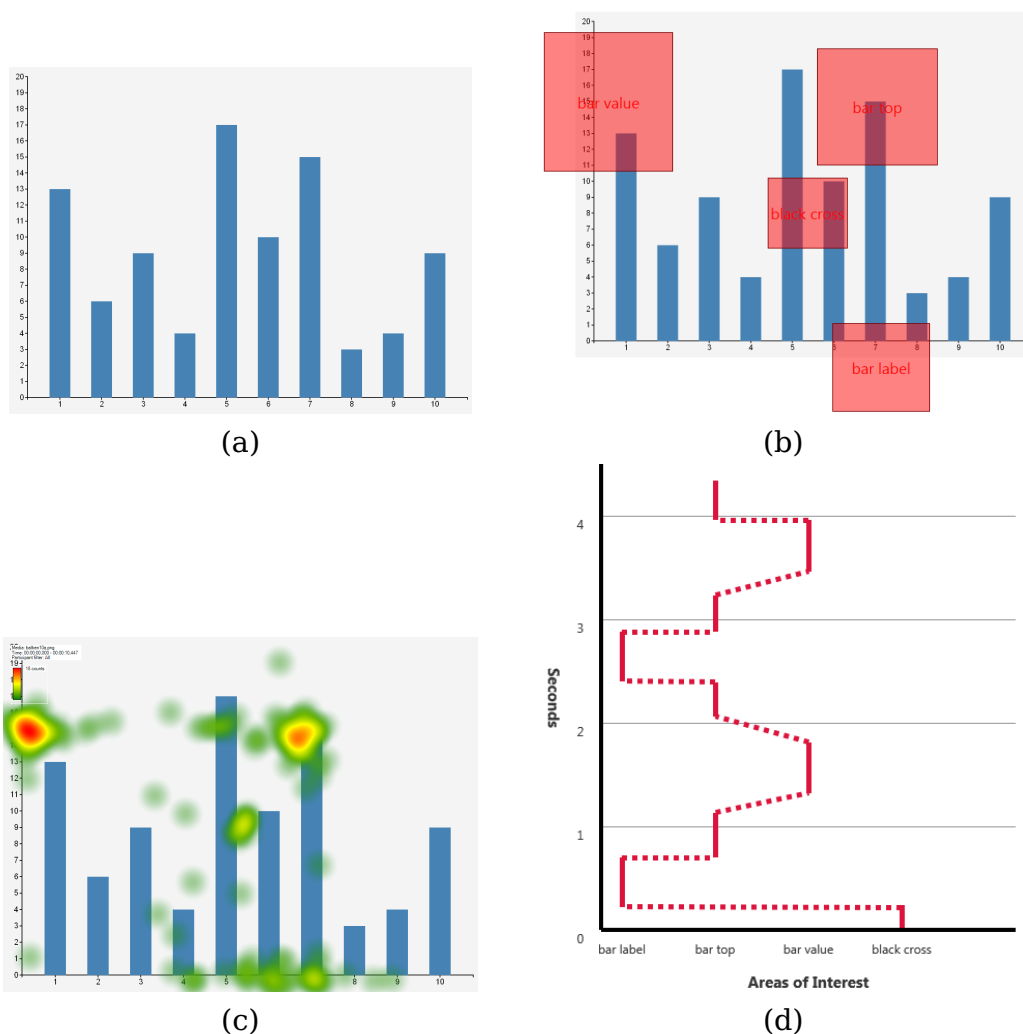


Abbildung 5.8: Stimulus (a), AOIs (b), Heat-Map (c) und PSP-Visualisierung (d) eines Probanden für die Aufgabe „Lesen Sie den Wert der Kategorie '7'.“.

eindeutige Konzentration der Fixationen in den für die Aufgabe relevanten Gebieten. Somit können die definierten AOIs bestätigt werden.

Abbildung 5.8d) zeigt die PSP-Visualisierung eines Probanden aus dem Experiment. Aus dem Diagramm kann direkt die Durchführungszeit von ungefähr 4,5 s abgelesen werden. Die Analyse des Fixationspfades zeigt, dass der Proband zunächst auf das schwarze Kreuz geschaut hat und anschließend schnell die Kategorie „7“ auf der X-Achse gefunden hat. Danach wechselte seine Aufmerksamkeit zwei Mal zwischen dem oberen Ende des Balkens der Kategorie „7“ und dem korrespondieren Bereich auf der Y-Achse. Schon nach dem ersten Mal hätte er die Aufgabe lösen können, da er alle aufgabenrelevanten Bereiche auf dem Stimulus gefunden und

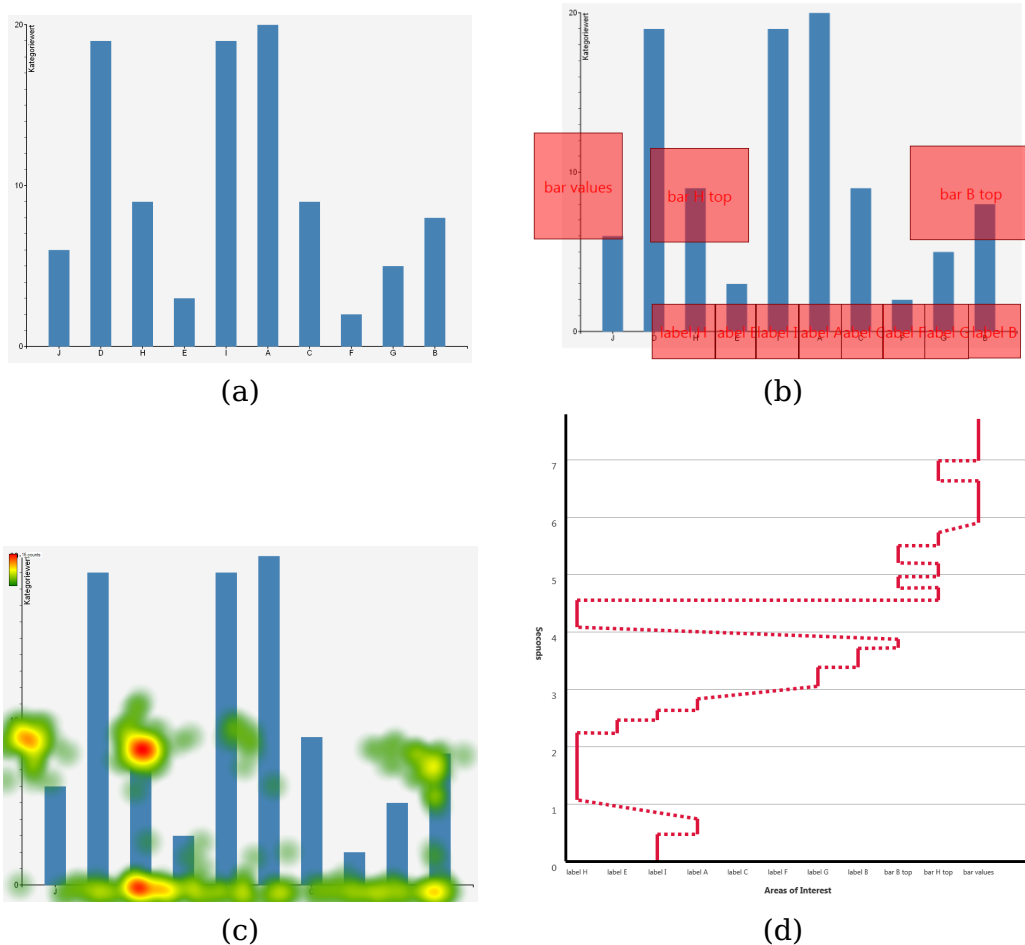


Abbildung 5.9: Stimulus (a), AOIs (b), Heat-Map (c) und PSP-Visualisierung (d) eines Probanden für die Aufgabe „Vergleichen Sie die Werte der Kategorien 'H' und 'B' und geben Sie die höhere Kategorie mit ihrem Wert an.“.

betrachtet hatte. Jedoch schien er sich bei der Beantwortung seiner Frage noch nicht ganz sicher zu sein und überprüfte seine mögliche Antwort durch eine zweite Augenbewegung zwischen dem oberen Ende des Balkens und der Y-Achse. Diese zweite Fixationsschleife über die gleichen Bereiche im Stimulus kann deutlich ab ca. 3 s in der PSP-Visualisierung erkannt werden. Diese Augenbewegung kann ein Hinweis darauf sein, dass der Proband den erkannten Wert des Balkens ein zweites Mal überprüfen wollte.

Visuelle Suche

Abbildung 5.9a) zeigt einen Stimulus zu der Aufgabe „Vergleichen Sie die Werte der Kategorien 'H' und 'B' und geben Sie die höhere Kategorie

zusammen mit ihrem Wert an.“ Für die Analyse des Blickverhaltens wurden die in Abbildung 5.9b) dargestellten AOIs definiert. So decken AOIs einzelne Bereiche auf der X-Achse des Säulendiagramms, die beiden oberen Enden der Säulen „H“ und „B“ sowie den korrespondierenden Wertebereich auf der Y-Achse ab. Eine Heat-Map über alle Probanden des Experiments bestätigt die Anordnung der AOIs (Abbildung 5.9c).

Die PSP-Visualisierung in Abbildung 5.9d) eines Probanden zeigt, dass er zunächst die Kategorien „H“ und „B“ startend von der Kategorie „I“ aus gesucht hat. Dieser Suchprozess wird in der PSP-Visualisierung als ein von links nach rechts aufsteigender Fixationspfad dargestellt. Für die Suche der beiden Kategorien benötigte der Proband ungefähr 4,5 s. Sobald er die Beschriftungen der Kategorien auf der X-Achse gefunden hatte, verglich der Proband die Höhen der Säulen „H“ und „B“. Dieser Prozess wird durch eine wechselnde Aufmerksamkeit zwischen den AOIs „top point bar B“ und „top point bar H“ gezeigt. Zum Schluss las der Proband den Wert der Säule „H“ auf der Y-Achse des Säulendiagramms.

Ziehen von Schlussfolgerungen

Die Analyse von kognitiven Prozessen während des Ziehens von Schlussfolgerungen wird anhand eines Venn-Diagramms demonstriert. Da Venn-Diagramme AND, OR, NAND, NOR oder XOR Beziehungen abbilden, bieten sie eine sehr gute Möglichkeit, das Ziehen von logischen Schlussfolgerungen zu untersuchen. Ein Venn-Diagramm aus dem Experiment zeigt Abbildung 5.10a). Die Aufgabe bestand darin anzugeben, wie viele Personen ein Abitur besitzen oder ihren Traumjob nur durch das Absolvieren einer Ausbildung gefunden hatten.

Abbildung 5.10b) zeigt die auf dem Stimulus definierten AOIs zur Analyse der Augenbewegungen. Es wurden insgesamt sieben AOIs definiert. Drei davon decken die Beschriftungen am rechten Rand des Venn-Diagramms ab („Abitur“, „d.j.“ für die Menge Traumjob, „Ausbildung“), vier jeweils, die für die Aufgabe relevanten Zahlenangaben im Venn-Diagramm („count Abitur“, „count Abi + d.j.“, „count d.j. + Ausbildung“ und „count d.j.“). Die Heat-Map in Abbildung 5.10c) zeigt, dass diese AOIs die Bereiche auf dem Stimulus abdecken, die am meisten von den zehn Probanden fokussiert worden sind.

Abbildung 5.10d) zeigt die PSP-Visualisierung eines Probanden. Während der ersten zwei Sekunden, die er den Stimulus betrachtet hatte, fokussierte der Proband die Beschriftungen auf der rechten Seite des Venn-Diagramms. Es ist davon auszugehen, dass der Proband sich relevanten Abschnitt im Venn-Diagramm dadurch eingepägt hat. Anschließend wechselte die Aufmerksamkeit des Probanden zwischen den Bereichen, die

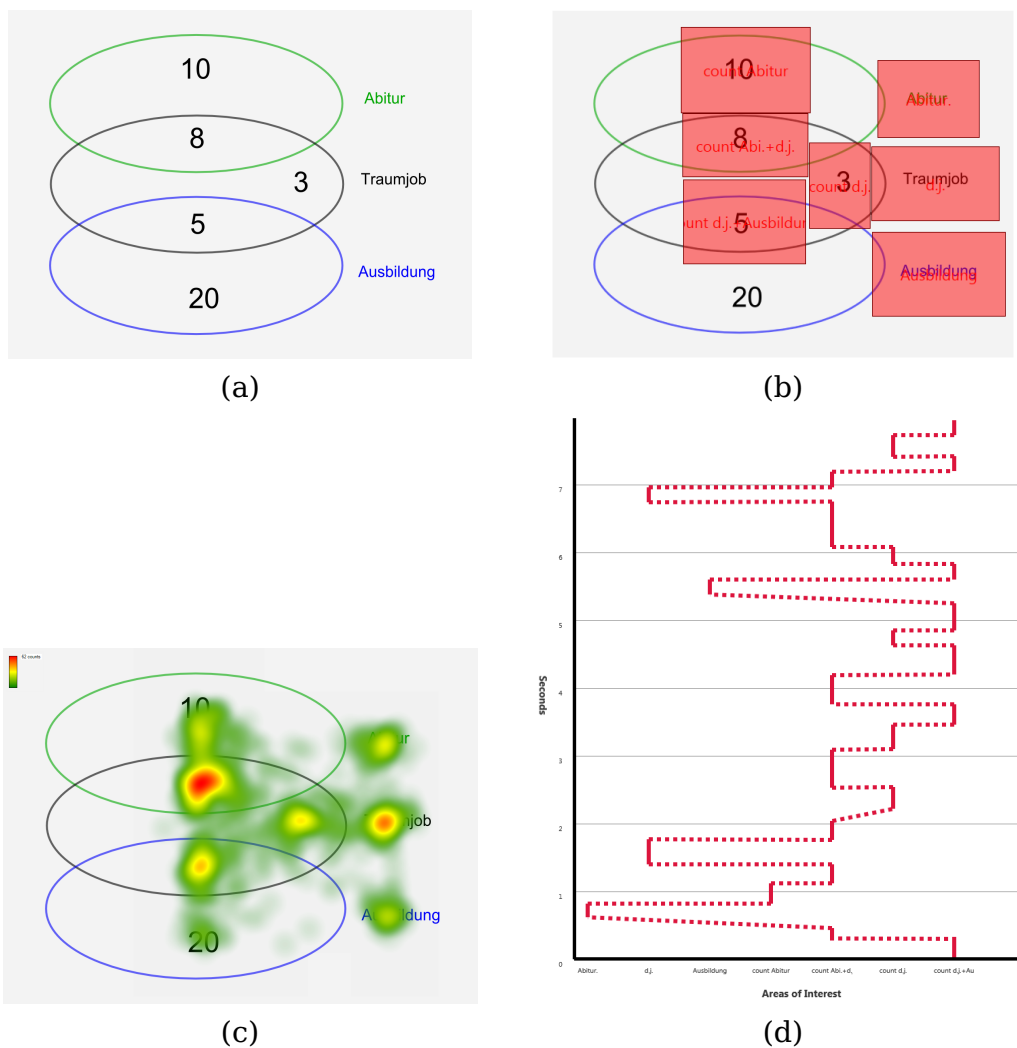


Abbildung 5.10: a) Stimulus zu der Aufgabe „Wie viele Menschen haben Abitur oder haben ihren Traumjob durch das Absolvieren einer Ausbildung gefunden?“. b) AOIs zur Analyse der kognitiven Prozesse decken die Beschriftung des Venn-Diagramms sowie die für die Aufgabe relevanten Angaben darin ab. c) Die Heat-Map über alle zehn am Experiment teilgenommenen Probanden bestätigen die Anordnung der AOIs. d) Die PSP-Visualisierung eines Probanden zeigt kognitive Prozesse, die als logische Prozesse interpretiert werden können.

folgende logische Verknüpfungen darstellten: „Anzahl der Menschen mit einem Traumjob UND Abitur“, „Anzahl der Menschen mit einem Traumjob UND einer Ausbildung“ und „Anzahl der Menschen mit einem Traumjob“. In dieser zweiten Phase der Augenbewegungen zeigt die PSP-Visualisierung kognitive Prozesse, die als logische Prozesse interpretiert werden können.

Im letzten Drittel des Experiments führte der Proband eine Fixations-

schleife durch, um die gefundene Antwort nochmals zu überprüfen.

5.4.3 Kognitive Verzerrung

Unter *kognitiver Verzerrung* versteht man, dass es zu einem Verständnisproblem beim Lesen von Informationen, die in einer Visualisierung dargestellt sind, kommt (siehe dazu auch Abschnitt 2.1.2). Dafür kann es vielfältige Ursachen geben. Visuell dargestellte Information kann von verschiedenen Menschen unterschiedlich interpretiert werden. Neben einer schlechten Wahl von Visualisierungsparametern kann dies auch an unterschiedlichen Ausgangsbedingungen für die Interpretation, wie z.B. kulturellen oder intellektuellen Unterschieden, liegen. Ebenso kann aber auch ein verschiedener Kontext oder die Vorliebe oder Abneigung für bestimmte graphische Darstellungen eine Rolle spielen. Im Extremfall hat dies zur Folge, dass eine graphische Darstellung in unterschiedlichen Situationen oder bei unterschiedlichen Aufgabenstellungen eine völlig andere Bedeutung besitzen kann.

In diesem Kapitel wurden mehrere Arten von kognitiven Prozessen vorgestellt, die mit Mustern im Blickverhalten identifiziert werden können. Damit kann die Analyse des Leseverhaltens einer Visualisierung dazu verwendet werden, um Ursachen für eine kognitive Verzerrung zu identifizieren. Auf Basis der Ergebnisse einer Analyse kann dann das Design einer Visualisierung optimiert werden.

In diesem Abschnitt wird eine Methode vorgestellt, wie die PSP-Visualisierung zusammen mit der semantischen Annotierung von Bereichen in einer Visualisierung für die Analyse von Ursachen der kognitiven Verzerrung verwendet wird⁴. Dazu wird ein Beispiel mit drei illustrativen Scan-Paths gegeben. Die in diesem Abschnitt beschriebene Demonstration lässt sich jedoch ohne weiteres auf Augenbewegungen aus realen Eye-Tracking-Experimenten übertragen.

In Abbildung 5.11a) ist ein zu Abbildung 5.7c) modifizierter Stimulus dargestellt, zusammen mit drei illustrativen Scan-Paths, zur Demonstration der Analyse. Das Beispiel zeigt wieder eine nasse Straße. Dieses Mal regnet jedoch auch die Wolke. Zusätzlich ist ein Auto auf der Straße abgebildet, das sowohl Teile der Pfütze als auch des Hydranten überdeckt. Die Frage ist wie in Abschnitt 5.4.1 wieder die gleiche: „Warum ist die Straße nass?“.

Die Methode besteht aus drei Schritten:

⁴Diese Methode wurde in Form des Kurzbeitrags „Michael Raschke and Thomas Ertl: Eye Tracking Framework for Analyzing Cognitive Bias“ während des VIS 2014 - Workshops *Dealing with Cognitive Biases in Visualisations (DECISIVE)* vorgestellt.

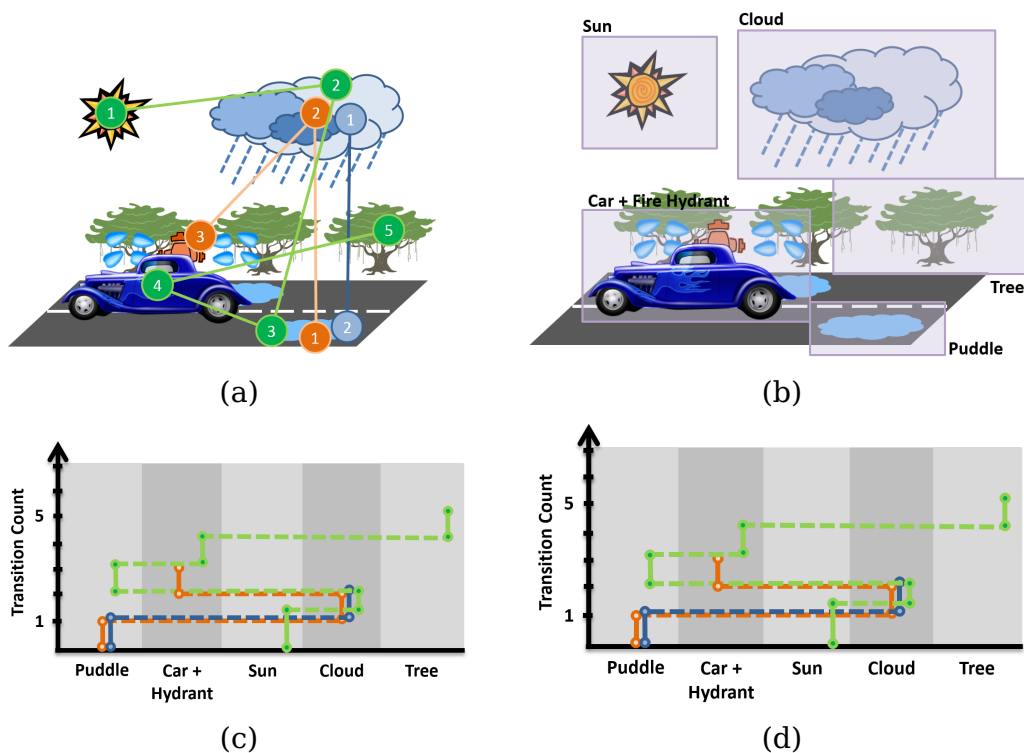


Abbildung 5.11: a) Modifizierter Stimulus zu der Frage „Warum ist die Straße nass?“ b) AOIs zur Analyse einer auftretenden kognitiven Verzerrung bei der Beantwortung der Frage c) PSP-Visualisierung eines idealen Scan-Paths (orange) und zwei illustrativen Scan-Paths zur Beantwortung der Frage d) Ergebnisse einer Ähnlichkeitsberechnung zwischen dem idealen Scan-Path und den zwei zwei illustrativen Scan-Paths.

1. Geeignete AOIs werden definiert.
2. Ein idealer Scan-Path wird definiert.
3. Dieser ideale Scan-Path wird mit den aufgenommenen Augenbewegungen verglichen.

1.) Definition von AOIs: Abbildung 5.11b) zeigt die zur Analyse der Augenbewegungen definierten AOIs. Es wurden vier AOIs definiert, welche die Sonne, Wolke, Pfütze und den Baum im Stimulus abdecken. Das Fahrzeug und der Hydrant werden gemeinsam von einer fünften AOI abgedeckt.

2.) Idealer Scan-Path: Ein idealer Scan-Path zum Beantworten der Frage wäre zum Beispiel, als erstes die Pfütze zu fokussieren, dann die Wolke und zum Schluss den Hydranten. Dieser ideale Scan-Path ist in Abbildung 5.11 a) und c) orange dargestellt. Mit diesem idealen Scan-Path sind

alle aufgabenrelevanten Bereiche betrachtet worden und die Antwort auf die Frage kann prinzipiell richtig gegeben werden.

3.) Vergleich der Scan-Paths: Abbildung 5.11c) zeigt den idealen Scan-Path zusammen mit zwei illustrativen Scan-Paths.

Der blaue Proband fokussierte nicht die AOI „Car + Firehydrant“. Aus diesem Grund kann er die Aufgabe nicht korrekt beantworten, da er nicht alle aufgabenrelevanten Bereiche betrachtet hat. Ein solcher Fixationspfad ist ein Hinweis darauf, dass eine Visualisierung die aufgabenrelevanten Informationen nicht optimal genug darstellt und wichtige Details in der Visualisierung übersehen werden können. Ein weiterer Grund für das Nichtfokussieren dieses Bereichs in der Visualisierung kann aber auch darin bestehen, dass der Proband eine Heuristik oder mentale Abkürzung angewendet hat. Als er eine Antwort auf die gegebene Frage gefunden hat, entschied er sich dafür die Suche nach weiteren Ursachen abzubrechen, warum die Straße nass ist. Der grüne Proband zeigt einen langen Fixationspfad. Er fokussierte alle aufgabenrelevanten Bereiche auf dem Stimulus, aber auch Bereiche, die nicht für die Aufgabe wichtig sein. Diese nicht relevanten Bereiche können ebenfalls zu einer kognitiven Verzerrung führen.

Abbildung 5.11d) zeigt das Ergebnis einer Ähnlichkeitsberechnung zwischen dem idealen Scan-Path und den zwei illustrativen Scan-Paths. Die Ähnlichkeitsberechnung ergibt, dass der blaue Proband einen Fixationspfad aufweist, der eine größere Ähnlichkeit zum idealen Scan-Path aufweist, als der Fixationspfad des grünen Probanden. Dieses Ergebnis lässt die Hypothese zu, dass die Wahrscheinlichkeit für das Auftreten einer kognitiven Verzerrung bei dem blauen Probanden geringer ist als beim grünen Probanden.

5.5 Leseregeln für Visualisierungen

Mit einem inversen WAS-Raumübergang können Augenbewegungen simuliert werden. Eine zentrale Rolle bei der Simulation von Augenbewegungen spielen Leseregeln für gegebene Visualisierungskonzepte und Aufgaben (Abschnitt 2.7.3). Diese werden in einen Satz von Produktionsregeln für eine ACT-R-Kognitionssimulation überführt und in der Kognitionssimulation verwendet (Abschnitt 2.7.6).

Für die Erstellung von Produktionsregeln gibt es zwei Ansätze: Erstens können Leseregeln manuell formuliert und auf deren Basis Produktionsregeln programmiert werden. Dieser Ansatz wird klassischerweise bei der Entwicklung von ACT-R-Simulationen verwendet und wurde auch für die

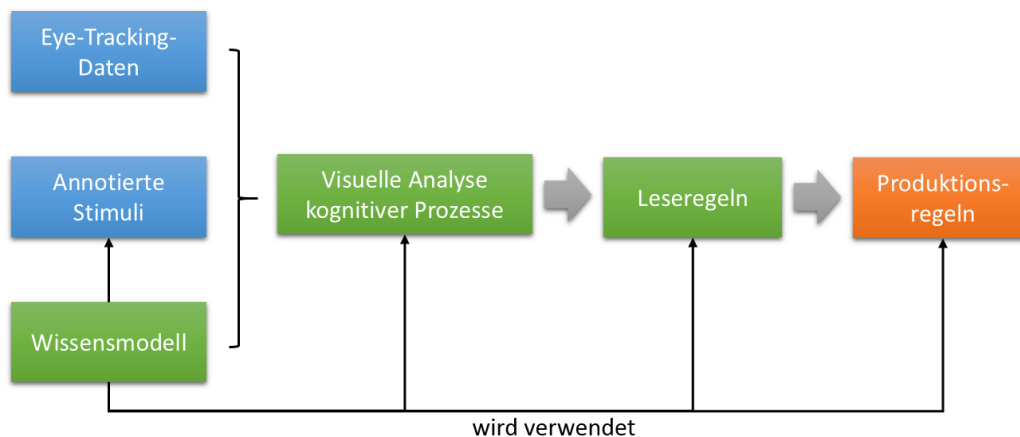


Abbildung 5.12: Für die Erstellung von Leseregeln werden Stimuli mit semantischer Information aus einem Wissensmodell annotiert, das ebenfalls für die visuelle Analyse der Augenbewegung und der Erstellung von Leseregeln und Produktionsregeln verwendet wird. Kognitive Prozesse werden mit Hilfe von aufgenommenen Augenbewegungen analysiert, um Repräsentanten von Fixationspfaden für die gegebenen Stimuli und Aufgaben berechnen zu können. Auf Basis dieser Repräsentanten werden Leseregeln erstellt, die dann in Produktionsregeln überführt werden.

Simulation der visuellen Suche in Liniendiagrammen von Peebles eingesetzt (Peebles und Cheng, 2002; Peebles, 2012). Dieser Ansatz ist jedoch sehr zeitaufwändig und unflexibel, da für jedes neue Visualisierungskonzept oder eine durchzuführende Aufgabe die Leseregeln bzw. Produktionsregeln angepasst oder komplett neu entwickelt werden müssen. Um den Aufwand für die Erstellung von Leseregeln zu reduzieren, stellt diese Arbeit einen alternativen Ansatz vor, mit dem Leseregeln aus Eye-Tracking-Daten durch eine visuelle Analyse von kognitiven Prozessen semiautomatisch erstellt und dann in Produktionsregeln überführt werden (Abbildung 5.12).

Für die Erstellung von Leseregeln müssen dazu die folgenden Schritte durchgeführt werden:

1. Annotierung der Stimuli mit semantischer Information
2. Berechnung von Repräsentanten aus den aufgenommenen Scan-Paths
3. Erstellung der Leseregeln auf Basis der Repräsentanten

Im Folgenden wird die Erstellung von Leseregeln mit diesem Verfahren für das Beispiel aus Kapitel 4.3.2 demonstriert, bei dem Probanden den

Wert des höheren Wertes von zwei Säulen in einem Säulendiagrammen angeben sollten.

Annotierung der Stimuli mit semantischer Information Ein Stimulus aus dem Experiment wurde bereits wie in Abbildung 4.17 gezeigt mit AOIs annotiert. Diese Annotierung muss für alle Stimuli, die im Experiment verwendet worden sind, durchgeführt werden. Durch die Annotierung des Stimulus mit AOIs werden Klassen aus dem Visualisierungsschema automatisch instanziiert.

Berechnung von Repräsentanten In Kapitel 4.2.1 wurde beschrieben, wie mit Hilfe von Clustering-Verfahren Repräsentanten von Fixationspfaden für verschiedene Gruppen von Betrachtern erstellt werden können. Dieser Repräsentant weist auf eine von einer größeren Anzahl von Betrachtern ausgeführte Leseregeln für einen gegebenen Stimulus und eine gestellte Aufgabe hin.

In Abbildung 4.20 sind zwei Repräsentanten zu der Aufgabe „Geben Sie den Wert des höheren Wertes der Säulen 'A' und 'D' an.“ dargestellt. Zwei mögliche Leseregeln bestehen darin, zuerst den Wert der Säule „A“ und dann der Säule „D“ zu lesen (blauer Repräsentant) oder zuerst den Wert der Säule „D“ und dann den der Säule „A“ zu lesen (roter Repräsentant), um anschließend beide miteinander zu vergleichen. In Abbildung 4.20 sind auch Unterziele zur Beantwortung dieser Frage zu erkennen. Zunächst müssen die Beschriftungen des Säulendiagramms gelesen werden, um die richtigen Balken identifizieren zu können. Für den Vergleich der beiden Säulen zeigt Abbildung 4.20 ebenfalls zwei Möglichkeiten: der blaue Proband liest zuerst den Wert der Säule „A“ auf der Y-Achse ab, dann den Wert der Säule „D“ und vergleicht beide gelesenen Werte mental miteinander. Der rote Proband fokussiert Säule „D“, dann Säule „A“ und vergleicht anschließend die Höhe der beiden Säulen direkt in der Visualisierung, um dann den Wert der höheren Säule auf der Y-Achse abzulesen. Beide Blickverhalten stellen zwei verschiedene Leseregeln dar, mit denen die Aufgabe korrekt gelöst werden kann.

Erstellung der Leseregeln Diese beiden Repräsentanten stellen aber nur das Leseverhalten für diesen einen Stimulus dar und geben an, welche graphischen Elemente des Visualisierungsmodells in welcher Reihenfolge fokussiert werden müssen. In einem Eye-Tracking-Experiment werden in der Regel jedoch mehrere Stimuli eines gleichen Visualisierungskonzepts gezeigt, mit denen Probanden gleiche Aufgaben durchführen müssen.

Da die graphischen Elemente des Visualisierungsmodells Instanzen von Elementen des Visualisierungsschemas sind, können für diesen Stimulus

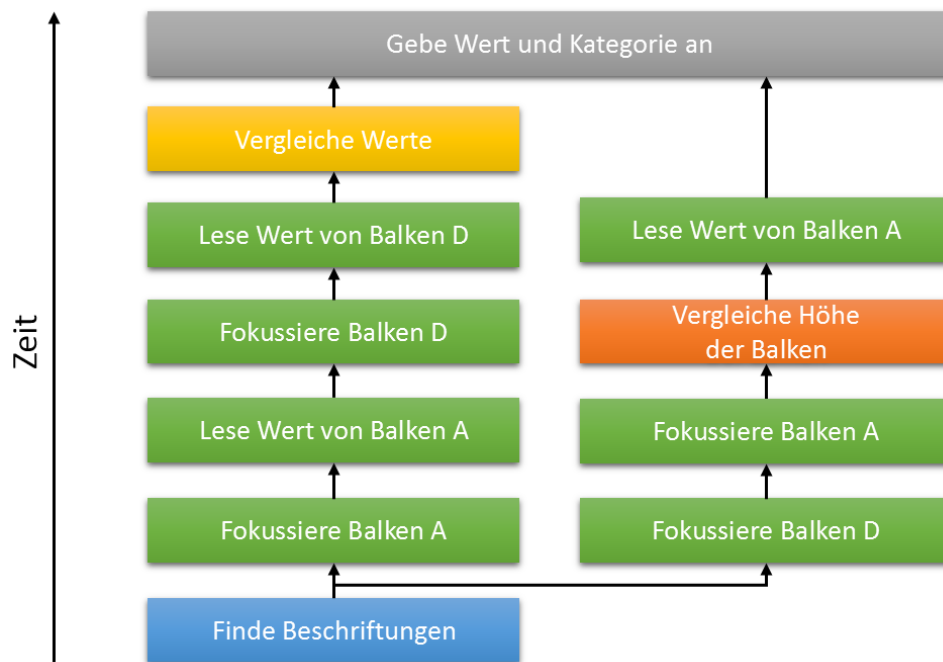


Abbildung 5.13: Zwei Leseregeln zur Aufgabe „Geben Sie den Wert des höheren Wertes der Säulen 'A' und 'D' an.“ und des Stimulus aus Abbildung 4.17a). Die Leseregeln wurden auf Basis der in Abbildung 4.20 dargestellten Repräsentanten erstellt.

gültige Leseregeln auf der Ebene des Visualisierungsschemas formuliert werden. Die Leseregeln bilden dazu einen Pfad an graphischen Elementen durch das Visualisierungsschema ab, die nacheinander fokussiert werden müssen.

Um für ein Visualisierungsschema allgemein gültige Leseregeln zu erhalten, wird auf der Ebene des Visualisierungsschemas ein zweites Mal ein Repräsentant berechnet. Dieser Repräsentant wird dann für die Überführung in Produktionsregeln verwendet. Dazu werden die URIs der Elemente aus dem Visualisierungsschema in der Reihenfolge ausgegeben, wie sie von dem Repräsentanten auf der Ebene des Visualisierungsschemas vorgegeben werden. Diese URIs werden in der Kognitionssimulation weiter zu Produktionsregeln verarbeitet.

5.6 Zusammenfassung und Diskussion

In diesem Kapitel wurde ein Verfahren vorgestellt, das eine visuelle Analyse von kognitiven Prozessen erlaubt. Das Verfahren verwendet die Annotierung von Bereichen oder graphischen Elementen in Visualisierung mit

Elementen aus einem Wissensmodell, um eine Analyse der Scan-Paths im WAS-Raum zu ermöglichen. Dazu wurde ein Wissensmodell über Visualisierungen, den darin enthaltenen graphischen Elementen und ihren Relationen zueinander entwickelt. Zur Analyse der kognitiven Prozesse wird eine modifizierte Version der parallelen Scan-Path-Visualisierungstechnik verwendet, mit deren Hilfe verschiedene Muster in den Augenbewegungen gefunden und mit kognitiven Prozessen identifiziert werden.

Die Analyse von kognitiven Prozessen mit Hilfe von Eye-Tracking basiert maßgeblich auf der Augen-Gedanken-Hypothese. Wie in Kapitel 2.7.1 beschrieben, wird in der Eye-Tracking-Forschung diese Hypothese oft als Grundlage verwendet, um experimentelle Ergebnisse interpretieren zu können. Jedoch zeigen eine Vielzahl von Experimenten, dass die Augen-Gedanken-Hypothese nur mit Vorsicht bei der Analyse von Eye-Tracking-Daten angewendet werden sollte. Im Rahmen der in dieser Arbeit durchgeführten Eye-Tracking-Experimente wird die Augen-Gedanken-Hypothese grundsätzlich als wahr angenommen. Während der Durchführung der Eye-Tracking-Experimente zeigte es sich, dass mehr Hinweise auf kognitive Prozesse zum Lösen der gegebenen Aufgaben in den Augenbewegungen gefunden werden konnten, je abstrakter eine durchzuführende Aufgabe war. Ein Extrembeispiel stellen die in Abschnitt 5.4.2 verwendeten Venn-Diagramme dar, die Regeln aus der Mengenlehre graphisch darstellen. Hier können Muster in den Augenbewegungen gefunden werden, welche darauf hinweisen, dass die Probanden die in Venn-Diagrammen graphisch kodierten Mengenoperatoren AND, OR, NAND, NOR oder XOR angewendet hatten. Es besteht jedoch die Möglichkeit, dass je weniger abstrakt eine Aufgabe ist, umso geringer dann kognitive Prozesse beim Verständnis des Stimulus eine Rolle spielen und sich somit im Blickverhalten zeigen. Dieser Zusammenhang könnte durch eine genauere Analyse von Blickbewegungen untersucht werden, die Aufgaben mit verschiedenen Ansprüchen an die kognitive Verarbeitung der Stimuli stellen. Weitere Untersuchungen zur Grenze der Anwendbarkeit der Augen-Gedanken-Hypothese könnten auch durch zusätzliche Auswertekanäle in Eye-Tracking-Experimenten wie retroperspektive Fragebögen oder die Analyse von lautem Denken erfolgen. Dazu stellten Blascheck und Ertl ein Konzept zur kombinierten Analyse vor (Blascheck und Ertl, 2014).

Neben den Beschränkungen bei der Interpretierbarkeit der Ergebnisse aus der visuellen Analyse von kognitiven Prozessen durch die Augen-Gedanken-Hypothese bestehen weitere technische Einschränkungen, die durch die Aufnahme von Augenbewegungen mit Hilfe von Eye-Tracking-Geräten entstehen. Mit Hilfe von Eye-Tracking-Geräten können seit einigen Jahren sehr präzise die Fixationen eines Probanden auf einem Stimulus aufgezeichnet werden. Jedoch müssen dabei die technischen Einschränkungen

der Eye-Tracking-Geräte wie Aufnahmefrequenz und Ortsauflösung beachtet werden. Zusätzlich muss während den Experimenten besonders auf eine sehr gute Kalibration der Eye-Tracking-Geräte geachtet werden. Dabei muss zwischen dem zeitlichen Aufwand, der für die Kalibration notwendig ist und der dadurch gewonnenen Erhöhung der Aufnahmegenauigkeit abgewogen werden. Ein weiterer kritischer Punkt bei der Analyse von Augenbewegungen ist die optimale Wahl des Algorithmus, der aus den vom Eye-Tracker aufgezeichneten „rohen“ Blickpunkten Fixationen berechnet (siehe dazu auch Abschnitt 2.4.2). Die Auswahl eines geeigneten Algorithmus und die passende Wahl der Parameter trägt maßgeblich zu einer möglichst korrekten Interpretation des Blickverhaltens und der daraus abgeleiteten kognitiven Prozesse bei.

Wird die Augen-Gedanken-Hypothese als wahr angenommen und ist die Qualität der aufgenommenen Augenbewegungen sehr gut, stellen sich weitere Fragen bezüglich des verwendeten Wissensmodells, der visuellen Analyse der Fixationspfade und der Formulierung von Leseregeln.

Die Aufteilung des Wissens über Visualisierungen und den darin enthaltenen graphischen Elementen auf drei Abstraktionsebenen war bei der Entwicklung des Verfahrens zunächst rein technisch motiviert, kann jedoch durch die Arbeit von Pinker (Pinker, 1990) auch aus Sicht der Kognitionswissenschaft motiviert werden. Weitere Hinweise auf das Vorhandensein von Schemata wurde auch in Form der Prototypen-Theorie vorgestellt, welche die Vorstellung einer rein taxonomisch aufgebauten Wissensstruktur im Gedächtnis abgelöst hatte (Rosch, 1973, 1975). Ein interessanter Aspekt eines graphbasierten Modells der Wissensrepräsentation besteht darin, dass wie von Anderson beschrieben (Anderson, 1983a) damit Prozesse der Aktivationsausbreitung im Gedächtnis untersucht werden können und bei Simulationen der visuellen Suche dadurch besser beachtet werden können. Hierzu muss jedoch das vorgestellte visuelle Analyseverfahren weiterentwickelt und die PSP-Visualisierung gegebenenfalls angepasst oder durch eine andere Visualisierungstechnik ersetzt werden.

Die visuelle Analyse mit Hilfe der PSP-Visualisierung sieht zunächst eine Abbildung der einzelnen Fixationen aus dem WO-Raum auf Elemente des WAS-Raums vor. Dazu werden exakte Informationen zur Position der Fixationen auf dem Stimulus verwendet. Damit wird jedoch die Fähigkeit des menschlichen Auges nicht beachtet, Informationen im peripheren Sehbereich wahrzunehmen. Hinzu kommt, dass die Wahrnehmungsleistung im peripheren Sehen je nach kognitiver Anforderung an die Verarbeitung der visuell wahrgenommenen Information schwankt (siehe dazu Abschnitt 2.2.1). Eine mögliche Lösung dafür könnte darin bestehen, dass nicht nur die exakt fokussierten graphischen Elemente bei der Analyse der kognitiven Prozesse beachtet werden, sondern auch graphische Elemente,

die sich weiter entfernt von der Fixation befinden. Diese weiter entfernten graphischen Elemente würden dann aber mit einer geringeren Gewichtung in die Analyse einfließen.

Die visuelle Analyse mit Hilfe der PSP-Visualisierung bringt die Einschränkung mit sich, dass nur eine beschränkte Anzahl von AOIs oder graphischen Elementen auf der X-Achse der Visualisierung abgebildet werden kann. Eine mögliche Lösung dazu wurde zwar in Kapitel 4.1 durch das Ausblenden von AOIs in der PSP-Visualisierung vorgestellt, diese Ausblendung führt jedoch direkt zu einer Einschränkung des WAS-Raums in der PSP-Visualisierung. Zusätzlich entsteht Visual Clutter je mehr Fixationspfade visualisiert werden oder je stärker sich diese unterscheiden. Ein Clustering der Fixationspfade reduziert diese Problematik grundsätzlich zwar. Diese Optimierung der Visualisierung wird jedoch dadurch erkauft, dass zusätzliche Variablen bei der Auswertung der kognitiven Prozesse in Form der Parameter des Cluster-Algorithmus hinzukommen. Diese müssen sowohl bei der Analyse der kognitiven Prozesse, wie auch bei der Erstellung von Leseregeln beachtet werden. Der Übergang vom WO- in den WAS-Raum sieht bisher vor, dass jeweils nur eine AOI einen bestimmten Bereich auf dem Stimulus bedeckt. Bei mehreren AOIs, die einen gleichen Bereich abdecken, muss manuell entschieden werden, welche graphischen Elemente in diesen AOIs im weiteren Verlauf der Analyse beachtet werden.

Der letzte Schritt des Verfahrens besteht in der Erstellung von Leseregeln. Hier findet sich zum einen wieder die Problematik der zusätzlichen Variablen in der Analyse, die durch die Anwendung von Clustering-Algorithmen zur Berechnung von Repräsentanten hinzukommen. Zum anderen können auch verschiedene Leseregeln für eine gleiche Aufgabe wie im Beispiel oben beschrieben gefunden werden. In diesem Zusammenhang stellt sich auch die Frage, wann ein Blickverhalten als eine neue Leseregeln interpretiert werden muss oder diese als eine ähnliche Variante einer existierenden Leseregeln angesehen werden kann. Innerhalb der Leseregeln kann auch ein „Rauschen“ an Augenbewegungen existieren, wenn beispielsweise zu viele Fixationsschleifen im Repräsentanten existieren. Ist dies der Fall, muss entschieden werden, welche der Leseregeln simuliert werden soll oder wie gefundene Leseregeln vereinfacht werden.

Wichtig bei der Erstellung der Leseregeln ist auch eine geeignete Repräsentation des Wissens über Visualisierungen im Wissensmodell und eine optimale Verbindung der Elemente im Wissensmodell mit graphischen Elementen oder AOIs in den Visualisierungen. Hier spielt der explorative Ansatz der Datenanalyse der visuellen Analytik seinen Vorteil aus. Das Wissensmodell kann dadurch schrittweise angepasst werden. Leseregeln können iterativ formuliert, in ACT-R-Code programmiert und durch den Vergleich mit empirisch ermittelten Scan-Paths effizient evaluiert werden.

Modellierung und Simulation der visuellen Suche

Meist werden Benutzerexperimente inkrementell entworfen und durchgeführt (Kapitel 2.4). Zunächst wird ein Konzept der Benutzerstudie basierend auf Forschungsfragen erstellt. Dann werden die Stimuli entworfen, die Anzahl der Stimuli im Experiment festgelegt, Aufgabenbeschreibungen und Übungsaufgaben formuliert. Anschließend wird eine Pilotstudie durchgeführt. Ergebnisse aus dieser Pilotstudie werden analysiert, um den experimentellen Aufbau weiter zu optimieren. Erst dann wird das eigentliche Hauptexperiment durchgeführt, dessen Ergebnisse abschließend analysiert werden, um eingangs formulierte Hypothesen zu beweisen oder zu falsifizieren (Field und Hole, 2003).

Dieser inkrementelle Vorgang ist sehr zeitaufwändig. Im Gegensatz zur Durchführung von Benutzerexperimenten für die Evaluation von Visualisierungstechniken existiert nur eine kleine Anzahl von Arbeiten zur Modellierung von visuellen Suchstrategien (Kapitel 2.7.7). Eine mögliche Ursache könnte darin liegen, dass der Aufwand zur Entwicklung von kognitiven Modellen der visuellen Suche als zu hoch wahrgenommen wird, um die Vorzüge der Simulation von Evaluationsergebnissen zu rechtfertigen.

Um den Aufwand zur Simulation der visuellen Suche zu reduzieren, stellt dieses Kapitel zwei Methoden vor, mit denen die visuelle Suche modelliert werden kann. Aufbauend auf den Kapiteln 4 und 5 wird in Abschnitt 6.1 das Simulationsframework *cSim* vorgestellt, mit dem Augenbewegungen simuliert werden können. Die zweite Methode besteht in Form eines operatorenbasierten *Diagram-Viewing-Modells* zur Vorhersage von Durchführungszeiten von visuellen Aufgaben (Abschnitt 6.2). Das Kapitel schließt mit einem kurzen Vergleich der beiden Methoden.

Teile dieses Kapitels wurden veröffentlicht in:

Raschke, Michael; Engelhardt, Stephan; Ertl, Thomas: A Framework for Simulating Visual Search Strategies. In: *Proceedings of the 11th International Conference on Cognitive Modeling, 2013*.

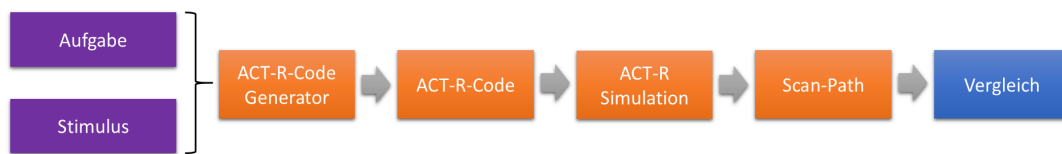


Abbildung 6.1: Auf Basis eines gegebenen Stimulus und einer Aufgabe erzeugt cSim automatisch ACT-R-Programmcode. Dieser wird durch das ACT-R-Simulationsframework ausgeführt. Der berechnete Scan-Path kann anschließend mit empirisch ermittelten Scan-Paths verglichen werden.

6.1 Simulation der visuellen Suche

Ein wichtiges Ziel der vorliegenden Arbeit war die Entwicklung eines Frameworks zur Simulation der visuellen Suche. Mit diesem Framework können für eine gegebene visuelle Aufgabe und eine bestimmte Visualisierungstechnik die Positionen und Reihenfolge von Fixationen simuliert werden, mit der diese Aufgabe durchgeführt wird. Dazu wurde eine auf dem Kognitionssimulationsframework ACT-R basierende Simulation entwickelt.

Da ACT-R-Programme in Lisp geschrieben sind, einen hohen Grad an Komplexität aufweisen und für jede neue Aufgabe oder Stimulus neu entwickelt oder angepasst werden müssen, bestand ein zweites Ziel bei der Entwicklung von cSim darin, die Simulation der visuellen Suche mit Hilfe von ACT-R möglichst benutzerfreundlich zu gestalten und gleichzeitig einen hohen Grad an Flexibilität zu erlauben. Das Ergebnis dieser Entwicklung bestand in einer in C# entwickelten Simulationsumgebung, welche auf Basis eines gegebenen Stimulus und einer gegebenen Aufgabe ACT-R-Programme automatisch erzeugt, diese durch ACT-R ausführen lässt und anschließend die simulierten Scan-Paths visualisiert. Die Scan-Paths können anschließend mit durchgeführten Eye-Tracking-Experimenten verglichen werden, um die Ergebnisse der Simulation zu validieren oder den Simulationscode gegebenenfalls weiter anzupassen (Abbildung 6.1).

Zunächst wird in Abschnitt 6.1.1 die grundlegende Architektur der Simulation vorgestellt. Dann werden in Abschnitt 6.1.2 die einzelnen Komponenten des Frameworks beschrieben. Für die Validierung des Ansatzes werden in Abschnitt 6.1.3 simulierte Augenbewegungen mit Ergebnissen eines Eye-Tracking-Experiments verglichen. Auf Basis dieser Demonstration werden die Stärken und Schwächen des Prototyps in Abschnitt 6.1.4 diskutiert und mit existierenden Arbeiten verglichen.

6.1.1 Grundlegende Architektur der Simulation

Das Simulationsframework cSim besteht aus den zwei Komponenten *Input-Modell* und *Kognitionssimulation* (Abbildung 6.2).

Das Input-Modell beinhaltet das Wissen über Visualisierungen. In dieser Komponente wird das in Abschnitt 5.2 vorgestellte Modell verwendet, um Wissen über Visualisierungen auf drei Abstraktionsebenen zu modellieren. Mit dem Input-Modell stellt der Entwickler eines Simulationsprogramms das deklarative und prozedurale Wissen für die Simulation zur Verfügung. Dieses Wissen wird automatisch durch cSim in ACT-R-Programmcode umgewandelt. Neben der Beschreibung des Wissens über einen vorliegenden Stimulus, beinhaltet das Input-Modell auch eine Beschreibung der Aufgabe, für welche die Augenbewegungen simuliert werden sollen. Aus dieser Aufgabe wird das Ziel des Simulationdurchlaufs abgeleitet.

Die zweite Komponente des Frameworks ist die eigentliche *Kognitionssimulation*. Diese Komponente simuliert die visuelle Suche in vier Schritten. Dazu verwendet diese Komponente das Input-Modell. Die Ausgabe der Kognitionssimulation besteht in Form einer Liste von Fixationen, die ausgeführt werden müssen, um eine in der Aufgabenbeschreibung vorliegende Aufgabe mit dem gegebenen Stimulus erfolgreich durchzuführen. Diese Fixationen können mit Ergebnissen aus Eye-Tracking-Experimenten verglichen werden.

6.1.2 Komponenten des Frameworks

Input-Modell

Kapitel 5.2 stellt ein Modell zur Beschreibung von Wissen über Visualisierungen und den darin enthaltenen graphischen Elementen auf den drei Abstraktionsebenen Visualisierungsontologie, Visualisierungsschema und Visualisierungsmodell vor. Für die Implementierung dieses Dreiebenenmodells wurde ein auf dem Managed Extensibility Framework von .NET basierendes Plug-In-System implementiert.

Jedes in der Visualisierungsontologie beschriebene graphische Element einer Visualisierung wurde in Form eines `VisualObject`-Plug-Ins realisiert. Die zwei wichtigsten Methoden dieses Objekts erzeugen einen *VisualLocation-Chunk* über die Angabe der Position des graphischen Elements und einen *VisualObject-Chunk* über die Klasse des sich an dieser Position befindenen graphischen Elements. Diese beiden Chunks werden für die Erzeugung von ACT-R-Code verwendet. Ein `VisualLocation-Chunk` für eine Säule in einem Säulendiagramm beinhaltet beispielsweise Angaben über deren Position, Höhe und Breite sowie Farbe:

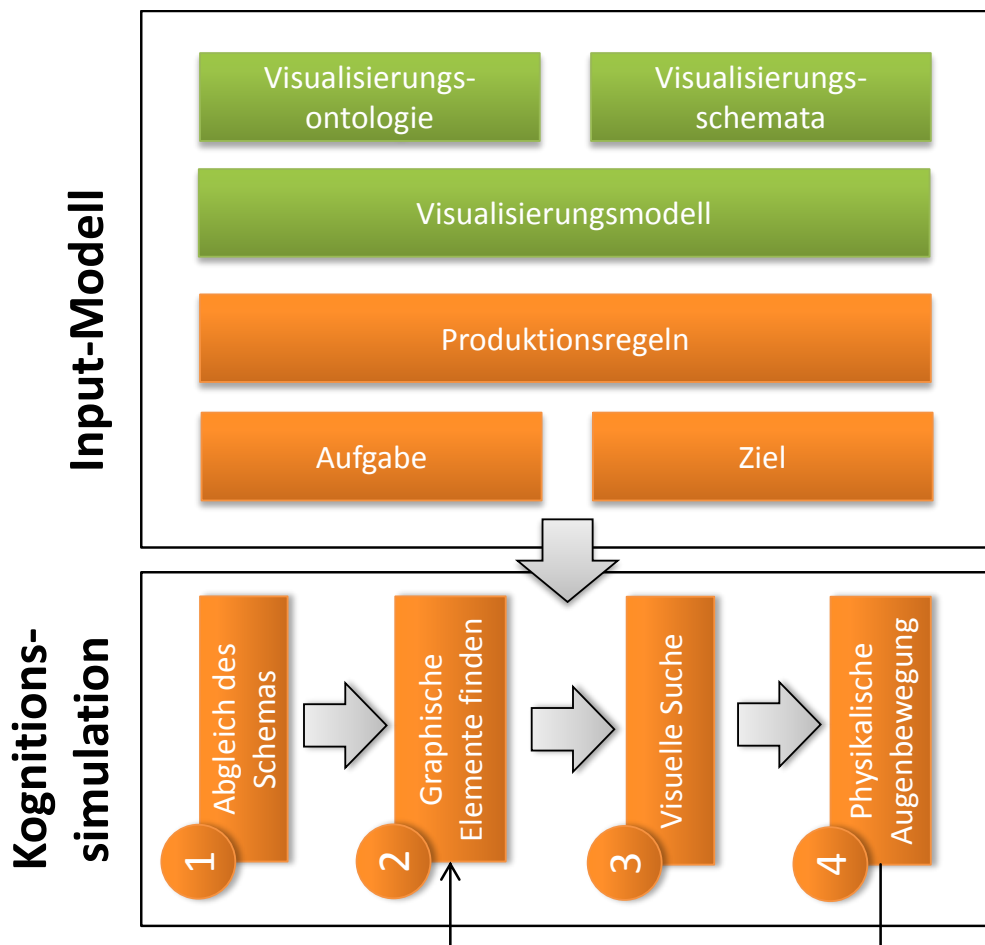


Abbildung 6.2: Das Simulations-Framework cSim besteht aus zwei Komponenten. Das Input-Modell beinhaltet die Wissenrepräsentation über eine gegebene Visualisierung sowie die durchzuführende Aufgabe. Die Kognitionssimulation simuliert die visuelle Suche in vier Schritten.

```
(ISA visual-location screen-x 93 screen-y 410 kind bar value bar
width 55 height 184 color black)
```

Der VisualObject-Chunk zu dieser Säule hat dann folgenden Inhalt:

```
(ISA bar value 'Saeule' width 55 height 184 color black
screen-x 93 screen-y 410)
```

Visualisierungsschemata für gegebene Visualisierungstechniken werden im Prototyp durch VisualizationSimulation-Plugins realisiert. Die wichtigste Methode jedes VisualizationSimulation-Plugins erzeugt ein ausführbares ACT-R-Programm für das im Visualisierungsschema beschriebene Visualisierungskonzept. Zur Erzeugung des ACT-R-Programms wird in der aktuellen Version des Prototypen eine Vorlage verwendet, die eine spezielle visuelle Aufgabe für dieses Visualisierungskonzept in ACT-R-Programmcode

beschreibt. Die Aufgabe muss in Form eines klar formulierten Textes vorliegen. Als Einschränkung für die Formulierung einer Aufgabe gilt, dass in dieser nur graphische Elemente und Eigenschaften sowie Literale dieser Elemente verwendet werden dürfen, die auch im Wissensmodell definiert sind. Um trotzdem in der Lage zu sein, Augenbewegungen für verschiedene Stimuli des gleichen Visualisierungskonzepts für eine gleiche Aufgabe simulieren zu können, beinhaltet die Programmvorlage Platzhalter im LISP-Programmcode für Werte oder Parameter, die durch das *Visualization Simulation*-Plugin ersetzt werden (Listing 6.1). Im Falle der Simulation einer Aufgabe mit einem Säulendiagramm sind dies Chunks für Achsen, Skalenstriche, Säulen, Säulenoberkanten und das Ziel (Goal) der Simulation (Listing 6.1).

Listing 6.1: Platzhalter im LISP-Programmcode des ACT-R-Programms werden durch das Simulationsframework automatisch ersetzt.

```
(chunk-type (axis (:include visual-object)) type)
(chunk-type (tick-label (:include text)) axis)
(chunk-type (bar (:include visual-object))
  screen-x screen-y)
(chunk-type (bar-top (:include visual-object))
  screen-x screen-y)
(chunk-type goal state abscissa x-axis-screen-y)
```

Hinzu kommen Chunks für das deklarative Wissen der Zwischenziele während des Simulationslaufs (Listing 6.2). Hier werden einzelne graphische Elemente aufgeführt, die nacheinander fokussiert werden müssen, um eine gegebene Aufgabe erfolgreich durchführen zu können. Für das Lesen des Wertes einer Säule in einem Säulendiagramm sind dies das Finden der X-Achse (Zeile 1), der passenden Abszisse (Zeile 2 und 3), Säule (Zeile 4) und das Lesen des zur Säulenoberkante (Zeile 5) korrespondierenden Wertes auf der Y-Achse (Zeile 6 und 7). Die Zwischenziele werden auf Basis der Leseregeln (Abbildung 6.3) eines gegebenen Visualisierungskonzepts definiert.

Listing 6.2: Zwischenziele für das Lesen eines Wertes in einem Säulendiagramm. Diese werden auf Basis der Leseregeln eines gegebenen Visualisierungskonzepts definiert.

```
1 (find-x-axis          ISA chunk)
2 (find-abscissa       ISA chunk)
3 (find-next-abscissa  ISA chunk)
4 (find-bar            ISA chunk)
5 (find-bar-top        ISA chunk)
6 (find-ordinate-label ISA chunk)
7 (read-value          ISA chunk)
```

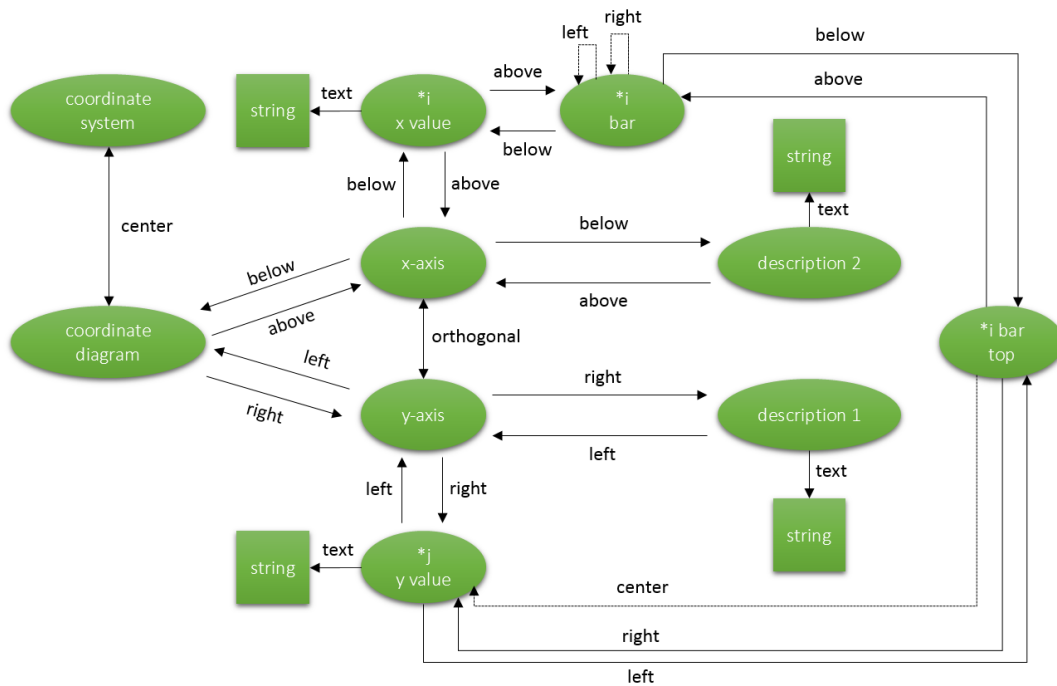


Abbildung 6.3: Leseregeln eines Säulendiagramms. Elliptische Knoten repräsentieren graphische Elemente in der Visualisierung. An diese sind Literale (Vierecke) angehängt. Quantifizierer (i, j) geben an, dass ein graphisches Element mehrmals in der Visualisierung dargestellt wird. Pfeile geben die räumliche Anordnung der graphischen Elemente an.

- 8 (x-axis ISA chunk)
- 9 (y-axis ISA chunk)

Die Produktionsregeln zum Erreichen der Ziele nehmen den Hauptteil der ACT-R-Programmvorlage ein und bilden die Leseregeln eines Visualisierungskonzepts in LISP-Code ab. Abbildung 6.4 zeigt ein Aktivitätsdiagramm der Produktionsregeln am Beispiel der Aufgabe „Lesen Sie bitte den Wert einer Säule im Säulendiagramm.“

Die Instanziierung des Visualisierungsschemas zu einem Visualisierungsmodell erfolgt mit Hilfe des graphischen Editors *cEdit*, mit dem Bereiche auf dem Stimulus mit Elementen des Wissensmodells verbunden werden (Abbildung 6.5). Im Gegensatz zu einem AOI-Editor bei einem Eye-Tracking-Experiment, in dem meist sehr großflächig AOIs definiert werden (Abbildung 5.4), müssen allerdings für die Simulation der visuellen Suche die graphischen Elemente im Stimulus räumlich sehr exakt markiert werden. Neben der manuellen Markierung der graphischen Elemente kann auch ein automatisches Verfahren eingesetzt werden (Kapitel 5.3).

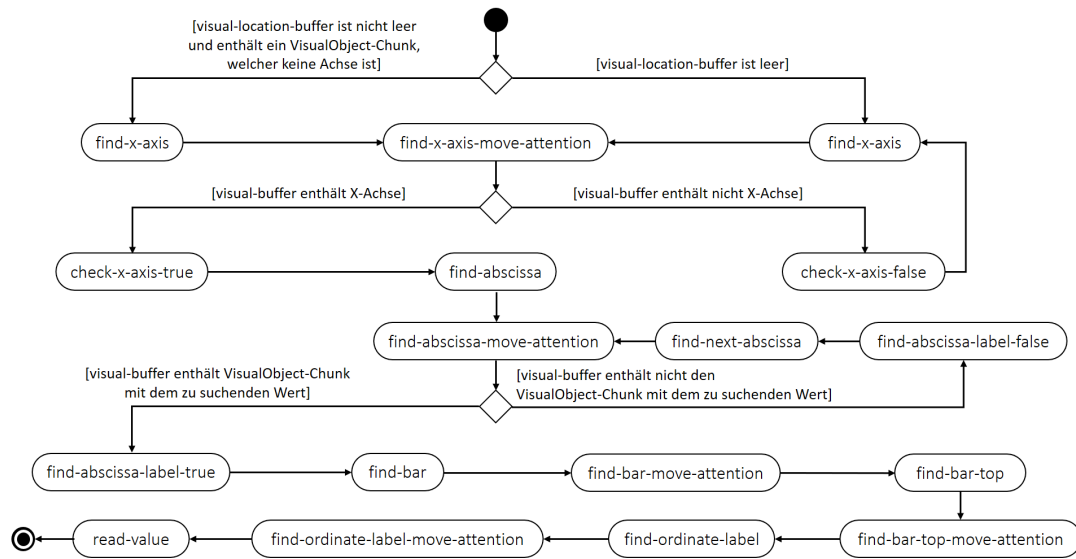


Abbildung 6.4: Aktivitätsdiagramm der Produktionsregeln für die Aufgabe: „Lesen Sie bitte den Wert einer Säule im Säulendiagramm.“

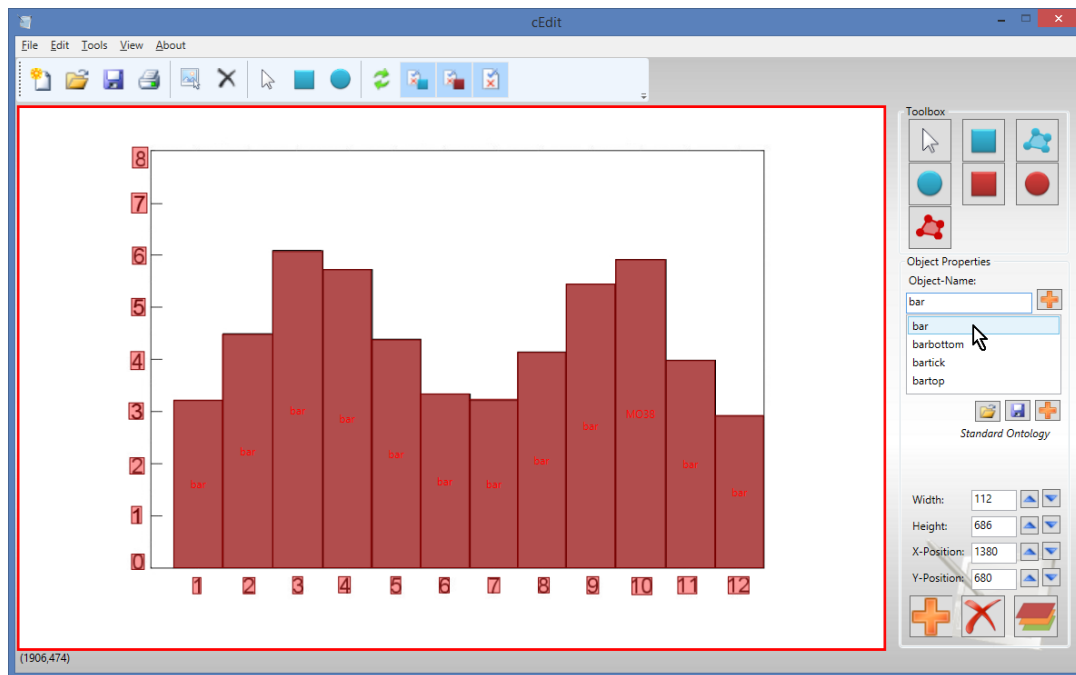


Abbildung 6.5: Mit dem Programm *cEdit* werden Bereiche auf dem Stimulus mit semantischer Information annotiert.

cEdit erzeugt ein Overlay über den eigentlichen Stimulus, in dem gespeichert ist, in welchen Bereichen auf dem Stimulus welche Klasse von graphischen Elementen dargestellt werden. Zusätzlich werden Informationen gespeichert, die für die Instanziierung des Visualisierungsschemas wichtig sind. Beispielsweise ist der Text einer Beschriftung wichtig, wenn gezielt nach einem bestimmten benannten Element im Stimulus gesucht werden soll. Das Overlay wird anschließend in einer Datei gespeichert (siehe XML-Datei im Listing 6.3) und steht der Kognitionssimulation zur Verfügung. An den passenden Stellen in der ACT-R-Programmvorlage (Listing 6.4) werden die `VISUAL_LOCATION_CHUNKS` und `VISUAL_OBJECT_CHUNKS` durch Informationen aus dem Overlay ersetzt (Listings 6.5 und 6.6).

Listing 6.3: Bereiche auf dem Stimulus werden mit Klassen aus der Visualisierungsontologie identifiziert und in einer XML-Datei gespeichert. Dazu werden neben dem Klassennamen (Class) auch räumliche Koordinaten der graphischen Elemente (X, Y) und Zusatzinformationen wie Text einer Beschriftung (Text) mit angegeben.

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Motif Name="Example" Scale="50">
<Bitmap File="C:\barchart.jpg" />
<Rectangle Class="bar" X1="562" Y1="314" X2="674" Y2="1022" />
<Rectangle Class="bar" X1="340" Y1="648" X2="452" Y2="1022" />
<Rectangle Class="bar" X1="450" Y1="500" X2="564" Y2="1024" />
...
<Rectangle Class="yaxis" X1="290" Y1="92" X2="296" Y2="1026" />
<Rectangle Class="xaxis" X1="290" Y1="1020" X2="1654" Y2="1028" />
<Rectangle Class="ticklabel" Text="8" X1="256" Y1="90" X2="284" Y2="132" />
<Rectangle Class="ticklabel" Text="7" X1="252" Y1="192" X2="284" Y2="232" />
<Rectangle Class="ticklabel" Text="6" X1="252" Y1="312" X2="280" Y2="348" />
<Rectangle Class="ticklabel" Text="5" X1="250" Y1="424" X2="278" Y2="462" />
<Rectangle Class="ticklabel" Text="4" X1="246" Y1="544" X2="276" Y2="580" />
...
</Motif>
```

Listing 6.4: `VISUAL_LOCATION_CHUNKS` und `VISUAL_OBJECT_CHUNKS` werden durch das Framework mit Informationen aus dem Overlay ersetzt.

```
(define-chunks {VISUAL_LOCATION_CHUNKS})
(define-chunks {VISUAL_OBJECT_CHUNKS})
```

Listing 6.5: Beispiel für die Ersetzung der `VISUAL_LOCATION_CHUNKS` mit Informationen aus der Overlay-Datei für ein Säulendiagramm.

```
(visual-location-chunks
 (define-chunks
 (ISA visual-location screen-x 202 screen-y 328 kind bar value bar
 width 55 height 348 color black)
```

```
(ISA visual-location screen-x 202 screen-y 154 kind bar-top value
  bar-top width 55 height 1 color black)
(ISA visual-location screen-x 93 screen-y 410 kind bar value bar
  width 55 height 184 color black)
...
(ISA visual-location screen-x 42 screen-y 274 kind axis value axis
  width 3 height 459 color black)
(ISA visual-location screen-x 376 screen-y 503 kind axis value axis
  width 671 height 4 color black)
(ISA visual-location screen-x 31 screen-y 54 kind tick-label value
  "8" width 13 height 20 color black)
(ISA visual-location screen-x 30 screen-y 104 kind tick-label value
  "7" width 15 height 20 color black)
...))
```

Listing 6.6: Beispiel für die Ersetzung der VISUAL_OBJECT_CHUNKS mit Informationen aus der Overlay-Datei für ein Säulendiagramm.

```
(visual-object-chunks
  (define-chunks (ISA bar value "Saeule" width 55 height 348 color
    black screen-x 202 screen-y 328)
    (ISA bar-top value "Saeuloberkante" width 55 height 1 color black
      screen-x 202 screen-y 154)
    (ISA bar value "Saeule" width 55 height 184 color black screen-x 93
      screen-y 410)
    ...
    (ISA axis value "Y-Achse" width 3 height 459 color black type
      y-axis)
    (ISA axis value "X-Achse" width 671 height 4 color black type
      x-axis)
    (ISA tick-label value "8" width 13 height 20 color black)
    (ISA tick-label value "7" width 15 height 20 color black)
    ...))
```

Kognitionssimulation

Die Kognitionssimulation verwendet das Input-Modell, um die visuelle Suche mit den folgenden vier Schritten zu simulieren:

1. Abgleich des Schemas
2. Identifizierung von aufgabenrelevanten graphischen Elementen
3. Simulation der visuellen Suche
4. Physikalische Simulation der Augenbewegung

Zum Schluss erfolgt die Ausgabe des berechneten Scan-Paths.

1.) Abgleich des Schemas Der erste Schritt bei der Simulation der visuellen Suche besteht darin, das Visualisierungskonzept eines vorliegenden Stimulus zu identifizieren. Für die Simulation dieses Schritts können Ergebnisse aus Eye-Tracking-Experimenten einfließen. Damit können zum einen Bereiche auf einem Stimulus erkannt werden, die für die Identifikation eines Visualisierungsschemas von Bedeutung sind. Zum anderen liefern diese Experimente aber auch Zeitdauern für die Erkennung eines Visualisierungsschemas. Alle graphischen Elemente der im Wissensmodell abgelegten Visualisierungsschemata werden mit dem gegebenen Stimulus verglichen. Je mehr graphische Elemente auf dem Stimulus mit den in einem Visualisierungsschema abgespeicherten graphischen Elementen übereinstimmen, umso wahrscheinlicher ist es, dass dieses Visualisierungsschema vorliegt. Als Ergebnis liefert dieser Schritt einen Wert für die Wahrscheinlichkeit, dass ein bestimmtes Visualisierungsschema vorliegt.

Dieser Schritt ist bisher nur sehr rudimentär im Prototyp implementiert. Der Benutzer muss aktuell manuell ein Visualisierungsschema für die Simulation vorgeben.

2.) Identifikation der aufgabenrelevanten graphischen Elemente

Wenn ein Visualisierungsschema erkannt worden ist, werden im Visualisierungsschema die graphischen Elemente identifiziert, die betrachtet werden müssen, um die gegebenen Aufgaben erfolgreich durchführen zu können. Die Identifizierung von aufgabenrelevanten graphischen Elementen wird mit Hilfe der Aufgabenbeschreibung durchgeführt. Für ein Säulendiagramm, bei dem die Kategorien der Säulen mit Zahlen beschriftet sind, könnte die Aufgabenstellung beispielsweise lauten:

Welche Höhe hat die Säule mit der Beschriftung '6'?

Diese kann in Form von Tupel beschrieben werden (Listing 6.7).

Listing 6.7: Die Aufgabenbeschreibung wird für die Simulation der visuellen Suche in Tupeln beschreiben.

```
1 (label1, label)
2 (label2, label)
3 (axis1, axis)
4 (bar1, bar)
5 (bartop1, bartop)
6
7 (axis1, label1)
8 (label1, text="6")
9 (label1, bar1)
10 (bar1, bartop1)
```

```

11 (bartop1, label2)
12 (label2, text=?)

```

Die erste Gruppe an Tupeln (Zeile 1 bis 5) definiert alle Variablen, die in der Frage verwendet wurden. Dazu werden die Variablen mit Hilfe von Klassen aus der Visualisierungsontologie definiert. Die zweite Gruppe (Zeile 6 bis 11) beschreibt die räumliche Suchstrategie und welche Variablen nacheinander gesucht werden müssen. Das Visualisierungsmodell wird anschließend nach diesen Variablen untersucht. Über die im Visualisierungsschema definierten Leseregeln wird dann im nächsten Schritt die visuelle Suchstrategie erstellt.

3.) Simulation der visuellen Suche Die Qualität der Simulationsergebnisse zur visuellen Suche hängt maßgeblich von der Modellierung der visuellen Suchprozesse nach graphischen Elementen auf einem Stimulus ab. Das Ziel der visuellen Suche ist es, mit Hilfe des Visualisierungsschemas eine Abfolge von Fixationen und Sakkaden zu berechnen, um alle aufgabenrelevanten graphischen Elemente im Stimulus zu fokussieren. Aus dem Visualisierungsschema leitet die Simulation die Bewegungsrichtung (nach Oben, Unten, Rechts, Links usw.) ab und springt von einem graphischen Element im Visualisierungselement so lange zum nächsten bis das nächste aufgabenrelevante graphische Element gefunden wird.

Meist führt jedoch die serielle Fokussierung aller im Visualisierungsschema angesprungenen graphischen Elemente nicht automatisch zur effizientesten Suchstrategie. Erlernete visuelle Suchstrategien für bestimmte Visualisierungskonzepte und Aufgaben führen zu visuellen Abkürzungen. Abbildung 6.6 zeigt dazu zwei Beispiele. Abbildung 6.6a) zeigt einen Scan-Path, der jede Kategorie auf der X-Achse einzeln fokussiert, um die gesuchte Säule der Kategorie F zu finden. Die Fokussierung von allen Beschriftungselementen ist deshalb notwendig, weil die Beschriftung keiner Ordnung folgt. Abbildung 6.6b) zeigt ein Säulendiagramm mit alphabetischer Beschriftung der Kategorien. Hier kann eine Abkürzung bei der visuellen Suche nach der Kategorie F angewendet werden. Um die Ergebnisse der Simulation weiter zu verbessern, können auch Gestaltgesetze (Wertheimer, 1923) oder Effekte der präattentiven Wahrnehmung beachtet werden.

Durch den modularen Aufbau des Frameworks können hierzu die Simulationsprogramme beliebig erweitert werden. Auch besteht die Möglichkeit, verschiedene visuelle Suchstrategien für eine gleiche Aufgabe zu simulieren.

4.) Physikalische Simulation der Augenbewegung Der letzte Schritt ist die physikalische Simulation der Augenbewegung entlang der aus dem

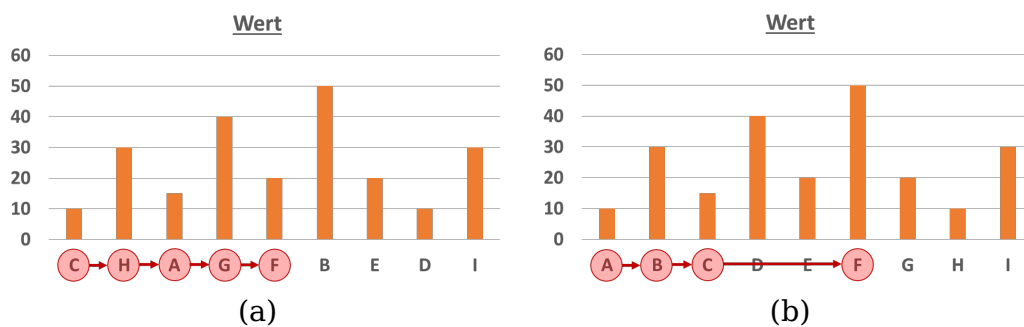


Abbildung 6.6: Abkürzung bei der visuellen Suche: a) Da keine geordnete Beschriftung vorliegt, muss jede Kategorie im Säulendiagramm einzeln fokussiert werden, um die gesuchte Kategorie F zu finden. b) Das Säulendiagramm hat eine alphabetisch geordnete Beschriftung. Hier kann eine Abkürzung bei der visuellen Suche nach der Kategorie F verwendet werden.

dritten Schritt ermittelten Richtungen. Aus dem Visualisierungsmodell erhält die Simulation hierzu die exakten Koordinaten der einzelnen graphischen Elemente auf dem Stimulus.

cSim nutzt für die Simulation der physikalischen Augenbewegungen das ACT-R-Modul EMMA (**E**ye **M**ovements and **M**ovement of **A**ttention), von Salvucci (Salvucci, 2001), um Fixationspositionen während der Durchführung einer visuellen Suche zu berechnen. EMMA sagt Zeitdauern und Positionen der Fixationen voraus und beachtet dabei Fixationsfrequenzeffekte und perzeptive Eigenschaften der Fovea. Damit verbindet EMMA höher angesiedelte kognitive Prozesse mit tiefer angesiedelten Prozessen der Perzeption.

Ausgabe des Scan-Paths Die Ausgabe des simulierten Scan-Paths erfolgt in Form einer Scan-Path-Visualisierung und einer CSV-Datei (Listing 6.8) mit zusätzlichen Informationen zur Klasse der fokussierten graphischen Elemente.

Listing 6.8: Eine Liste von simulierten Fixationen kann als CSV-Datei für die weitere Verarbeitung exportiert werden.

```
Zeit;X;Y;Class;
0,337;408;408;bar;
0,510;373;506;;
0,630;376;503;xaxis;
0,857;151;530;;
1,029;93;520;ticklabel;
...
```

6.1.3 Anwendung des Simulations-Frameworks

In diesem Abschnitt wird die Anwendung von cSim anhand von zwei verschiedene Aufgaben mit Säulendiagrammen demonstriert. Dazu wird der Verlauf der simulierten Scan-Paths qualitativ mit Scan-Paths verglichen, die in einer Pilotstudie mit sieben Probanden aufgenommen worden sind. Das Eye-Tracking-Experiment hatte einen wie in Kapitel 2.4.1 beschriebenen Aufbau. Insgesamt wurden im Benutzerexperiment jedem Probanden neun verschiedene Säulendiagramme gezeigt. Jeder Proband führte mit jedem Stimulus die folgenden zwei Aufgaben durch:

1. Welche Höhe hat die Säule X?
2. Welche Säule hat den höheren Wert? Säule X oder Säule Y?

X und Y wurden dabei durch den Beschriftungstext der Säulen ersetzt.

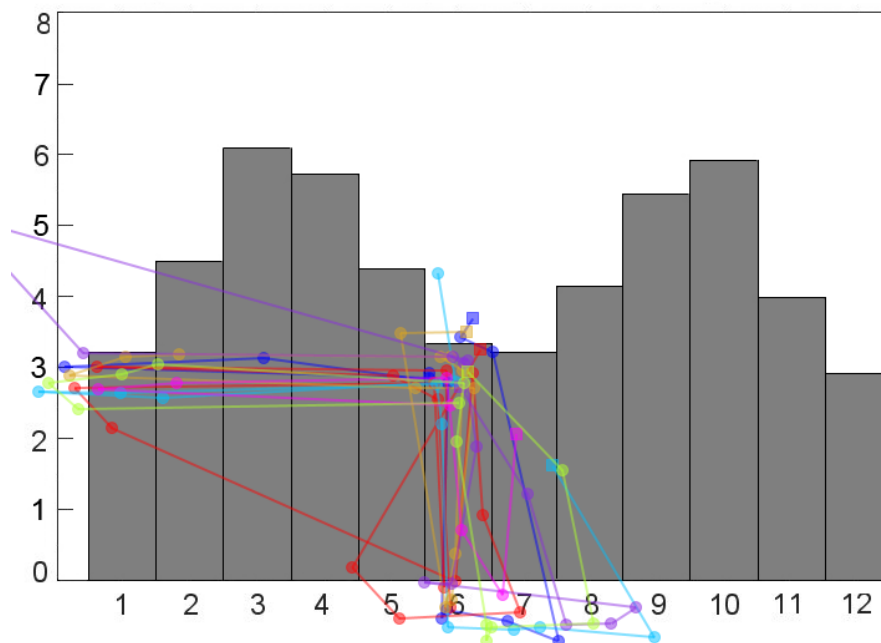
Für die Demonstration von cSim für die erste Aufgabe der Studie wird im Folgenden das in Abbildung 6.7a) gezeigte Säulendiagramm verwendet, für die zweite Aufgabe das in Abbildung 6.8a) dargestellte Diagramm. Beide Diagramme wurden dazu semantisch annotiert.

Welche Höhe hat Säule X? Abbildung 6.7a) zeigt alle sieben im Eye-Tracking-Experiment aufgezeichneten Scan-Paths für die Aufgabe:

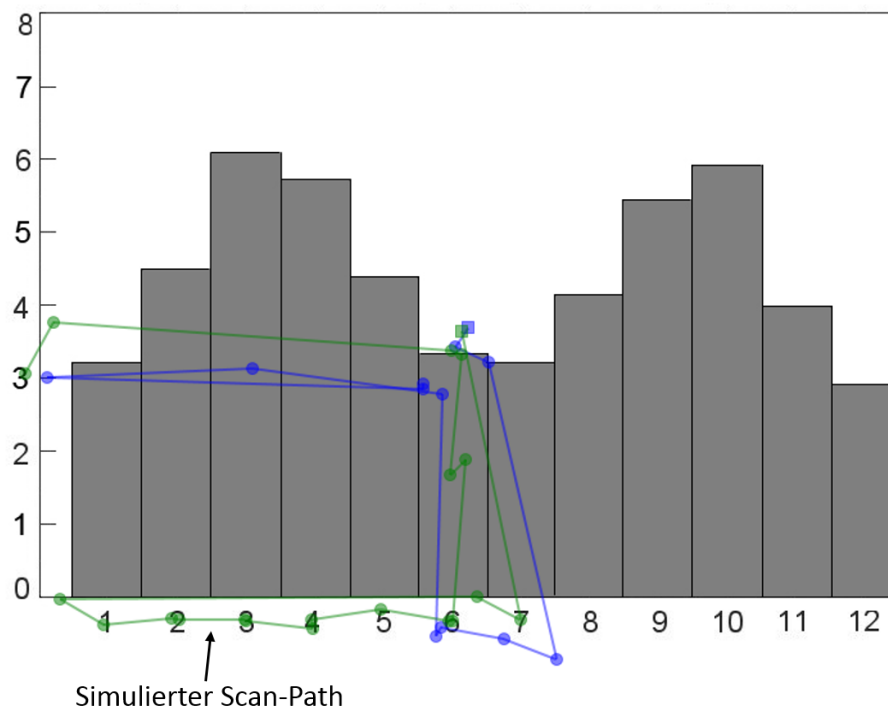
Welche Höhe hat die Säule mit der Beschriftung '6'?

Der simulierte Scan-Path (grün) ist zusammen mit einem ausgewählten Scan-Path aus dem Experiment (blau) in Abbildung 6.7b) visualisiert.

In der Abbildung ist zu erkennen, dass beide Scan-Paths in der Mitte des Stimulus starten und dann mit wenigen Fixationen schnell auf die horizontale Achse springen. Hier ist dann jedoch ein deutlicher Unterschied zwischen dem simulierten und dem aufgenommenen Scan-Path zu erkennen. Die Simulation erkennt die horizontale Achse der Visualisierung und springt sofort ganz nach links um von dort aus Fixation für Fixation den Beschriftungstext '6' auf der horizontalen Diagrammachse zu suchen. Der Proband findet sofort die Beschriftung '7' und sucht dann in der Nähe nach dem Beschriftungstext '6', den er mit zwei weiteren Fixationen auch findet. Damit wendet der Proband eine Heuristik an, die von cSim in der vorliegenden Version noch nicht simuliert wird. Sobald sowohl die Simulation als auch der Proband die richtige Kategorie gefunden haben, springen beide in die Nähe der oberen Kante der Säule '6' und lesen dann anschließend den Wert dieser Säule auf der vertikalen Achse des Diagramms ab. Der



(a)



(b)

Abbildung 6.7: a) Sieben Scan-Paths aus einer Pilotstudie für die Aufgabe „Welche Höhe hat die Säule mit der Beschriftung '6'?“ b) Mit cSim simulierter Scan-Path (grün) zur gleichen Aufgabe zusammen mit einem ausgewählten Scan-Path (blau) aus dem Experiment.

Proband überprüft anschließend nochmals den abgelesenen Wert. Dieses Verhalten wird durch cSim in der vorliegenden Version nicht simuliert.

Welche Säule hat den höheren Wert? Säule X oder Säule Y? Abbildung 6.8a) zeigt alle sieben im Eye-Tracking-Experiment aufgezeichneten Scan-Paths für die Aufgabe:

Welche Säule ist höher, die für die Kategorie X-Box oder für Playstation?

Der simulierte Scan-Path (grün) ist zusammen mit zwei ausgewählten Scan-Paths aus dem Experiment (blau und hellblau) in Abbildung 6.8b) visualisiert.

Zu erkennen ist, dass der simulierte Scan-Path und der blaue Scan-Path eines Probanden in der Mitte des Stimulus starten und dann auf der horizontalen Achse verlaufen. Der blaue Scan-Path zeigt einen kurzen Ausreisser zur Säule „Gameboy“. Da die Achsenbeschriftung nicht alphabetisch vorliegt, muss der blaue Proband alle Beschriftungstexte lesen, um die Kategorie „X-Box“ und „Playstation“ zu finden. Dieses Verhalten wird korrekt von cSim simuliert. Beim Erkennen der Höhe der Säule „X-Box“ findet sich ein Unterschied zwischen dem blauen Probanden und dem simulierten Scan-Path. cSim berechnet dazu Fixationen zur Suche nach der Oberkante der Säule. Dann folgen sowohl beim blauen Probanden als auch beim simulierten Scan-Path eine Sakkade zur Oberkante der Säule „Playstation“. Die Abbildung zeigt zusätzlich einen zweiten Probanden in Hellblau. Dieser zeigt ein grundsätzlich ähnliches Verhalten wie von der Simulation berechnet. Die vertikale Verschiebung der Fixationen des Scan-Paths kann auf eine unzureichende Kalibration des Eye-Trackers zurückgeführt werden.

6.1.4 Diskussion und Vergleich mit existierenden Arbeiten

Das wichtigste Ziel bei der Entwicklung von cSim bestand darin, für die Evaluation von Visualisierungen ein flexibles und effizientes Werkzeug bereitzustellen, mit der die visuelle Suche simuliert werden kann. Je genauer die simulierten Augenbewegungen den Augenbewegungen von realen Probanden entsprechen, umso stärker können Simulationenwerkzeuge Schritt für Schritt aufwändige Benutzerexperimente in der Visualisierungsforschung ersetzen.

Die Demonstration von cSim zur Simulation der visuellen Suche für verschiedene Säulendiagramme und Aufgaben zeigt, dass dieser Ansatz prinzipiell funktioniert. AOIs müssen nur noch auf den jeweiligen Stimuli definiert werden, sobald Produktionsregeln für ein gegebenes Visualisierungskonzept und eine visuelle Aufgabe implementiert sind. Dieser

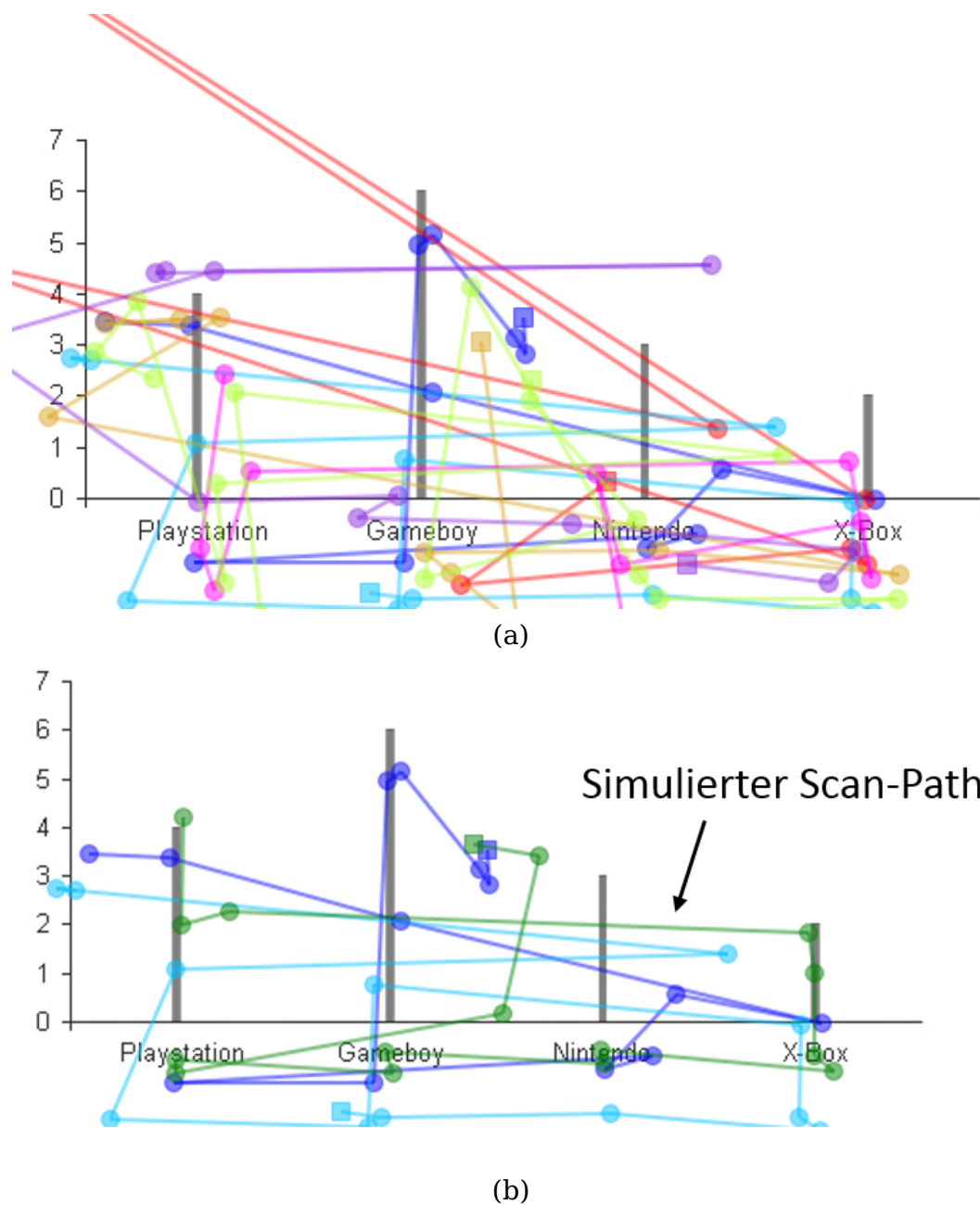


Abbildung 6.8: a) Sieben Scan-Paths aus einem Eye-Tracking-Experiment für die Aufgabe „Welche Säule ist höher, X-Box oder Playstation?“ b) Mit cSim simulierter Scan-Path (grün) zur gleichen Aufgabe zusammen mit zwei ausgewählten Scan-Paths (blau und hellblau) aus dem Experiment.

Ansatz ermöglicht es dem Entwickler einer Visualisierung, sich nahezu vollständig auf das Design einer Visualisierung konzentrieren zu können und die Evaluation seiner Visualisierungstechnik mit cSim durchzuführen. Benutzerexperimente mit wenigen Probanden müssen dann nur noch zur Verifizierung und Verfeinerung der Ergebnisse durchgeführt werden. Der Aufwand zur Vorbereitung der Stimuli kann noch weiter verringert werden, wenn der in Kapitel 5.3 vorgestellte Ansatz zur automatischen Annotierung von graphischen Elementen verwendet wird.

Im aktuellen Prototypen wurden die wichtigsten Komponenten des in diesem Kapitel vorgestellten Modells zur Simulation der visuellen Suche implementiert. Damit umfasst der vorliegende Prototyp nicht den vollen im Modell beschriebenen Funktionsumfang und besitzt daher noch Einschränkungen in der Demonstrationsfähigkeit. Zum einen wird angenommen, dass jedes graphische Element in einer Visualisierung nur einmal fokussiert wird, um damit seinen Informationsgehalt zu erkennen und kognitiv zu verarbeiten. Sowohl die Ergebnisse der Pilotstudie aus den Kapiteln 4.3.2 und 5.4.2 als auch die Analyse der Eye-Tracking-Daten aus den Benutzerexperimenten in Abschnitt 6.2.4 zeigen, dass dies eine sehr starke Idealisierung ist. In der Regel werden mehrere Fixationen auf einem graphischen Element direkt hintereinander ausgeführt oder das graphische Element wird abermals fokussiert, nachdem weitere graphische Elemente betrachtet worden sind. Dieser Effekt wird als „Cross-Checking“ bezeichnet (Burch et al., 2011; Raschke et al., 2014b) und findet sich typischerweise bei komplexeren visuellen Aufgaben. Das vorliegende Simulationsmodell beschränkt sich auf zweidimensionale Balkendiagramme. Da typische Informationsvisualisierungen eine komplexere Struktur besitzen, muss untersucht werden, ob der präsentierte Ansatz für diese Visualisierungen auch anwendbar ist.

Halverson und Hornof führen zwei weitere Eigenschaften auf, die ein minimales Modell der visuellen Suche besitzen sollte (Halverson und Hornof, 2007). Die erste Eigenschaft ist, dass Augenbewegungen meist zu graphischen Elementen durchgeführt werden, die sich in der Nähe des Startpunktes einer Sakkade befinden. Diese Eigenschaft wurde im Prototypen grundsätzlich umgesetzt. Die Sakkaden zielen jeweils immer auf das nächstliegende graphische Element. Die zweite Eigenschaft ist, dass die nächste Sakkade immer erst dann ausgeführt wird, wenn das graphische Element erkannt und verstanden wurde. Auch diese Eigenschaft wird von cSim umgesetzt.

Im Gegensatz zu existierenden Arbeiten (Kapitel 2.7.7) für die Simulation von Augenbewegungen für Aufgaben mit Visualisierungen wurde in dieser Arbeit auch das Ziel verfolgt, den Benutzer bei der Entwicklung des Programmcodes für die Simulation maximal zu unterstützen.

6.2 Operatorenbasiertes Diagram-Viewing-Modell

Die zweite Methode zur Simulation der visuellen Suche ist ein operatorenbasiertes Modell zur Vorhersage von Durchführungszeiten für visuelle Aufgaben mit Visualisierungen, die ein kartesischen Layout besitzen. Ergebnisse aus Eye-Tracking-Experimenten zeigen, dass eine visuelle Aufgabe in einzelne Teilaufgaben zerlegt werden kann. Diese Eigenschaft wird für ein einfaches *Diagram-Viewing-Modell (DVM)* verwendet, das in Abschnitt 6.2.1 vorgestellt wird. Die Grundannahmen und Modellparameter des DVMs wurden in zwei Benutzerexperimenten überprüft (Abschnitt 6.2.4). Die Ergebnisse zeigten jedoch, dass die Anwendung eines operatorenbasierten Modells zur Berechnung von Durchführungszeiten von visuellen Aufgaben stark eingeschränkt ist. Abschnitt 6.2.5 diskutiert die Einschränkungen des DVMs im Detail.

6.2.1 Das Diagram-Viewing-Modell

Das DVM basiert auf zwei Grundannahmen:

1. Graphische Elemente in einer Visualisierung werden in einer seriellen Reihenfolge betrachtet. Die einzelnen Schritte dieser Reihenfolge werden im DVM als *Visual Unit Tasks* bezeichnet.
2. Ähnlich zum KLM (Kapitel 2.7.4) wird die Aneinanderreihung der einzelnen Visual Unit Tasks mit einer Modellgleichung modelliert und als visuelle Suchstrategie interpretiert, mit der Probanden eine visuell enkodierte Information lesen.

Das DVM definiert in seiner ersten Version vier Klassen von Operatoren für die Visual Unit Tasks:

- \mathcal{A} : Lesen eines Datenpunktes auf einer Koordinatenachse.
- \mathcal{V} : Lesen eines bestimmten Datenpunktes aus einem Satz von Datenpunkten.
- \mathcal{F} : Fokussierung einer Position in der Visualisierung.
- \mathcal{M} : Mentale Aktivität, um die wahrgenommene visuelle Information zu verarbeiten.

Lesen des Wertes eines Datenpunktes auf einer Koordinatenachse (\mathcal{A})

Die Analyse von Eye-Tracking-Experimenten zeigt, dass Probanden nacheinander die verschiedenen Werte eines Datenpunktes, bezüglich der gegebenen Koordinatenachsen lesen. Dieses Verhalten wird im DVM durch den Operator \mathcal{A} abgebildet, der den Visual Unit Tasks zur Identifikation und zum Lesen eines gegebenen Datenpunktes auf einer Koordinatenachse repräsentiert. \mathcal{A} wird für horizontale und vertikale Koordinatenachsen mit Skalenstrichen und ohne Skalenstriche definiert (siehe dazu auch Abbildungen 6.9a+b):

- \mathcal{A}_{hns} : Horizontale Koordinatenachse ohne Skalenstriche
- \mathcal{A}_{hws} : Horizontale Koordinatenachse mit Skalenstrichen
- \mathcal{A}_{vns} : Vertikale Koordinatenachse ohne Skalenstriche
- \mathcal{A}_{vws} : Vertikale Koordinatenachse mit Skalenstrichen

Lesen des Wertes eines Datenpunktes mit einer bestimmten Eigenschaft (\mathcal{V})

Bei den meisten Aufgaben, die mit Hilfe von Visualisierungen durchgeführt werden, müssen Beziehungen der in Visualisierungen dargestellten Daten gefunden werden. $\mathcal{V}[\text{prop}]$ modelliert dazu den Leseprozess eines Wertes in einer Visualisierung, der durch einen Datenpunkt mit einer bestimmten Eigenschaft *prop* dargestellt wird. Dazu werden von $\mathcal{V}[\text{prop}]$ zwei Varianten definiert (siehe dazu auch Abbildungen 6.9c+d):

- $\mathcal{V}[\text{out of}]$: Lesen eines Datenpunktes in einer Menge von Datenpunkten.
- $\mathcal{V}[\text{highest of } n]$: Lesen des Wertes eines Datenpunktes mit der höchsten vertikalen Position in einer Menge von n Datenpunkten.

Wie im Fall des Operators \mathcal{A} wird dabei zwischen Koordinatenachsen ohne Skalenstrichen (\mathcal{V}_{vns} , \mathcal{V}_{hns}) und mit Skalenstrichen (\mathcal{V}_{vws} , \mathcal{V}_{hws}) unterschieden. Im Fall von $\mathcal{V}[\text{highest of } n]$ kann das DVM diesen Visual Unit Task auch für Koordinatenachsen mit einem Gitter modellieren. Dieser Operator wird durch das Symbol \mathcal{V}_{wg} im DVM repräsentiert.

Fokussierung

Visuelle Suchstrategien benötigen in der Regel viele Fixationen auf einem Stimulus. Eine einzelne Fixation kann mit Hilfe des Modelloperators

\mathcal{F} modelliert werden. In der Regel dauert eine Fixation zwischen 180 ms und 275 ms (Rayner, 1998). \mathcal{F} modelliert nur den reinen Fixationsprozess (Sakkade plus Fixation) ohne eine Akkomodierungsphase. Im DVM wird $\mathcal{F} = 200\text{ ms}$ gesetzt.

Mentale Aktivität

Falls notwendig, können mentale Prozesse mit dem Modelloperator \mathcal{M} modelliert werden. Zum Beispiel müssen mentale Prozesse ausgeführt werden, um eine bestimmte Position eines graphischen Elements in der Visualisierung aus dem Gedächtnis abzurufen oder um sich an den Wert eines zuvor gelesenen Datenpunktes zu erinnern. Wie im KLM ist die Modellierung dieser Prozesse durch \mathcal{M} eine Vereinfachung. Die Originalarbeit von Card et al. schlägt einen Wert für \mathcal{M} von $1,35\text{ s}$ vor. Allerdings muss in weiteren Experimenten überprüft werden, inwieweit ein konstanter Wert von $1,35\text{ s}$ auch bei der Berechnung von Durchführungszeiten von visuellen Aufgaben zu einer exakten Vorhersage führt.

Modellgleichung

Eine visuelle Suchstrategie wird durch das Aneinanderhängen der Operatoren \mathcal{O}_i für alle Visual Unit Tasks modelliert:

$$\sum_i \mathcal{O}_i = \sum_j \mathcal{A}_j + \sum_k \mathcal{V}_k + \sum_l \mathcal{F}_l + \sum_m \mathcal{M}_m$$

Heuristik zum Erstellen der Modellgleichung

Im Gegensatz zum KLM (Kapitel 2.7.4), in dem die einzelnen Interaktionsschritte die Reihenfolge der Operatoren definieren, ist beim DVM die Aneinanderreihung der Operatoren komplexer. Grundannahme des DVMs ist es, dass die Modellgleichung einen idealen Scan-Path zu einer bestimmten visuellen Aufgabe darstellt (Kapitel 2.5.1). Dabei wird angenommen, dass der Betrachter ein *Experte* ist. Dies bedeutet, dass er sehr erfahren mit der jeweiligen Visualisierungstechnik ist. Das DVM in der aktuellen Version modelliert keine kognitiven Verstehensprozesse sowie keine Fehler, die aufgrund von Wahrnehmungseinschränkungen, wie Farbenblindheit oder Sehschwäche, entstehen können.

Um die Modellgleichung für eine gegebene visuelle Aufgabe aufzustellen, müssen folgende Schritte durchgeführt werden:

1. Identifikation aller Bereiche auf dem Stimulus, die betrachtet werden müssen, um eine gegebene Aufgabe korrekt bearbeiten zu können.
2. Auflistung aller möglichen Transitionen zwischen diesen Gebieten.
3. Auswahl der Reihenfolge von Transitionen aus Schritt 2, die zu einer semantisch korrekten Interpretation der graphisch dargestellten Information führen. Jede dieser Reihenfolgen führt zu einer möglichen visuellen Suchstrategie.
4. Beschreibung jedes einzelnen Schritts dieser visuellen Suchstrategien durch einen Visual Unit Task.
5. Zuordnung eines geeigneten Operators aus dem DVM zu jedem Visual Unit Task und Formulierung der Modellgleichung.
6. Wenn eine Koordinatenachse Zahlenstriche besitzt, der zu lesende Wert sich jedoch zwischen zwei Skalenstrichen befindet, müssen in der Modellgleichung die Modelloperatoren \mathcal{A}_{vws} und \mathcal{A}_{vns} im Falle einer vertikalen Koordinatenachse und \mathcal{A}_{hws} und \mathcal{A}_{hns} im Falle einer horizontalen Koordinatenachse hintereinander aufgeführt werden.
7. Wenn eine Fixation auf dem Stimulus aufgabenrelevant ist, jedoch nicht durch die Operatoren \mathcal{A} oder \mathcal{V} modelliert werden kann, wird der Modelloperator \mathcal{F} in die Modellgleichung eingesetzt.
8. Wenn eine Information, beispielsweise die Position oder der Wert eines Datenpunktes, aus dem Gedächtnis abgerufen werden muss, wird der Operator \mathcal{M} für diesen Schritt in die Modellgleichung eingesetzt. \mathcal{M} ist nicht notwendig, wenn ein Wert im Gedächtnis behalten werden muss, beispielsweise für einen späteren Vergleich. Dieser Vorgang wird bereits durch die Modelloperatoren \mathcal{A} und \mathcal{V} abgedeckt.

6.2.2 Ermittlung der Modellparameter

Um das Modell in der Praxis einsetzen zu können, wurde ein Benutzerexperiment mit 27 Probanden durchgeführt. Das Ziel des Benutzerexperiments war es, empirisch Durchführungszeiten für die Modelloperatoren \mathcal{A} und \mathcal{V} zu messen. Dazu wurde für jeden Modelloperator ein eigenes Experiment durchgeführt. Im Folgenden werden die Abweichungen des Benutzerexperiments vom in Kapitel 2.4.1 beschriebenen Aufbau und Ablauf beschrieben.

Aufbau des Benutzerexperiments

Insgesamt wurden jedem Probanden 160 Stimuli gezeigt. Jeder Proband führte zwei Aufgabenblöcke mit dem gleichen experimentellen Aufbau durch. Der erste Block war aufgeteilt in zwei Abschnitte: Als erstes mussten die Probanden die Position eines Datenpunktes auf vertikalen und horizontalen Koordinatenachsen lesen. Als zweites mussten sie die Position eines Datenpunktes in einer Menge von fünf Datenpunkte angeben, die jeweils wieder auf horizontal und vertikal angeordneten Koordinatenachsen gezeichnet waren. In diesem Block wurden den Probanden 16 Unterblöcke mit jeweils 5 Stimuli gleichen Typs gezeigt. Im zweiten Block mussten die Probanden Datenpunkte mit der vertikal höchsten Position in einem Diagramm identifizieren. Dazu wurden ihnen Diagramme mit zwei, fünf und zehn Datenpunkten präsentiert. Dieser Block war wieder in 16 Unterblöcke aufgeteilt, in denen wiederum fünf Stimuli gleichen Typs gezeigt wurden. Um Durchführungszeiten für die Operatoren definieren zu können, wurden für jeden Probanden und jeden Stimulus Durchführungszeiten und Fehlerraten gemessen.

Stimuli und Aufgaben

Alle Datensätze der Datenpunkte in den Stimuli wurden synthetisch nach einem Zufallsalgorithmus generiert. Für die Darstellung der Koordinatensysteme in Form von Bitmaps wurde ein selbstentwickelter Renderer verwendet. Die Bildgröße aller Bitmaps betrug 1250 x 1250 Pixel. Die Bitmaps wurden von der verwendeten Eye-Tracking-Software auf die Bildschirmhöhe auf 1200 x 1200 Pixel skaliert. Die Koordinatenachsen und Datenpunkte in den Diagrammen waren in Schwarz eingefärbt. Der Hintergrund der Diagramme wurde in Hellgrau eingefärbt. Die Koordinatenachsen, Skalenstriche und Datenpunkte wurden in einer gut erkennbaren Liniendicke von 5 Pixeln und einer gut lesbaren Textgröße dargestellt. Sowohl die Skalenstriche in den ein- wie auch zweidimensionalen Koordinatensystemen zeigten ein Intervall von 0 bis 10 Einheiten auf den Achsen. Wenn Zahlenstriche dargestellt wurden, war ihr Intervall eine Einheit. Nur der erste Zahlenstrich und der letzte Zahlenstrich waren mit einer „0“ und einer „10“ beschriftet. Skalenstriche zwischen diesen beiden Zahlenstrichen waren nicht beschriftet. Wenn Datenpunkte beschriftet waren, wurden diese mit Buchstaben von „A“ bis „E“ für den ersten Aufgabenblock benannt. Im zweiten Aufgabenblock wurden Buchstaben aus dem gesamten Alphabet verwendet. Alle Datenpunkte wurden mit einem Zufallsalgorithmus beschriftet. Eine Auswahl der Stimuli zeigt Abbildung 6.9.

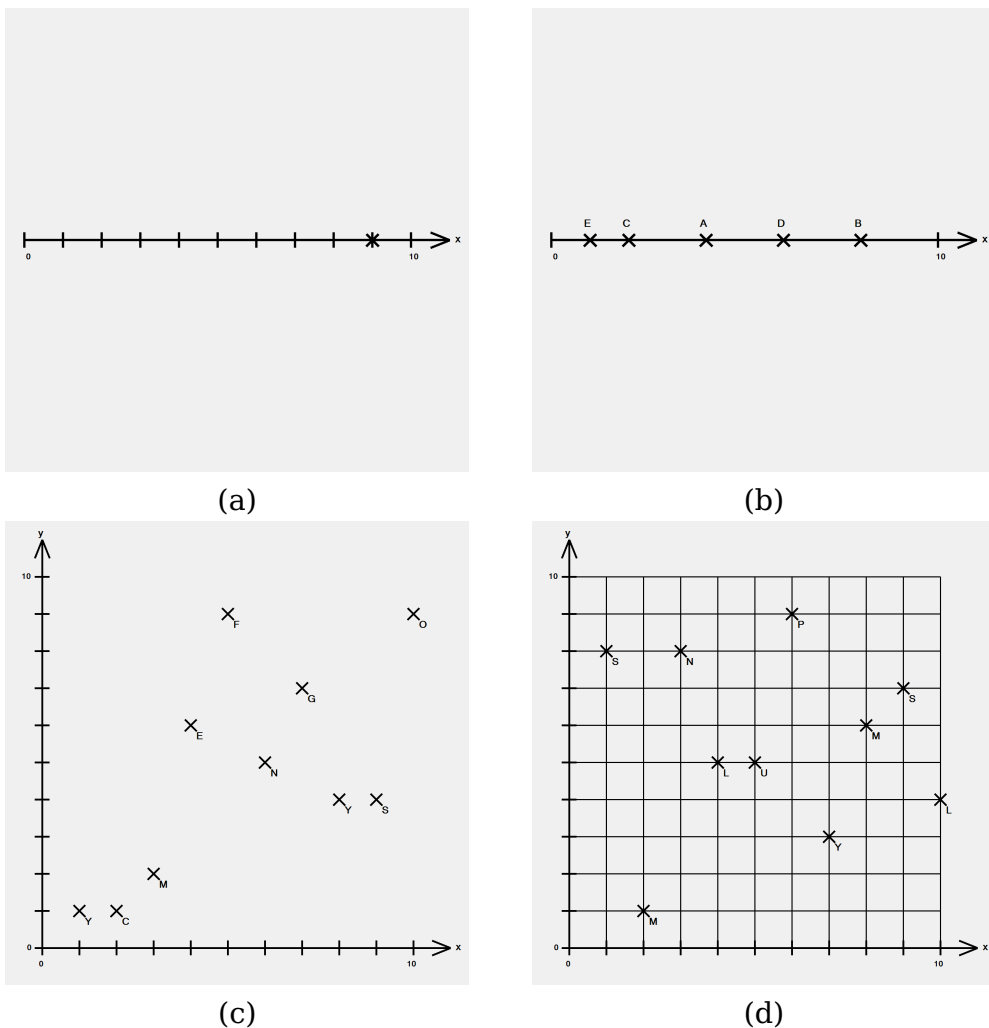


Abbildung 6.9: Beispielstimuli für den ersten Aufgabenblock (a+b) und zweiten Aufgabenblock (c+d) des Benutzerexperiments zur Bestimmung der Modellparameter.

Im Experiment zur Ermittlung der Modellparameter mussten die Probanden folgende Aufgabenblöcke durchführen:

Aufgabenblock 1: Probanden mussten die Aufgaben „Geben Sie die Position des gegebenen Punktes an.“ und „Geben Sie die Position eines Punktes aus einer Menge von fünf Punkten an.“ durchführen. Im zweiten Fall gab der Studiendurchführer dem Probanden den Punkt vor (wie beispielsweise „B“) und der Proband sollte die Position des Punktes, der mit „B“ beschriftet war, angeben. In Abbildung 6.9a+b) sind Beispielstimuli aus diesem Aufgabenblock dargestellt.

Aufgabenblock 2: Probanden mussten den Datenpunkt mit der höchsten vertikalen Position in einer Menge von zwei, fünf und zehn Datenpunkten angeben. Dazu sollten sie den Buchstaben angeben, mit dem dieser Datenpunkt beschriftet war. In Abbildung 6.9c+d) sind Beispielesstimuli aus diesem Aufgabenblock dargestellt.

Eigenschaften der Probanden

27 Probanden nahmen an diesem Experiment teil. Vier davon waren weiblich, 23 waren männlich. Das Durchschnittsalter war 28 Jahre, das Minimum 21 Jahre und das Maximum 65 Jahre. Die Probanden hatten eine weit gefächerte Berufsausbildung: von Informatik, Software-Engineering über Computer-Linguistik, Wirtschaftswissenschaften, Umwelttechnik, Medienwissenschaften bis hin zu Ingenieurwissenschaften und Mathematik. Alle Probanden hatten ein gutes Sehvermögen. Dies wurde mit einem Ishihara-Test und einem Snellen-Diagramm überprüft. Zwölf der Probanden trugen eine Brille, einer trug Kontaktlinsen. Den Probanden wurde als Ausgleich für ihre Teilnahme 10 € gezahlt. Auf einer Likert-Skala von 1 (sehr gut) bis 5 (nicht gut) gaben die Probanden im Durchschnitt ihre mathematischen Kenntnisse als 2,4 an. Zusätzlich gaben alle an, dass sie Erfahrung mit Diagrammen hatten.

6.2.3 Modellparameter

Die Tabellen 6.1 und 6.2 zeigen die Modellparameter. Tabelle 6.1 zeigt den Median und das obere (Q_{oben}) und untere Quartil (Q_{unten}) für die Modelloperatoren \mathcal{A} und $\mathcal{V}[\text{out of}]$ in Sekunden. Die Zeit des \mathcal{M} -Operators wurde auf 1.35 s gesetzt. Tabelle 6.2 zeigt die Ergebnisse einer linearen Regression für den Operator $\mathcal{V}[\text{highest of } n]$ an die Funktion $f(x) = ax + c$. Der Operator $\mathcal{V}[\text{highest of } n]$ gibt dabei die notwendige Zeit für das Lesen von n Punkten in einer Visualisierung in Sekunden an. Da im Experiment dazu nur Visualisierungen mit bis zu zehn Datenpunkten verwendet worden sind, wird für $\mathcal{V}[\text{highest of } n]$ das Intervall $[1, 10]$ angegeben. Die negative Steigung könnte durch auftretende Lerneffekte erklärt werden.

T-Tests zeigen, dass es notwendig ist, verschiedene Operatoren für die einzelnen Eigenschaften der Achsen zu verwenden. Für die t-Tests wurde die Hypothese angenommen, dass die Menge für die Operatoren \mathcal{A} und $\mathcal{V}[\text{out of}]$ jeweils identisch sind. Zusätzlich wurden alle Datensätze, die zur Definition der Operatorenzeiten verwendet werden, auf mögliche Untergruppen mit Hilfe eines Ryan-Joiner-Tests auf eine Gaußverteilung hin analysiert. Die Ergebnisse der Ryan-Joiner-Tests zeigten, dass in den aufgenommenen Datensätze keine Untergruppen existieren.

Operator	Median	Q _{unten}	Q _{oben}
\mathcal{A}_{hns}	3.6	2.6	5.0
\mathcal{A}_{hws}	2.4	1.7	3.0
\mathcal{A}_{vns}	3.2	2.6	4.1
\mathcal{A}_{vws}	1.9	1.5	2.4
$\mathcal{V}[\text{out of}]_{hns}$	3.4	2.8	4.1
$\mathcal{V}[\text{out of}]_{hws}$	2.2	1.9	2.7
$\mathcal{V}[\text{out of}]_{vns}$	3.8	2.9	4.8
$\mathcal{V}[\text{out of}]_{vws}$	2.8	2.2	3.3
\mathcal{F}	0.2	-	-
\mathcal{M}	1.35	-	-

Tabelle 6.1: Erster Satz von Modeloperatoren für \mathcal{A} , $\mathcal{V}[\text{out of}]$, \mathcal{F} und \mathcal{M} . Angabe der Operatoren erfolgt in Sekunden.

Operator	Operatorgleichung	Fehler
$\mathcal{V}[\text{highest of n}]_{ns}$	$t(n) = -0.1n + 3.5$	$R^2 = 0.99$
$\mathcal{V}[\text{highest of n}]_{ws}$	$t(n) = -0.07n + 2.2$	$R^2 = 0.89$
$\mathcal{V}[\text{highest of n}]_{wg}$	$t(n) = -0.1n + 2.3$	$R^2 = 0.92$

Tabelle 6.2: Zweiter Satz der Modeloperatoren für $\mathcal{V}[\text{highest of n}]$. Die Operatoren gelten im Definitionsbereich $[1, 10]$. Angabe der Operatoren erfolgt in Sekunden. In der rechten Spalten ist der Fehler der linearen Regression angegeben.

6.2.4 Validierung

Um den Ansatz des DVM und seine Modellparameter zu überprüfen, wurden zwei Validierungsexperimente mit Säulen-, Linien- und Flächendiagrammen durchgeführt. Im ersten Validierungsexperiment wurden vorhergesagte Zeiten aus dem DVM mit empirisch gemessenen Zeiten für neun Säulendiagramme verglichen. Im zweiten Validierungsexperiment wurden die vorhergesagten Zeiten mit fünf Flächendiagrammen und sechs Liniendiagrammen getestet.

Die Abbildungen 6.10 und 6.11 zeigen verwendete Operatoren (linke Spalte) zusammen mit Heat-Maps über alle Probanden, welche den jeweiligen Stimulus betrachtet haben (rechte Spalte). Bei der Darstellung der verwendeten Operatoren geben die Zahlen die Reihenfolge der Fixationen auf dem idealen Scan-Path an, in der später die Operatoren in der Modellgleichung nacheinander ausgeführt werden. Die Wahl der Operatoren kann zusätzlich mit Hilfe der Heat-Maps überprüft werden.

Beide Validierungsexperimente wurden als eigenständige Experimente mit Probanden, die nicht im ersten Experiment zum Aufbau der Operato-

6 Modellierung und Simulation der visuellen Suche



Abbildung 6.10: Stimuli mit Modelloperatoren (linke Spalte) und Heat-Maps aus den Validierungsexperimenten.



Abbildung 6.11: Stimuli mit Modelloperatoren (linke Spalte) und Heat-Maps aus den Validierungsexperimenten.

rensammlung teilgenommen hatten, durchgeführt. Die Validierungsexperimente hatten jedoch den gleichen Aufbau, die gleiche Umgebung und technische Ausstattung wie das Benutzerexperiment in Abschnitt 6.2.2. Im Gesamten nahmen 33 Probanden an den Validierungsexperimenten teil, zwölf im ersten Experiment und 21 im zweiten. Im ersten Validierungsexperiment waren vier der Probanden weiblich und acht männlich. Sechs der Probanden trugen eine Brille und zwei Kontaktlinsen. Im zweiten Validierungsexperiment waren 13 der Probanden weiblich und acht männlich. Fünf der Probanden trugen eine Brille. Das Durchschnittsalter der Probanden war 23 Jahre, der jüngste 19 und der älteste 58 Jahre alt. Die Probanden studierten entweder an der Universität oder hatten einen akademischen Abschluss. In beiden Validierungsexperimenten hatten die Probanden eine weit gefächerte Ausbildung von Physik, Biologie, Lebensmittelchemie, Chemie über Informatik bis hin zu Ingenieurwissenschaften. Den Probanden wurden 10 € für ihre Teilnahme gezahlt.

Die Stimuli bestanden im ersten Validierungsexperiment aus neun Säulendiagrammen mit fünf, zehn und fünfzehn Säulen. Diese Säulendiagramme wurden künstlich erzeugt mit zufälligen Werten für jede Kategorie. Die

Erstellung der Säulendiagramme erfolgte mit dem Visualisierungsframework D3. Drei der Stimuli aus dem ersten Validierungsexperiment sind in Abbildung 6.10 abgebildet. Die Probanden sollten zwei Klassen von Aufgaben mit diesen Stimuli durchführen: „Geben Sie den Namen der höchsten Kategorien der beiden Kategorien X und Y zusammen mit seinem Wert an.“ (Aufgabe A + B) und „Geben Sie den Namen der höchsten Kategorien der Kategorien X, Y und Z zusammen mit seinem Wert an.“ (Aufgabe C). Für die Modellierung dieser Aufgaben wurden die Operatoren $\mathcal{V}[\text{out of}]_{hws}$, $\mathcal{V}[\text{highest of}]_{ws}$, \mathcal{A}_{vws} , \mathcal{A}_{vns} und \mathcal{F} verwendet. Folgende Modellgleichungen bilden diese Aufgaben im DVM ab:

- $\mathcal{V}[\text{out of}]_{hws} + \mathcal{V}[\text{out of}]_{hws}$
 $+ \mathcal{V}[\text{highest of } 2]_{ws} + \mathcal{A}_{vws} = 2, 2s + 2, 2s + 1, 9s + 1, 9s = 8, 2s$
 (Aufgabe A, Abbildung 6.10a+b)
- $\mathcal{V}[\text{out of}]_{hws} + \mathcal{V}[\text{out of}]_{hws}$
 $+ \mathcal{V}[\text{highest of } 2]_{ws} + \mathcal{A}_{vws} + \mathcal{A}_{vns} = 2, 2s + 2, 2s + 1, 9s + 1, 9s + 3, 2s =$
 $11, 4s$ (Aufgabe B, Abbildung 6.10c+d)
- $\mathcal{V}[\text{out of}]_{hws} + \mathcal{V}[\text{out of}]_{hws}$
 $+ \mathcal{V}[\text{out of}]_{hws} + \mathcal{V}[\text{highest of } 3]_{ws} + \mathcal{A}_{vws} = 2, 2s + 2, 2s + 2, 2s + 1, 85s +$
 $1, 9 = 10.35s$ (Aufgabe C, Abbildung 6.10e+f)

In Abbildung 6.12 sind die durch das Modell vorhergesagten Zeiten (Kreise) zusammen mit Boxplots der gemessenen Zeiten im Experiment aufgetragen. Die durchschnittliche Differenz zwischen den vorhergesagten Werten und dem Median der gemessenen Werte beträgt 19%.

Aus dem zweiten Validierungsexperiment zeigt Abbildung 6.11 zwei Stimuli mit den verwendeten Operatoren (linke Spalte) sowie Heat-Maps der Fixationen aller Probanden (rechte Spalte). Die Probanden sollten im zweiten Validierungsexperiment folgende zwei Aufgaben durchführen: „Geben Sie den Wert der Kategorie X an.“ (Aufgabe D) und „Geben Sie den Namen und den Wert der höchsten Kategorie an.“ (Aufgabe E). Die Durchführungszeiten wurden im DVM mit den folgenden beiden Modellgleichungen vorhergesagt:

- $\mathcal{V}[\text{out of}]_{hws} + \mathcal{F} + \mathcal{A}_{vns} = 2.2s + 0.2s + 3.2s = 5.6s$ (Aufgabe D, Abbildung 6.11a+b)
- $\mathcal{V}[\text{out of}]_{hws} + \mathcal{V}[\text{out of}]_{hws}$
 $+ \mathcal{V}[\text{highest of } 2]_{ns} + \mathcal{A}_{vns} = 2.2s + 2.2s + 3.3s + 3.2s = 10.9s$
 (Aufgabe E, Abbildung 6.11c+d)

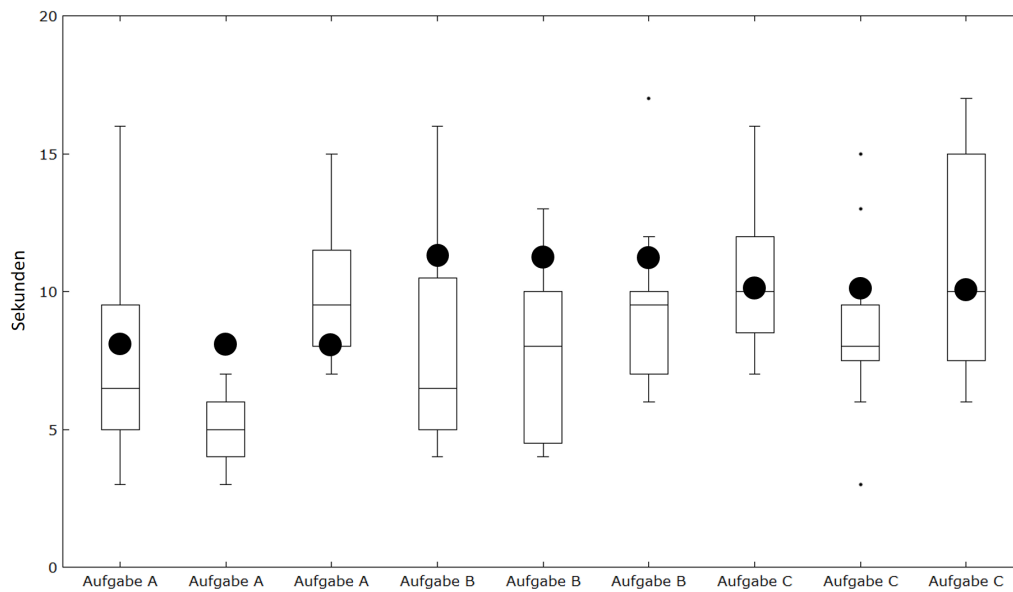


Abbildung 6.12: Vorhergesagte Zeiten durch das DVM (Kreise) mit Boxplots der im ersten Validierungsexperiment gemessenen Durchführungszeiten. Die durchschnittliche Differenz zwischen den vorhergesagten Zeiten und den gemessenen Zeiten beträgt 19%. Dabei wurde über alle drei Aufgaben A, B und C des ersten Validierungsexperiments der Mittelwert gebildet.

Alle Stimuli des zweiten Validierungsexperiments waren ebenfalls synthetisch erzeugt worden. Zur Darstellung der Stimuli wurde Excel 2013 verwendet. Die Diagramme zeigen fünf, zehn oder fünfzehn Datenpunkte. Für die Modellierung der Aufgaben wurden die Modelloperatoren $\mathcal{V}[\text{out of}]_{hws}$, $\mathcal{V}[\text{highest of}]_{ns}$, \mathcal{A}_{vns} und \mathcal{F} verwendet. Im Gegensatz zum ersten Validierungsexperiment wurden die Kategorien in den Diagrammen in einer geordneten Weise benannt. So wurden Jahre und Monate chronologisch aufsteigend in den Diagrammen beschriftet.

Abbildung 6.13 zeigt die durch das Modell vorhergesagten Zeiten (Kreise) zusammen mit Boxplots der gemessenen Zeiten. Die durchschnittliche Differenz zwischen den vorhergesagten Zeiten und den tatsächlich gemessenen Zeiten beträgt 43% für Aufgabe D bzw. 58% für Aufgabe E.

6.2.5 Diskussion

Die Ergebnisse aus den Validierungsexperimenten zeigen mit prozentualen Fehlerabweichungen zwischen 19% und 58%, dass das DVM deutliche Einschränkungen für eine sinnvolle Anwendung hat.

Das erste Validierungsexperiment zeigt, dass das DVM Durchführungs-

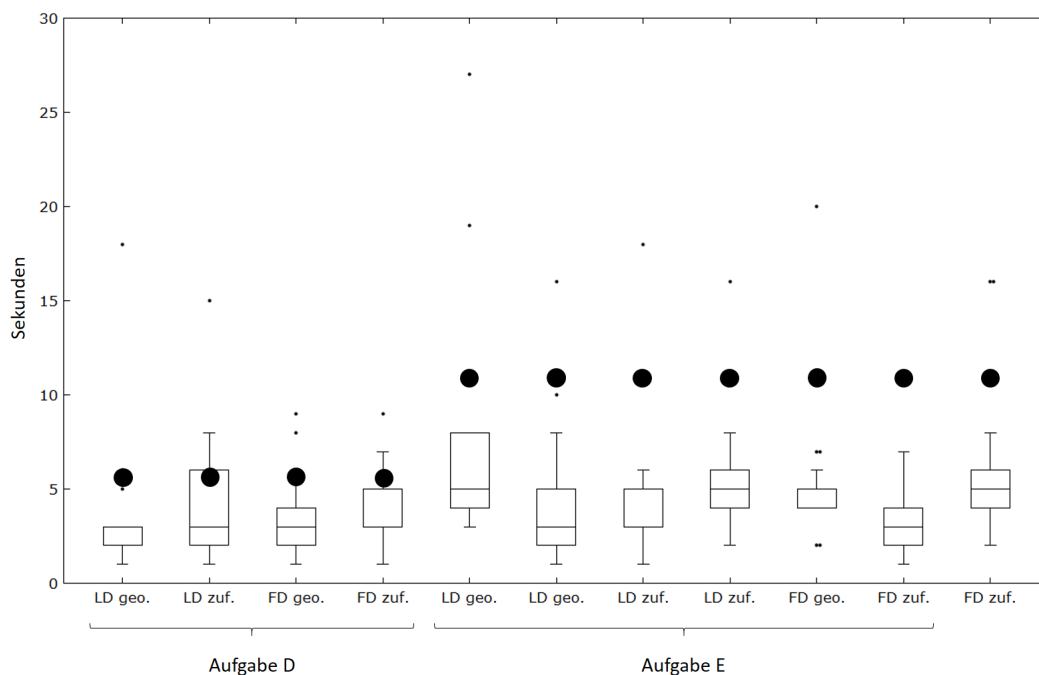


Abbildung 6.13: Vorhergesagte Zeiten durch das DVM (Kreise) mit Boxplots der im zweiten Validierungsexperiment gemessenen Durchführungszeiten. Die durchschnittliche Differenz zwischen den vorhergesagten Zeiten und den gemessenen Zeiten beträgt 43% für Aufgabe D bzw. 58% für Aufgabe E. Dazu wurde über alle gemessenen Durchführungszeiten der Mittelwert gebildet.

zeiten vorhersagt, die in acht von neun Fällen höher als der Median der gemessenen Zeiten sind (Abbildung 6.12). Nur vier der vorhergesagten Zeiten liegen innerhalb der Quartile. Über alle neun Messreihen ergibt sich eine prozentuale Abweichung zwischen Vorhersage und Messung von 19%. Die Heat-Maps zeigen, dass die visuelle Suchstrategie durch die gewählten Operatoren abgedeckt wird. Im Gegensatz dazu zeigt das zweite Validierungsexperiment jedoch eine größere Differenz zwischen Vorhersage und empirisch ermittelten Durchführungszeiten von 43% bzw. 58% (Abbildung 6.12). Im zweiten Fall sagt das DVM ebenfalls wieder Durchführungszeiten voraus, die höher als alle Mediane der tatsächlich gemessenen Zeiten sind. Ebenfalls zeigt die Analyse der Heat-Maps, dass die Modellierung der visuellen Suchstrategien durch die gewählten Operatoren eigentlich zu einer guten Vorhersagequalität führen müsste.

Betrachtet man die prozentuale Fehlerabweichung in Abhängigkeit der gewählten Visualisierungstechniken Säulen-, Linien- und Flächendiagramm, so erkennt man, dass das DVM nur für Säulendiagramme visuelle Aufgaben gut vorhersagt (19% Fehlerabweichung). Für Liniendiagramme

ergibt das DVM eine Fehlerabweichung von 43% und für Flächendiagramme 58%. Eine mögliche Erklärung für diese Ergebnisse könnte sein, dass Säulendiagramme in ihrem graphischen Aussehen am stärksten den für die Parameterdefinition der Operatoren verwendeten Scatter-Plots entsprechen. Sowohl Linien- als auch Flächendiagramme verbinden die einzelnen Datenpunkte mit Linien. Diese Linien führen zu anderen perzeptiven Wahrnehmungsprozessen und visuellen Suchestrategien. Deshalb müssen sie im DVM durch andere Operatoren abgebildet werden.

Zusätzlich erkennt man in den Ergebnissen, dass die Vorhersagequalität ebenfalls abnimmt, wenn ein gleicher Operator mehrmals hintereinander ausgeführt wird. Dies ist zum Beispiel für Aufgaben mit dem Operator \forall [out of] der Fall. Wenn die Probanden mehrmals nach bestimmten Datenpunkten in den Visualisierungen suchen sollten, können Lerneffekte eine zunehmende Rolle spielen. Diese werden in der vorliegenden Version des DVMs nicht berücksichtigt und führen aus diesem Grund zu einer schlechteren Qualität der Vorhersage. Aus dem gleichen Grund ist die prozentuale Fehlerabweichung ebenfalls höher, wenn die in den Visualisierungen dargestellten Kategorien alphabetisch oder chronologisch aufsteigend sortiert und beschriftet sind. Eine mögliche Lösung könnte darin bestehen, dass wenn mehrere Operatoren gleichen Typs hintereinander ausgeführt werden, die Durchführungszeiten für die auf den ersten Operator nachfolgenden Operatoren reduziert werden. Eine wichtige Frage dabei ist, wie stark die Durchführungszeiten für nachfolgende Operatoren reduziert werden müssen, um eine gute Vorhersagequalität zu erreichen? Diese Frage könnte in Experimenten geklärt werden, in denen kontrolliert die Anzahl von gleichen durchzuführen Visual Unit Tasks, die hintereinander durchgeführt werden müssen, erhöht wird.

Um den Ansatz des KLMs auf die Modellierung der visuellen Suche übertragen zu können, wurde das Konzept des idealen Scan-Paths (Abschnitt 2.5.1) verwendet. Allerdings setzt die Überführung einer visuellen Suchstrategie in eine Modellgleichung, mit Hilfe eines idealen Scan-Paths, einen linearen Leseprozess voraus. Je komplexer die gegebene Aufgabe oder je umfangreicher die Leseregeln sind, mit denen eine bestimmte Information aus einer Visualisierung gelesen werden muss, umso weniger entsprechen die tatsächlich durchgeführten Scan-Paths einem linearen Leseprozess. Dazu wird zum einen die Vorhersage der Reihenfolge von Operatoren in den Modellgleichungen schwierig, zum anderen wird die Gesamtanzahl der verwendeten Operatoren groß. Dieses Problem könnte wie folgt gelöst werden: Zum einen könnten alle Durchführungszeiten für alle möglichen visuellen Suchstrategien im Modell abgebildet und ein Durchschnittswert berechnet werden. Dann könnten mit Hilfe der Analyse von Eye-Tracking-Experimenten eine Häufigkeitsverteilung für diese einzelnen

visuellen Suchstrategien berechnet werden. Dazu können die in Kapitel 4 und 5 dieser Arbeit vorgestellten Methoden und Techniken eingesetzt werden. Basierend auf dieser Häufigkeitsverteilung könnte das DVM für bestimmte Nutzergruppen von Visualisierungen die Durchführungszeiten der visuellen Suche vorhersagen.

6.3 Vergleich der beiden Methoden

cSim bietet die Möglichkeit, die visuelle Suche mit Hilfe der Simulation von kognitiven Prozessen sehr detailliert vorherzusagen. Dazu müssen allerdings umfangreiche ACT-R-Programme entwickelt werden. Um den zeitlichen Aufwand dafür zu reduzieren, wurde das in Abschnitt vorgestellte Framework entwickelt. Sobald eine ACT-R-Programmvorlage entwickelt ist, kann diese für eine Vielzahl von verschiedenen Stimuli für eine gleiche Aufgabe eingesetzt werden. Ein operatorenbasierter Ansatz besteht dagegen durch seine Einfachheit. Für die Modellierung einer visuellen Aufgabe müssen lediglich passende Operatoren aus dem DVM miteinander kombiniert werden. Jedoch besitzt diese Methode, wie in Abschnitt 6.2.5 diskutiert, viele Einschränkungen. Wichtig bei der Anwendung des DVMs ist die Verwendung von möglichst gut passenden Operatoren. Diese müssen in aufwändigen Benutzerexperimenten bestimmt werden.

Als ideale Lösung könnte eine Kombination aus beiden Methoden dienen. Die visuelle Suche selbst wird durch eine Simulation berechnet, die allerdings auch auf empirisch ermittelten Zeiten für einzelne atomare Aufgaben basiert. Dieses Verfahren wird beispielsweise zur Simulation in der Mensch-Computer-Interaktion in der Software CogTool (John et al., 2004) angewendet.

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Das Hauptaugenmerk bei der Entwicklung von Visualisierungstechniken lag bisher meist auf der Frage, wie möglichst viele Daten in möglichst kurzer Zeit, in einer möglichst hohen Auflösung dargestellt werden können. In den letzten Jahren gewann in der Visualisierungsforschung jedoch auch die Fragestellung an Bedeutung, ob eine Visualisierung auch nützlich und leicht lesbar ist. Um diese Frage umfangreich für eine Visualisierungstechnik beantworten zu können, müssen Benutzerexperimente durchgeführt werden. In kontrollierten Laborexperimenten werden dazu den Probanden Aufgaben gestellt, die sie mit Hilfe einer oder mehrerer Visualisierungstechniken beantworten müssen. Neben der Analyse der Durchführungszeiten und der Fehlerraten, erlaubt das Eye-Tracking auch die Augenbewegungen der Probanden aufzuzeichnen, um die Lösungsstrategie der Probanden im Detail untersuchen zu können. Die dabei aufgenommenen Scan-Paths erlauben auch Rückschlüsse auf kognitive Prozesse während der Durchführung der visuellen Aufgaben.

Wie der Stand der Technik zeigt, existieren eine Vielzahl von Visualisierungstechniken zur Analyse von Eye-Tracking-Daten sowie verschiedene Ansätze zur Modellierung der perzeptiven und kognitiven Prozesse. Jedoch fehlt bisher ein Framework, mit dem Simulationen zur Vorhersage von visuellen Suchstrategien entwickelt und mit Ergebnissen aus Eye-Tracking-Experimenten verglichen werden können.

Kapitel 3 stellt für die Entwicklung eines solchen Frameworks einen interdisziplinären Ansatz vor, mit dem Modelle und Techniken aus den drei Forschungsgebieten des Eye-Trackings, der Wissensrepräsentation und der Kognitionswissenschaften miteinander kombiniert werden. Auf Basis dieser Kombination wurden im Rahmen dieser Arbeit neue Methoden und Techniken zur Evaluation von Visualisierungen entwickelt. Diese neuen Methoden und Techniken umfassen die parallele Scan-Path-Visualisierungstechnik für eine effiziente Identifikation von Lösungsstrategien und den Vergleich von

Scan-Paths. Mit Hilfe der Annotation von graphischen Elementen in Visualisierungen mit semantischer Information erlaubt diese Visualisierungstechnik die Formulierung von Lesestrategien, die dann für die Entwicklung von Kognitionssimulationen verwendet werden. Die Ergebnisse aus diesen Simulationen können dann visuell wieder mit Ergebnissen aus realen Eye-Tracking-Experimenten verglichen werden, um die Simulationsmodelle weiter anzupassen.

In Kapitel 4 wird die parallele Scan-Path-Visualisierungstechnik vorgestellt. Mit dieser kann der Aufwand für die Auswertung von Eye-Tracking-Experimenten reduziert werden und es wird möglich, dass ähnliche Augenbewegungsmuster über viele Probanden hinweg identifiziert werden können. Diese Visualisierungstechnik erlaubt es zusätzlich, dass Lesestrategien von Visualisierungen über mehrere Probanden hinweg erkannt und miteinander verglichen werden können. Dazu wurde ein Ansatz der visuellen Analytik verwendet, in dem die PSP-Visualisierungstechnik als Visualisierungskomponente mit einem Hierarchical Agglomerative Clustering erweitert wurde. Als Abstandsmaß wurde die Levenshtein-Distanz verwendet. Es wurde gezeigt, wie die parallele Scan-Path-Visualisierungstechnik in verschiedenen Anwendungsfällen für die Analyse von Augenbewegungen bei der Durchführung von einfachen visuellen Aufgaben und bei der Bearbeitung von komplexeren Aufgaben mit Balkendiagrammen eingesetzt wird. Im Falle der Balkendiagramme wurde zusätzlich demonstriert, wie das Hierarchical Agglomerative Clustering eingesetzt wird, um mit kleinem Aufwand Gruppen von Probanden zu identifizieren, die eine ähnliche Strategie zur Lösung der gestellten Aufgabe verwendet haben.

Kapitel 5 stellt zunächst auf Basis von existierenden Arbeiten ein Wissensmodell über Visualisierungen und die darin enthaltenen, graphischen Elementen auf drei Ebenen vor. Jede Ebene beschreibt in Form einer Ontologie eine unterschiedliche Abstraktionsebene des Wissens über Visualisierungen und die darin enthaltenen, graphischen Elemente. Dieses Wissensmodell wird dazu verwendet, um graphische Elemente in Visualisierungen mit semantischen Informationen zu annotieren. Damit wird es möglich, nicht nur wie bisher zu analysieren, welche Bereiche in einer Visualisierung in welcher Reihenfolge betrachtet worden sind (WO-Raum), sondern auch, was für graphische Elemente dort wahrgenommen (WAS-Raum) und wie diese kognitiv weiter verarbeitet wurden. Da jedoch Benutzerexperimente meist eine große Anzahl von Stimuli besitzen und die Annotation aller aufgabenrelevanten graphischen Elemente in diesen Stimuli viel Zeit benötigt, wurde ein Verfahren zur automatischen Annotation von Visualisierungen mit semantischer Information entwickelt. Drei Anwendungsfälle demonstrieren, wie die Kombination der Annotation von graphischen Elementen mit der parallelen Scan-Path-Visualisierungstechnik dazu verwendet wer-

den kann, unterschiedliche kognitive Prozesse bei der Durchführung einer visuellen Aufgaben zu untersuchen. Zusätzlich wird gezeigt, wie mit der vorgestellten Methode Eigenschaften der Visualisierung identifiziert werden können, die zu einer kognitiven Verzerrung führen können. Für die Entwicklung von Simulationen der visuellen Suche wird gezeigt, wie mit Hilfe der semantischen Annotation Lesestrategien für visuelle Aufgaben formuliert werden können. Das fünfte Kapitel schließt mit einer Diskussion der Einschränkungen des vorgestellten Ansatzes. Insbesondere wird die dieser Arbeit zugrundeliegende Augen-Gedanken-Hypothese diskutiert, die besagt, dass davon ausgegangen werden kann, dass ein Betrachter die in einem Bereich einer Visualisierung dargestellte Information kognitiv verarbeitet, solange er diesen Bereich fokussiert.

Kapitel 6 stellt, aufbauend auf die vorhergehenden Kapitel zwei Methoden zur Simulation und Modellierung der visuellen Suche vor. Wie im Stand der Technik beschrieben ist, werden Benutzerexperimente meist inkrementell entworfen und durchgeführt. Dieser inkrementelle Vorgang ist sehr zeitaufwändig. Im Gegensatz zur Durchführung von Benutzerexperimenten für die Evaluation von Visualisierungstechniken existiert nur eine kleine Anzahl von Arbeiten zur Modellierung von visuellen Suchstrategien. Um den Aufwand für die Entwicklung von Simulationen der visuellen Suche zu reduzieren, werden zwei Methoden vorgestellt. Die erste Methode besteht in der Simulationsumgebung cSim, mit der Augenbewegungen während der visuellen Suche mit Hilfe des Kognitionssimulationsframeworks ACT-R simuliert werden. Da ACT-R-Programme in Lisp geschrieben sind, einen hohen Grad an Komplexität aufweisen und für jede neue Aufgabe oder Stimulus neu entwickelt oder angepasst werden müssen, erlaubt es cSim mit Hilfe einer Kombination aus dem Wissensmodell und der Identifikation von Lesestrategien, ACT-R-Programme automatisch generieren zu lassen. Sowohl die Generierung, Ausführung als auch Visualisierung der berechneten Scan-Paths findet in der gleichen Simulationsumgebung statt. Die zweite Methode besteht in einem operatorenbasierten Diagram-Viewing-Modell, das ähnlich dem aus der Mensch-Computer-Interaktionsforschung bekannten Keystroke-Level-Modell Durchführungszeiten von visuellen Aufgaben vorhersagt. Mit Hilfe eines Satzes von primitiven Operatoren und Heuristiken sagt das Modell Durchführungszeit für eine visuelle Aufgabe voraus. Jedoch zeigte eine umfangreiche Validierung dieser Methode, dass sie deutliche Einschränkungen besitzt. So zeigte es sich beispielsweise, dass die Verwendung von möglichst gut passenden Operatoren eine sehr wichtige Rolle bei der Anwendung des Diagram-Viewing-Modells spielt. Die Definition von passenden Operatoren benötigt jedoch aufwändige Benutzerexperimente, die vor der Modellierung durchgeführt werden müssen. Das sechste Kapitel schließt mit einer umfangreichen Diskussion der Ein-

schränkungen des Diagram-Viewing-Modells sowie einem Vergleich der beiden in diesem Kapitel vorgestellten Methoden.

Zusammengefasst unterstützen die in dieser Arbeit vorgestellten neuen Methoden und Techniken die Evaluation von Visualisierungen und können zu einer Effizienzsteigerung bei der Auswertung von Eye-Tracking-Experimenten führen. Die vorgestellten Methoden und Techniken wurden dazu in Eye-Tracking-Experimenten mit Informationsvisualisierungen evaluiert. Durch die Kombination der visuellen Analyse von Scan-Paths mit ontologiebasierten Wissensmodellen können die semantischen Charakteristika von Scan-Paths untersucht werden. Damit lassen sich Visualisierungen besser benutzerorientiert entwickeln. Darüber hinaus zeigt die Analyse der semantischen Eigenschaften von Scan-Paths Schwachstellen im Visualisierungskonzept. Modelle zur Vorhersage von Durchführungszeiten, sowie der Einsatz von Kognitionssimulationen zur Berechnung von Scan-Paths reduzieren den Aufwand, der nötig ist, um visuelle Suchstrategien zu simulieren.

7.2 Ausblick

Wie im Abschnitt zum Stand der Technik zur Visualisierung von Eye-Tracking-Daten gezeigt, bestehen Lücken in der visuellen Analyse von Eye-Tracking-Daten in den Domänen der interaktiven und dynamischen Stimuli. Ausgehend von der parallelen Scan-Path-Visualisierungstechnik und der erfolgreichen Kombination dieser Visualisierungstechnik mit Mustererkennungsverfahren können weitere Visualisierungstechniken für interaktive und dynamische Stimuli entwickelt werden. So bietet es sich an, weitere Algorithmen aus dem Gebiet des String-Editings für die Distanzberechnung zu verwenden. Ebenso ist es denkbar, die Ähnlichkeit von Scan-Paths auch mit Methoden der Künstlichen Intelligenz wie beispielsweise Neuronalen Netzen zu berechnen.

Insgesamt eröffnet sich mit Hilfe der in dieser Arbeit vorgestellten Methoden und Techniken die Möglichkeit, Visualisierungstechniken iterativ zu entwickeln. Dieser neuartige Design-Prozess, der eine mehrmals durchgeführte Evaluation benötigt, kann in einer auf Tory und Möller basierenden Form (Tory und Möller, 2004) mit Hilfe von vier Schritten beschrieben werden (Abbildung 7.1). Zunächst wird die Visualisierung entworfen. Anschließend wird ein Eye-Tracking-Experiment vorbereitet, um die neue Visualisierungstechnik oder ein bestimmtes Layout dieser Visualisierungstechnik zu evaluieren. Möglicherweise wird auch eine Pilotstudie durchgeführt, mit der die Parameter des Hauptexperiments getestet und gegebenenfalls verändert werden. Der Hauptschritt besteht in der

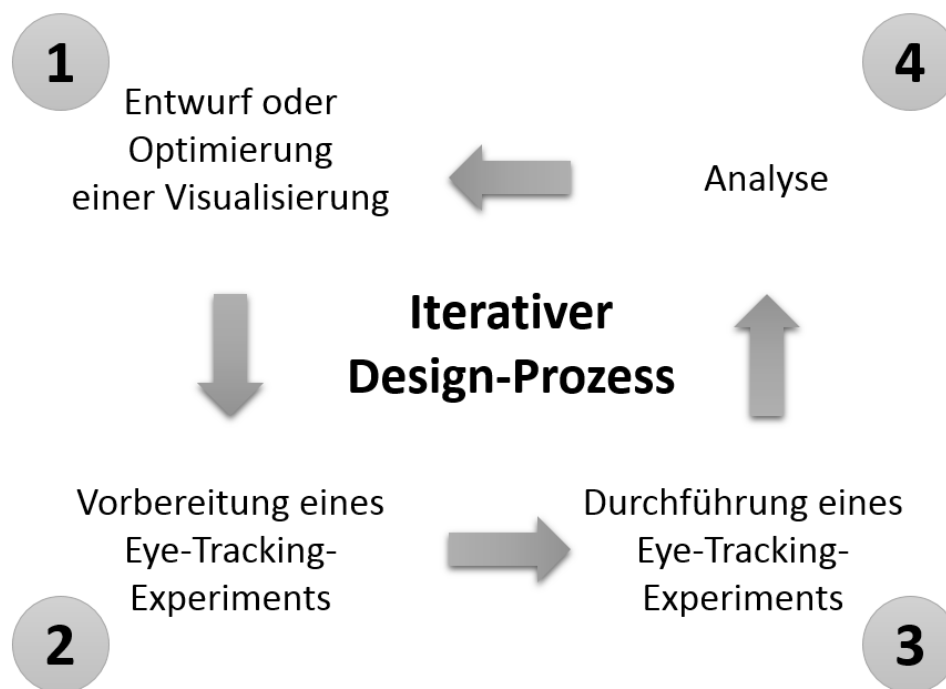


Abbildung 7.1: Iterativen Entwurfprozess einer Visualisierung in vier Schritten (basierend auf Tory und Möller (Tory und Möller, 2004)).

eigentlichen Durchführung des Eye-Tracking-Experiments und seiner anschließenden Auswertung. Auf Basis der Ergebnisse wird anschließend die Visualisierung weiterentwickelt. Dann startet der iterative Entwurfprozess wieder von vorne. Falls für die Datenanalyse AOIs eingesetzt werden sollen, kann der Zeitaufwand für den zweiten und vierten Schritt deutlich gesenkt werden, wenn AOIs automatisch erstellt werden können. Durch die teilautomatisierte Analyse der Augenbewegungen können gleiche Musterrerkennungsverfahren in jedem Iterationsschritt angewendet werden. Wird dieses Verfahren weiter optimiert, ist eine Analyse der Eye-Tracking-Daten mit sehr wenigen Mausclicks denkbar.

Durch die Kombination der visuellen Analyse von Scan-Paths mit ontologiebasierten Wissensmodellen ergeben sich auch neue Möglichkeiten in Hinblick auf die Analyse von Gedächtnisprozessen und der Entwicklung von Gedächtnismodellen. Beispielsweise lässt sich in der PSP-Visualisierung schon jetzt die Zeit ablesen, die benötigt wird, um eine bestimmte Information aus dem Gedächtnis abzurufen oder dort abzuspeichern. Weiter in die Zukunft gedacht, könnte die Kombination aus Eye-Tracking und ontologiebasierten Wissensmodellen dazu verwendet werden, um das Modell der *Spreading Activation* (Anderson, 1983a) weiterzuentwickeln. Dazu könnten auch weitere Informationskanäle aus Benutzerexperimenten wie

Gesprächsprotokolle (Thinking-Aloud) oder Information über die Interaktion mit einer Visualisierungssoftware verwendet werden. Werden all diese Informationen zusammengeführt und existieren detaillierte Modelle der kognitiven und perceptiven Prozesse, so könnten Kognitionssimulationen sehr zuverlässig die visuelle Suche vorhersagen. Damit wäre die Durchführung von Benutzerexperimenten nicht mehr notwendig.

Vielleicht einer der faszinieresten Möglichkeiten der vorliegenden Arbeit besteht darin, dass es mit den entwickelten Techniken prinzipiell möglich ist, mit einem günstigen Verfahren die Lücke zwischen psychologischen und kognitiven Modellen der menschlichen Informationsverarbeitung auf der einen Seite und Informationen von Aufnahmen aus Magnetresonanztomographen zu schließen. Mit geeigneten Methoden der visuellen Analytik können Wahrnehmungsprozesse im Scan-Path wie Prozesse der perceptiven Wahrnehmung, der Augensteuerung, Gedächtnis- und Verstehensprozesse erkannt werden. All diesen Prozessen können prinzipiell Gehirnareale zugeordnet werden. Somit besteht die Möglichkeit, aus Eye-Tracking-Daten relativ günstig die Aktivität von bestimmten Hirnarealen zu berechnen, die anschließend mit Informationen von Elektroenzephalografiegeräten und Magnetresonanztomographen verglichen werden können.

Werden weitere Sensordaten mit in die Analyse der Eye-Tracking-Daten eingebunden, ergibt sich ein immer umfassenderes Bild über den psychische und physischen Zustand des Probanden. Wird daraus der Verlauf des emotionalen Zustands ermittelt und mit Informationen aus dem WAS-Raum verknüpft, so könnte erkannt werden, welche Art von graphischen Elementen oder welche Art von visueller Suchstrategie oder visueller Aufgabe zu welchem Gefühlszustand führt. Steht erstmal ein so umfangreiches Bild des Probanden während der Interaktion mit einer Visualisierung zur Verfügung, könnten Assistenten eine Visualisierung optimal an die mentalen Fähigkeiten und den aktuellen psychischen, physischen und kognitiven Zustand des Betrachters anpassen. Diese Assistenten könnten sogar durch eine Simulation der möglichen visuellen Suchstrategien „vorausdenken“ und erkennen, welche Informationen der Benutzer als nächstes in der Visualisierung verstehen möchte. Hier reicht die Anwendung von adaptiven Visualisierungen bis hin zu prädiktiven Analysesystemen für die visuelle Analytik. Noch weiter in die Zukunft gedacht, könnten damit Computer in die Lage versetzt werden, das menschliche Denken immer besser nachzubilden und damit eine bessere Mensch-Computer-Interaktion zu erlauben.

Literaturverzeichnis

- Anderson, J. R. (1976), *Language, Memory and Thought*. (Hillsdale, NJ: Erlbaum). (Zitiert auf Seite 42.)
- Anderson, J. R. (1983a), „A spreading activation theory of memory“, *Journal of Verbal Learning and Verbal Behavior*, 22/3, 261–295. (Zitiert auf Seite 44, 113, and 151.)
- Anderson, J. R. (1983b), *The Architecture of Cognition* (Cambridge, MA, USA: Harvard University Press). (Zitiert auf Seite 42.)
- Anderson, J. R. (2001), *Kognitive Psychologie* (Spektrum Akademischer Verlag), 3 edition. (Zitiert auf Seite 30.)
- Anderson, J. R. (2007), *How can the human mind occur in the physical universe* (New York: Oxford University Press). (Zitiert auf Seite 42.)
- Anderson, J. R. und Lebiere, C. (1998), *The atomic components of Thought* (NJ: Mahwah Hillsdale). (Zitiert auf Seite 42.)
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., und Qin, Y. (2004), „An integrated theory of the mind.“, *Psychological Review*, 111/4, 1036–1060. (Zitiert auf Seite 42 and 43.)
- Andrienko, G., Andrienko, N., Burch, M., und Weiskopf, D. (2012), „Visual Analytics Methodology for Eye Movement Studies“, *IEEE Transactions on Visualization and Computer Graphics*, 18/12, 2889–2898. (Zitiert auf Seite 26 and 49.)
- Azevedo, F. A. C., Carvalho, L. R. B., Grinberg, L. T., Farfel, J. M., Ferretti, R. E. L., Leite, R. E. P., Filho, W. J., Lent, R., und Herculano-Houzel, S. (2009), „Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain“, *The Journal of Comparative Neurology*, 513/5, 532–541. (Zitiert auf Seite 31.)
- Baddeley, A. (1981), „The concept of Working memory“, *Cognition*, 10, 17–23. (Zitiert auf Seite 35.)
- Bälter, O. (2000), „Keystroke level analysis of email message organization“, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00 (New York, NY, USA: ACM), 105–112. (Zitiert auf Seite 40.)

- Bednarik, R. und Tukiainen, M. (2008), „Temporal Eye-tracking Data: Evolution of Debugging Strategies with Multiple Representations“, in *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, ETRA '08 (New York, NY, USA: ACM), 99–102. (Zitiert auf Seite 25 and 49.)
- Bellamy, R., John, B., Richards, J., und Thomas, J. (2010), „Using CogTool to Model Programming Tasks“, in *Evaluation and Usability of Programming Languages and Tools*, PLATEAU '10 (New York, NY, USA: ACM), 1:1–1:6. (Zitiert auf Seite 45.)
- Beymer, D. und Russell, D. M. (2005), „WebGazeAnalyzer: A System for Capturing and Analyzing Web Reading Behavior Using Eye Gaze“, in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI Extended Abstracts '05 (New York, NY, USA: ACM), 1913–1916. (Zitiert auf Seite 25.)
- Blascheck, T. und Ertl, T. (2014), „Towards Analyzing Eye Tracking Data for Evaluating Interactive Visualization Systems“, in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (New York, NY, USA: ACM), 70–77. (Zitiert auf Seite 112.)
- Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., und Ertl, T. (2014), „State-of-the-Art of Visualization for Eye Tracking Data“, *EuroVis STAR*. (Zitiert auf Seite 21.)
- Blascheck, T., Burch, M., Raschke, M., und Weiskopf, D. (2015a), „Challenges and Perspectives in Big Eye-Movement Data Visual Analytics“, in *Proceedings of the 2015 IEEE International Symposium on Big Data Visual Analytics*, BDVA '15 (Hobart, Tasmania, Australia: IEEE). (Zitiert auf Seite 8.)
- Blascheck, T., John, M., Koch, S., Kurzhals, K., und Ertl, T. (2015b), „VA²: A Visual Analytics Approach for Evaluating Visual Analytics Applications“, *IEEE Transactions on Visualization and Computer Graphics*, 22/1. (Zitiert auf Seite 26.)
- Blignaut, P. (2010), „Visual Span and Other Parameters for the Generation of Heatmaps“, in *Proceedings of the 2010 Symposium on Eye Tracking Research and Applications*, ETRA '10 (ACM), 125–128. (Zitiert auf Seite 23.)
- Bojko, A. (2009), „Informative or Misleading? Heatmaps Deconstructed“, in Jacko, J., Herausgeber, *Human-Computer Interaction. New Trends*, 5610, Lecture Notes in Computer Science (Springer Berlin Heidelberg), 30–39. (Zitiert auf Seite 23.)

- Burch, M., Konevtsova, N., Heinrich, J., Höferlin, M., und Weiskopf, D. (2011), „Evaluation of Traditional, Orthogonal, and Radial Tree Diagrams by an Eye Tracking Study“, *IEEE Transactions on Visualization and Computer Graphics*, 17/12, 2440–2448. (Zitiert auf Seite 14 and 131.)
- Burch, M., Beck, F., und Weiskopf, D. (2012), „Radial Edge Splatting for Visualizing Dynamic Directed Graphs“, in *Proceedings of the International Conference on Information Visualization Theory and Applications (IVAAP)*, 603–612. (Zitiert auf Seite 14.)
- Burch, M., Kull, A., und Weiskopf, D. (2013), „AOI Rivers for Visualizing Dynamic Eye Gaze Frequencies“, *Computer Graphics Forum*, 32/3, 281–290. (Zitiert auf Seite 25.)
- Burch, M., Kurzhals, K., und Weiskopf, D. (2014a), „Visual Task Solution Strategies in Public Transport Maps“, in *Proceedings of the 2nd International Workshop on Eye Tracking for Spatial Research (Vienna)*. (Zitiert auf Seite 14.)
- Burch, M., Raschke, M., Blascheck, T., Kurzhals, K., und Weiskopf, D. (2014b), „How Do People Read Metro Maps? An Eye Tracking Study“, in *Proceedings of 1st International Workshop on Schematic Mapping (Essex, England)*. (Zitiert auf Seite 14.)
- Burch, M., Schmauder, H., Raschke, M., und Weiskopf, D. (2014c), „Saccade Plots“, in *Proceedings of the 2014 Symposium on Eye Tracking Research & Applications, ETRA '14 (New York, NY, USA: ACM)*, 307–310. (Zitiert auf Seite 24.)
- Card, S. K., Moran, T. P., und Newell, A. (1980), „The keystroke-level model for user performance time with interactive systems“, *Commun. ACM*, 23 /7, 396–410. (Zitiert auf Seite 38.)
- Card, S. K., Mackinlay, J. D., und Shneiderman, B., Herausgeber (1999), *Readings in information visualization: using vision to think* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.). (Zitiert auf Seite 5.)
- Card, S. K., Newell, A., und Moran, T. P. (2000), *The Psychology of Human-Computer Interaction* (Hillsdale, NJ, USA: L. Erlbaum Associates Inc.). (Zitiert auf Seite 38 and 40.)
- Chang, Y.-H. und Chuang, T.-R. (2003), „Embedding domain semantics in SVG“, in *SVG Open 2003: 2nd Annual Conference on Scalable Vector Graphics (Vancouver, Canada: SVG Open 2003 Conference and Exhibition)*. (Zitiert auf Seite 37.)

- Chen, M., Alves, N., und Sol, R. (2013), „Combining spatial and temporal Information of Eye movement in Goal-Oriented Tasks“, in *Human Factors in Computing and Informatics*, 7946 (Springer), 827–830. (Zitiert auf Seite 24.)
- Cleveland, W. S. und McGill, R. (1984), „Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods“, *Journal of the American Statistical Association*, 79/387, pp. 531–554. (Zitiert auf Seite 41.)
- Collins, A. M. und Quillian, M. R. (1969), „Retrieval time from Semantic Memory“, *Journal of Verbal Learning and Verbal Behavior*, 8, 240–247. (Zitiert auf Seite 35 and 36.)
- Conversy, S., Hurter, C., und Chatty, S. (2010), „A descriptive model of visual scanning“, in *Proceedings of the 3rd BELIV'10 Workshop: BEyond time and errors: novel evaluation methods for Information Visualization*, BELIV '10 (New York, NY, USA: ACM), 35–42. (Zitiert auf Seite 27.)
- Conversy, S., Chatty, S., und Hurter, C. (2011), „Visual scanning as a reference framework for interactive representation design“, *Information Visualization*, 10/3, 196–211. (Zitiert auf Seite 27.)
- Cosgrove, K. P., Mazure, C. M., und Staley, J. K. (2007), „Evolving knowledge of sex differences in brain structure, function, and chemistry“, *Biological Psychiatry*, 62/8, 847–855. (Zitiert auf Seite 31.)
- Cowan, N. (2001), „The magical number 4 in short-term memory“, *Behavioral and Brain Sciences*, 24/1, 87–114. (Zitiert auf Seite 35.)
- Crane, H. D. (1994), „The Purkinje Image Eyetracker, Image Stabilization, and Related Forms of Stimulus Manipulation“, in Kelly, D. H., Herausgeber, *Visual Science and Engineering: Models and Applications* (New York: Marcel Dekker), 15–89. (Zitiert auf Seite 13.)
- Crane, H. D. und Steele, C. M. (1985), „Generation-V dual Purkinje-image eyetracker“, *Applied Optics*, 24/4, 527–537. (Zitiert auf Seite 13.)
- Croft, J., Pittman, D., und Scialfa, C. (2006), „Gaze Behavior of Spotters During an Air-to-ground search“, in *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, ETRA '06 (New York, NY, USA: ACM), 163–179. (Zitiert auf Seite 24.)
- Crowe, E. C. und Narayanan, N. H. (2000), „Comparing interfaces based on what users watch and do“, in *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00 (New York, NY, USA: ACM), 29–36. (Zitiert auf Seite 25.)

- Dorr, M., Jarodzka, H., und Barth, E. (2010a), „Space-variant Spatio-temporal Filtering of Video for Gaze Visualization and Perceptual Learning“, in *Proceedings of the 2010 Symposium on Eye Tracking Research & Applications*, ETRA '10 (New York, NY, USA: ACM), 307–314. (Zitiert auf Seite 23.)
- Dorr, M., Martinetz, T., Gegenfurtner, K. R., und Barth, E. (2010b), „Variability of eye movement when viewing dynamic natural scenes“, *Journal of Vision*, 10/10, 1–17. (Zitiert auf Seite 23.)
- Drury, C. G. und Clement, M. R. (1978), „The effect of area, density, and number of background characters on visual search“, *Human Factors*, 20 /5, 597–602. (Zitiert auf Seite 10.)
- Duchowski, A. (2007), *Eye Tracking Methodology: Theory and Practice*, Science+Business Media (Springer), 2nd edition. (Zitiert auf Seite 13.)
- Duchowski, A. und McCormick, B. (1998), „Gaze-contingent Video Resolution Degradation“, in *Human Vision and Electronic Imaging III*, Proc. SPIE (), 318–329. (Zitiert auf Seite 24.)
- Duchowski, A. T. (2002), „A breadth-first survey of eye-tracking applications.“, *Behavior Research Methods, Instruments, and Computers*, 34/4, 455–470. (Zitiert auf Seite 13.)
- Duchowski, A. T., Driver, J., Jolaoso, S., Tan, W., Ramey, B. N., und Robbins, A. (2010), „Scanpath comparison revisited“, in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10 (New York, NY, USA: ACM), 219–226. (Zitiert auf Seite 27, 28, 70, and 93.)
- Duchowski, A. T., Price, M. M., Meyer, M., und Orero, P. (2012), „Aggregate Gaze Visualization with Real-time Heatmaps“, in *Proceedings of the 2012 Symposium on Eye Tracking Research and Applications*, ETRA '12 (ACM), 13–20. (Zitiert auf Seite 23.)
- Ebbinghaus, H. (1885), *Über das Gedächtnis* (Leipzig). (Zitiert auf Seite 35.)
- Etemadpour, R., Motta, R., de Souza Paiva, J. G., Minghim, R., de Oliveira, M. C. F., und Linsen, L. (2015), „Perception-Based Evaluation of Projection Methods for Multidimensional Data Visualization“, *IEEE Transactions on Visualization and Computer Graphics*, 21/1, 81–94. (Zitiert auf Seite 14.)
- Ferecatu, M., Boujemaa, N., und Crucianu, M. (2008), „Semantic interactive image retrieval combining visual and conceptual content description“, *Multimedia Systems*, 13/5, 309–322. (Zitiert auf Seite 37.)

- Field, A. und Hole, G. (2003), *How to Design and Report Experiments* (London: SAGE Publications Ltd), 1 edition. (Zitiert auf Seite 16 and 115.)
- Goldberg, J. H. und Helfman, J. (2011), „Eye tracking for visualization evaluation: Reading values on linear versus radial graphs“, *Information Visualization*, 10/3, 182–195. (Zitiert auf Seite 14.)
- Goldberg, J. H. und Helfman, J. I. (2010a), „Visual scanpath representation“, in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10 (New York, NY, USA: ACM), 203–210. (Zitiert auf Seite 21 and 24.)
- Goldberg, J. H. und Helfman, J. I. (2010b), „Scanpath clustering and aggregation“, in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10 (New York, NY, USA: ACM), 227–234. (Zitiert auf Seite 26.)
- Goldberg, J. H. und Kotval, X. P. (1999), „Computer interface evaluation using eye movements: methods and constructs“, *International Journal of Industrial Ergonomics*, 24/6, 631–645. (Zitiert auf Seite 19 and 26.)
- Goldstein, R., Woods, R., und Peli, E. (2007), „Where people look when watching movies: Do all viewers look at the same place?“, *Computers in Biology and Medicine*, 37/7, 957–964. (Zitiert auf Seite 23.)
- Gong, R. und Elkerton, J. (1990), „Designing minimal documentation using a GOMS model: a usability evaluation of an engineering approach“, in *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, CHI '90 (New York, NY, USA: ACM), 99–107. (Zitiert auf Seite 40.)
- Gong, R. und Kieras, D. (1994), „A validation of the GOMS model methodology in the development of a specialized, commercial software application“, in *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94 (New York, NY, USA: ACM), 351–357. (Zitiert auf Seite 40.)
- Görz, G., Schneeberger, J., und Schmid, U., Herausgeber (2014), *Handbuch der Künstlichen Intelligenz* (Oldenbourg Verlag München), 5 edition. (Zitiert auf Seite 30.)
- Grant, E. R. und Spivey, M. J. (2002), „Guiding Attention Produces Inferences in Diagram-Based Problem Solving“, in *Proceedings of the Second International Conference on Diagrammatic Representation and Inference*, 2317, *Diagrams 2002* (London, UK, UK: Springer-Verlag), 236–248. (Zitiert auf Seite 85 and 86.)

- Gray, W. D., John, B. E., und Atwood, M. E. (1992), „The precis of Project Ernestine or an overview of a validation of GOMS“, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92 (New York, NY, USA: ACM), 307–312. (Zitiert auf Seite 40.)
- Grindinger, T., Duchowski, A., und Sawyer, M. (2010), „Group-Wise Similarity and Classification of Aggregate Scanpaths“, in *Proceedings of the 2010 Symposium on Eye Tracking Research & Applications*, ETRA '10 (New York, NY, USA: ACM), 101–104. (Zitiert auf Seite 21.)
- Gruber, T. R. (1993), „A Translation Approach to Portable Ontology Specifications“, *Knowledge Acquisition*, 5/2, 199–220. (Zitiert auf Seite 37.)
- Guan, Z. und Cutrell, E. (2007), „An eye tracking study of the effect of target rank on web search“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07 (New York, NY, USA: ACM), 417–420. (Zitiert auf Seite 61.)
- Haber, R. B. und McNabb, D. A. (1990), „Visualization idioms: A conceptual model for scientific visualization systems“, in *Visualization in Scientific Computing*, 74–93. (Zitiert auf Seite 5.)
- Halverson, T. und Hornof, A. J. (2007), „A minimal model for predicting visual search in human-computer interaction“, in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, CHI '07 (New York, NY, USA: ACM), 431–434. (Zitiert auf Seite 131.)
- Halverson, T. und Hornof, A. J. (2011), „A Computational Model of Active Vision for Visual Search in Human-Computer Interaction“, *Human-Computer Interaction*, 26/4, 285–314. (Zitiert auf Seite 42.)
- Hart, S. G. und Staveland, L. E. (1988), „Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research“, *Advances in Psychology*, 52, 139–183. (Zitiert auf Seite 18.)
- Haunold, P. und Kuhn, W. (1993), „A Keystroke Level Analysis of Manual Map Digitizing“, in *Spatial Information Theory, European Conference, COSIT 93 . Lecture Notes in Computer Science 716* (Springer-Verlag), 406–420. (Zitiert auf Seite 40.)
- Hembrooke, H., Feusner, M., und Gay, G. (2006), „Averaging Scan Patterns and What they can tell us“, in *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, ETRA '06 (New York, NY, USA: ACM), 41. (Zitiert auf Seite 24.)

- Holleis, P., Kern, D., und Schmidt, A. (2007a), „Integrating user performance time models in the design of tangible UIs“, in *CHI '07 extended abstracts on Human factors in computing systems*, CHI EA '07 (New York, NY, USA: ACM), 2423–2428. (Zitiert auf Seite 41.)
- Holleis, P., Otto, F., Hussmann, H., und Schmidt, A. (2007b), „Keystroke-level model for advanced mobile phone interaction“, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07 (New York, NY, USA: ACM), 1505–1514. (Zitiert auf Seite 41.)
- Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., und Van de Weijer, J. (2011), *Eye Tracking: A Comprehensive Guide to Methods and Measures* (Oxford University Press), 1 edition. (Zitiert auf Seite 18 and 20.)
- Holsanova, J. (2001), *Picture viewing and picture description: Two windows on the mind*. (Lund: Lund University Cognitive Studies 38). (Zitiert auf Seite 25.)
- Holsanova, J., Holmberg, N., und Holmqvist, K. (2008), „Reading information on graphics: The role of spatial contiguity and dual attentional guidance“, *Applied Cognitive Psychology*, 23/9, 1215–1226. (Zitiert auf Seite 34.)
- Horn, B. K. P. und Schunck, B. G. (1981), „Determining Optical Flow“, *Artificial Intelligence*, 17/1-3, 185–203. (Zitiert auf Seite 23.)
- Huang, W. und Eades, P. (2005), „How people read graphs“, in *proceedings of the 2005 Asia-Pacific symposium on Information visualisation*, 45 (Darlinghurst, Australia), 51–58. (Zitiert auf Seite 14.)
- Huang, W., Eades, P., und Hong, S.-H. (2009), „A graph reading behavior: Geodesic-path tendency“, in *Proceedings of the IEEE PacificVIS 2013, PACIFICVIS '09* (Washington, DC, USA: IEEE Computer Society), 137–144. (Zitiert auf Seite 14.)
- Hurter, C., Ersoy, O., Fabrikant, S., Klein, T., und Telea, A. (2013), „Bundled Visualization of Dynamic Graph and Trail Data“, *IEEE Transactions on Visualization and Computer Graphics*, 20/8, 1141–1157. (Zitiert auf Seite 24.)
- Ishihara, S. (1917), *Tests for Colour Blindness* (Tokyo: Handaya). (Zitiert auf Seite 16.)
- Itoh, K., Tanaka, H., und Seki, M. (2000), „Eye-movement analysis of track monitoring patterns of night train operators: Effects of geographic knowledge and fatigue“, in *Proceedings of the Human Factors and Ergonomics*

- Society Annual Meeting*, 44 (Human Factors and Ergonomics Society), 360–363. (Zitiert auf Seite 25 and 49.)
- Jacob, R. J. K. und Karn, K. S. (2003), „Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises“, *The Mind’s Eye: Cognitive and Applied Aspects of Eye Movement Research*. (Zitiert auf Seite 18.)
- Janecek, P. und Pu, P. (2003), „Searching with Semantics: An Interactive Visualization Technique for Exploring an Annotated Image Collection.“, in *OTM Workshops*, 2889, Lecture Notes in Computer Science (Springer), 185–196. (Zitiert auf Seite 37.)
- John, B. E. und Suzuki, S. (2009), „Toward Cognitive Modeling for Predicting Usability“, in Jacko, J., Herausgeber, *Human-Computer Interaction. New Trends*, 5610, Lecture Notes in Computer Science (Springer Berlin Heidelberg), 267–276. (Zitiert auf Seite 45.)
- John, B. E., Prevas, K., Salvucci, D. D., und Koedinger, K. (2004), „Predictive human performance modeling made easy“, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04 (New York, NY, USA: ACM), 455–462. (Zitiert auf Seite 45, 50, and 146.)
- Just, M. und Carpenter, P. (1976), „Eye Fixations and Cognitive Processes“, *Cognitive Psychology*, 8/4, 441–480. (Zitiert auf Seite 33.)
- Just, M. A. und Carpenter, P. A. (1980), „A theory of reading: from eye fixations to comprehension.“, *Psychological review*, 87/4, 329–354. (Zitiert auf Seite 33.)
- Kalogerakis, E., Christodoulakis, S., und Moutoutzis, N. (2006), „Coupling Ontologies with Graphics Content for Knowledge Driven Visualization“, in *Proceedings of the IEEE Virtual Reality Conference*, IEEE VR '06 (), 43–50. (Zitiert auf Seite 37.)
- Keim, D., Kohlhammer, J., Ellis, G., und Mansmann, F. (2010), *Mastering the Information Age: Solving Problems with Visual Analytics* (VisMaster, <http://www.vismaster.eu/book/>). (Zitiert auf Seite 7.)
- Kim, S.-H., Dong, Z., Xian, H., Upatizing, B., und Yi, J. S. (2012), „Does an Eye Tracker tell the truth about visualizations? Findings while investigating visualizations for decision making“, *IEEE Transactions on Visualization and Computer Graphics*, 18/12, 2421–2430. (Zitiert auf Seite 25.)

- Koerner, C. (2011), „Eye movements reveal distinct search and reasoning processes in comprehension of complex graphs“, *Applied Cognitive Psychology*, 25/6, 893–905. (Zitiert auf Seite 34, 85, and 86.)
- Kosslyn, S. M. (1986), *Image and Mind* (Harvard University Press). (Zitiert auf Seite 41.)
- Krejtz, K., Szmidt, T., Duchowski, A. T., und Krejtz, I. (2014), „Entropy-based Statistical Analysis of Eye Movement Transitions“, in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14 (New York, NY, USA: ACM), 159–166. (Zitiert auf Seite 27.)
- Kurzhals, K. und Weiskopf, D. (2013), „Space-Time visual analytics of eye-tracking data for dynamic stimuli“, *IEEE Transactions on Visualization and Computer Graphics*, 19/12, 2129 – 2138. (Zitiert auf Seite 23, 24, and 26.)
- Kurzhals, K. und Weiskopf, D. (2015), „AOI Transition Trees“, in *Proceedings of the 41st Graphics Interface Conference*, 2015 (Toronto, Ontario, Canada), 41–48. (Zitiert auf Seite 25.)
- Kurzhals, K., Heimerl, F., und Weiskopf, D. (2014), „ISeeCube: Visual Analysis of Gaze Data for Video“, in *Proceedings of the 2014 Symposium on Eye Tracking Research & Applications*, ETRA '14 (New York, NY, USA: ACM), 43–50. (Zitiert auf Seite 25, 26, 27, and 28.)
- Kurzhals, K., Hlawatsch, M., Heimerl, F., Burch, M., Ertl, T., und Weiskopf, D. (2015), „Gaze Stripes: Image-Based Visualization of Eye Tracking Data“, in *IEEE Transactions on Visualization and Computer Graphics* (to be published). (Zitiert auf Seite 23.)
- Latimer, C. R. (1988), „Eye-movement data: Cumulative fixation time and cluster analysis“, *Behavior Research Method, Instruments, & Computers*, 20/5, 437–470. (Zitiert auf Seite 23.)
- Levenshtein, V. I. (1965), „Binary Codes Capable of Correcting Deletions, Insertions, and Reversals“, *Doklady Akademii Nauk SSSR*, 163/4, 845–848. (Zitiert auf Seite 28.)
- Li, R., Pelz, J., Shi, P., Ovesdotter Alm, C., und Haake, A. (2012), „Learning Eye Movement patterns for characterization of perceptual expertise“, in *Proceedings of the 2012 Symposium on Eye Tracking Research & Applications*, ETRA '12 (New York, NY, USA: ACM), 393–396. (Zitiert auf Seite 26.)

- Li, X., Çöltekin, A., und Kraak, M.-J. (2010), „Visual Exploration of eye movement data using the space-time-cube“, in Fabrikant, S., Reichenbacher, T., Kreveld, M., und Christoph, S., Herausgeber, *Geographic Information Science*, 6292, LNCS '10 (Springer), 295–309. (Zitiert auf Seite 24 and 41.)
- Liu, Z. und Stasko, J. (2010), „Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective“, *IEEE Transactions on Visualization and Computer Graphics*, 16/6, 999–1008. (Zitiert auf Seite 42.)
- Lohmann, S., Ziegler, J., und Tetzlaff, L. (2009), „Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration“, in *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part I, INTERACT '09* (Berlin, Heidelberg: Springer-Verlag), 392–404. (Zitiert auf Seite 14.)
- Lohse, G. L. (1993), „A Cognitive Model for Understanding Graphical Perception“, *Human-Computer Interaction*, 8/4, 353–388. (Zitiert auf Seite 41.)
- Mackworth, J. F. und Mackworth, N. H. (1958), „Eye Fixations Recorded on Changing Visual Scenes by the Television Eye-Marker“, *Journal of the Optical Society of America*, 48/7, 439–444. (Zitiert auf Seite 23.)
- Manning, C. D., Raghavan, P., und Schütze, H. (2008), *Introduction to Information Retrieval* (New York, NY, USA: Cambridge University Press). (Zitiert auf Seite 69.)
- Marr, D. (1982), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (New York, NY, USA: Henry Holt and Co., Inc.). (Zitiert auf Seite 12.)
- Meyer, D. E. und Kieras, D. E. (1997), „A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms“, *Psychological Review*, 104/3, 3–65. (Zitiert auf Seite 42.)
- Miller, G. (1956), „The magical number seven plus minus two“, *Psychological Review*, 63/2, 81–97. (Zitiert auf Seite 35.)
- Neisser, U. (1979), *Kognition und Wirklichkeit: Prinzipien und Implikationen der kognitiven Psychologie* (Stuttgart: Klett-Cotta), 1. aufl. edition. (Zitiert auf Seite 35.)
- Neurowissenschaften, L. (2001), *Lexikon der Neurowissenschaften* (Spektrum Akademischer Verlag). (Zitiert auf Seite 12.)

- Norman, D. A. (1987), „Some observations on mental models“, in Baecker, R. M. und Buxton, W. A. S., Herausgeber, *Human-computer interaction* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), 241–244. (Zitiert auf Seite 6.)
- Noton, D. und Stark, L. (1971a), „Scanpaths in saccadic eye movements while viewing and recognizing patterns“, *Vision Research*, 11/9, 929–942. (Zitiert auf Seite 24.)
- Noton, D. und Stark, L. (1971b), „Scanpaths in eye movements during pattern perception.“, *Science*, 171/3968, 308–311. (Zitiert auf Seite 24.)
- Pasternak, T., Bisley, J. W., und Calkins, D. (2003), „Visual Processing in the Primate Brain“, in *Handbook of Psychology* (John Wiley & Sons, Inc.). (Zitiert auf Seite 12.)
- Patel, D., Sture, O., Hauser, H., Giertsen, C., und Gröeller, M. E. (2009), „Knowledge-assisted visualization of seismic data.“, *Computers & Graphics*, 33/5, 585–596. (Zitiert auf Seite 37.)
- Peebles, D. (2012), „A cognitive architecture-based model of graph comprehension“, in *Proceedings of 11th International Conference on Cognitive Modeling* (Berlin, Germany: International Conference on Cognitive Modeling), 37–42. (Zitiert auf Seite 45 and 109.)
- Peebles, D. und Cheng, P. C.-H. (2002), „Extending task analytic models of graph-based reasoning: A cognitive model of problem solving with Cartesian graphs in ACT-R/PM.“, *Cognitive Systems Research*, 3/1, 77–86. (Zitiert auf Seite 44 and 109.)
- Perini, L. (2005), „The Truth in Pictures“, *Philosophy of Science*, 72/1, 262–285. (Zitiert auf Seite 87.)
- Pettitt, M., Burnett, G., und Stevens, A. (2007), „An extended keystroke level model (KLM) for predicting the visual demand of in-vehicle information systems“, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07 (New York, NY, USA: ACM), 1515–1524. (Zitiert auf Seite 40.)
- Pinker, S. (1990), „A Theory of Graph Comprehension“, in Freedle, R., Herausgeber, *Artificial Intelligence and the Future of Testing* (Lawrence Erlbaum Assoc Inc), 73–126. (Zitiert auf Seite 37, 39, 41, 89, and 113.)
- Pohl, M., Schmitt, M., und Diehl, S. (2009), „Comparing the Readability of Graph Layouts using Eyetracking and Task-oriented Analysis“, in *Proceedings of the Fifth Eurographics conference on Computational Aesthetics*

- in *Graphics, Visualization and Imaging* (Aire-la-Ville, Switzerland), 49–56. (Zitiert auf Seite 14.)
- Poole, A. und Ball, L. J. (2005), „Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future“, in *Prospects, Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc* (). (Zitiert auf Seite 18.)
- Prats, M., Garner, S., Jowers, I., McKay, A., und Pedreira, N. (2010), „Interpretation of geometric shapes: an eye movement study.“, in *ETRA '10* (New York, NY, USA: ACM), 243–250. (Zitiert auf Seite 14.)
- Privitera, C. und Stark, L. (2000), „Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22/9, 970–982. (Zitiert auf Seite 20, 27, 28, and 93.)
- Quillian, M. R. (1966), *Semantic Memory* (Cambridge, MA: Bolt, Beranak and Newman). (Zitiert auf Seite 35.)
- Räihä, K.-J., Aula, A., Majaranta, P., Rantala, H., und Koivunen, K. (2005), „Static Visualization of Temporal Eye-tracking data“, in Costabile, M. F. und Paternò, F., Herausgeber, *Lecture Notes in Computer Science*, 3585, LNCS '05 (Springer), 946–949. (Zitiert auf Seite 25.)
- Rao, S. C., Rainer, G., und Miller, E. K. (1997), „Integration of what and where in the primate prefrontal cortex.“, *Science (New York, N.Y.)*, 276 /5313, 821–824. (Zitiert auf Seite 12.)
- Raschke, M., Chen, X., und Ertl, T. (2012), „Parallel scan-path visualization“, in *Proceedings of the 2012 Symposium on Eye Tracking Research & Applications*, ETRA '12 (New York, NY, USA: ACM), 165–168. (Zitiert auf Seite 3 and 52.)
- Raschke, M., Engelhardt, S., und Ertl, T. (2013), „A Framework for Simulating Visual Search Strategies“, in *Proceedings of the 11th International Conference on Cognitive Modeling* (Ottawa, Canada). (Zitiert auf Seite 4 and 53.)
- Raschke, M., Blascheck, T., und Ertl, T. (2014a), „Cognitive Ergonomics in Visualization“, in Ebert, A., van der Veer, G. C., Domik, G., Gershon, N. D., und Scheler, I., Herausgeber, *Building Bridges: HCI, Visualization, and Non-formal Modeling* (Berlin, Heidelberg: Springer), 80–94. (Zitiert auf Seite 3 and 52.)

- Raschke, M., Blascheck, T., Richter, M., Agapkin, T., und Ertl, T. (2014b), „Visual Analysis of Perceptual and Cognitive Processes“, in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2014, IVAAP '14 (Lisboa, Portugal). (Zitiert auf Seite 4, 52, and 131.)
- Raschke, M., Blascheck, T., Strohmaier, S., und Ertl, T. (2014c), „Annotation of Graphical Elements in Visualizations for an Efficient Analysis of Visual Tasks“, in *CHI 2013 Extended Abstracts*, CHI '11 EA (New York, NY, USA: ACM), 1387–1398. (Zitiert auf Seite 4 and 52.)
- Raschke, M., Herr, D., Blascheck, T., Burch, M., Schrauf, M., Willmann, S., und Ertl, T. (2014d), „A Visual Approach for Scan Path Comparison“, in *Proceedings of the 2014 Symposium on Eye Tracking Research & Applications*, ETRA '14 (New York, NY, USA: ACM), 135–142. (Zitiert auf Seite 3 and 52.)
- Rayner, K. (1998), „Eye movements in reading and information processing: 20 years of research.“, *Psychological bulletin*, 124/3, 372–422. (Zitiert auf Seite 85, 86, and 134.)
- Rensink, R. A. (2002), „Change Detection“, *Annual Review of Psychology*, 53, 245–277. (Zitiert auf Seite 12.)
- Richardson, D. C. und Dale, R. (2005), „Looking To Understand: The Coupling Between Speakers' and Listeners' Eye Movements and Its Relationship to Discourse Comprehension“, *Cognitive Science*, 29/6, 1045–1060. (Zitiert auf Seite 25.)
- Ristovski, G., Hunter, M., Olk, B., und Linsen, L. (2013), „EyeC: Coordinated Views for Interactive Visual Data Exploration of Eye-Tracking Data“, in *Proceedings of the 17th International Conference on Information Visualisation*, IV '13 (London, Great Britain), 239–248. (Zitiert auf Seite 25 and 26.)
- Rodrigues, R., Veloso, A., und Mealha, O. (2012), „A Television News Graphical Layout Analysis Method Using Eye Tracking“, in *Proceedings of the 16th International Conference on Information visualization*, IV '12 (Montpellier, France), 357–362. (Zitiert auf Seite 20.)
- Rosch, E. (1975), „Cognitive representations of semantic categories.“, *Journal of Experimental Psychology: General*, 104/3, 192–233. (Zitiert auf Seite 113.)
- Rosch, E. H. (1973), „Natural categories“, *Cognitive Psychology*, 4/3, 328 – 350. (Zitiert auf Seite 113.)

- Rosenholtz, R., Li, Y., Mansfield, J., und Jin, Z. (2005), „Feature Congestion: a Measure of Display Clutter“, in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (New York, NY, USA: ACM), 761–770. (Zitiert auf Seite 24, 49, and 81.)
- Salvucci, D. D. (2001), „An integrated model of eye movements and visual encoding“, *Cognitive Systems Research*, 1/4, 201–220. (Zitiert auf Seite 126.)
- Salvucci, D. D. und Goldberg, J. H. (2000), „Identifying Fixations and Saccades in Eye-tracking Protocols“, in *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00 (New York, NY, USA: ACM), 71–78. (Zitiert auf Seite 18.)
- Santella, A. und DeCarlo, D. (2004), „Robust clustering of eye movement recordings for quantification of visual interest“, in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04 (New York, NY, USA: ACM), 27–34. (Zitiert auf Seite 20.)
- Schneegaß, S., Pfleging, B., Kern, D., und Schmidt, A. (2011), „Support for modeling interaction with automotive user interfaces“, in *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '11 (New York, NY, USA: ACM), 71–78. (Zitiert auf Seite 40.)
- Scinto, L. F., Pillalamarri, R., und Karsh, R. (1986), „Cognitive strategies for visual search“, *Acta Psychologica*, 62/3, 263 – 292. (Zitiert auf Seite 24.)
- Shah, P. (1997), „A model of the cognitive and perceptual processes in graphical display comprehension.“, in *Proc. American Association for Artificial Intelligence Spring Symposium*, AAAI Technical Report FS-97-03 (Stanford University). (Zitiert auf Seite 42.)
- Simkin, D. und Hastie, R. (1987), „An Information-Processing Analysis of Graph Perception“, *Journal of the American Statistical Association*, 82 /398, 454–465. (Zitiert auf Seite 41.)
- Špakov, O. (2008), *iComponent - Device-Independent Platform for Analyzing Eye Movement Data and Developing Eye-Based Applications* (Tampere and Finnland: Department of Computer Science). (Zitiert auf Seite 25.)
- Špakov, O. und Miniotas, D. (2007), „Visualization of eye gaze data using heat maps“, *Electronics and Electrical Engineering*, 2/74, 55–58. (Zitiert auf Seite 23.)

- Sperling, G. (1960), „The Information available in brief visual Representations“, in *Psychological Monographs: General and Applied*, 74 (American Psychological Association), 1–29. (Zitiert auf Seite 35.)
- Steinberger, M., Waldner, M., Streit, M., Lex, A., und Schmalstieg, D. (2011), „Context-Preserving Visual Links.“, *IEEE Transactions on Visualization and Computer Graphics*, 17/12, 2249–2258. (Zitiert auf Seite 14.)
- Teo, L. und John, B. E. (2006), „Comparisons of keystroke-level model predictions to observed data“, in *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06 (New York, NY, USA: ACM), 1421–1426. (Zitiert auf Seite 40.)
- Teo, L. und John, B. E. (2008), „Cogtool-explorer: Towards a Tool for Predicting User Interaction“, in *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08 (New York, NY, USA: ACM), 2793–2798. (Zitiert auf Seite 45.)
- Thomas, J. und Kielman, J. (2009), „Challenges for visual analytics“, *Information Visualization*, 8/4, 309–314. (Zitiert auf Seite 8.)
- Tobii Technology AB (2008), Tobii Studio 1.X User Manual. (Zitiert auf Seite 23.)
- Toker, D., Conati, C., Steichen, B., und Carenini, G. (2013), „Individual User Characteristics and Information Visualization: Connecting the Dots Through Eye Tracking“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13 (New York, NY, USA: ACM), 295–304. (Zitiert auf Seite 14.)
- Toro, R., Perron, M., Pike, B., Richer, S. L. Veillette, Pausova, Z., und Paus, T. (2008), „Brain size and folding of the human cerebral cortex“, *Cerebral cortex*, 18/10, 2352–2357. (Zitiert auf Seite 31.)
- Tory, M. und Möller, T. (2004), „Human Factors in Visualization Research“, *IEEE Transactions on Visualization and Computer Graphics*, 10/1, 72–84. (Zitiert auf Seite 150 and 151.)
- Treisman, A. (1985), „Preattentive Processing in Vision“, *Computer vision, graphics, and image processing*, 31/2, 156–177. (Zitiert auf Seite 12.)
- Trewin, S., John, B. E., Richards, J., Swart, C., Brezin, J., Bellamy, R., und Thomas, J. (2010), „Towards a Tool for Keystroke Level Modeling of Skilled Screen Reading“, in *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10 (New York, NY, USA: ACM), 27–34. (Zitiert auf Seite 45.)

- Triesch, J., Ballard, D. H., Hayhoe, M. M., und Sullivan, B. T. (2003), „What you see is what you need“, *Journal of Vision*, 3/1, 86–94. (Zitiert auf Seite 34.)
- Tsang, H. Y., Tory, M. K., und Swindells, C. (2010), „eSeeTrack - Visualizing Sequential Fixation Patterns“, *IEEE Transactions on Visualization and Computer Graphics*, 16/6, 953–962. (Zitiert auf Seite 26.)
- Tulving, E. (1972), „Episodic and semantic Memory“, in Tulving, E. und Wayne, D., Herausgeber, *Organization of Memory* (New York), 381–403. (Zitiert auf Seite 34.)
- Ullman, S. (1984), „Visual routines“, *Cognition*, 18, 97–159. (Zitiert auf Seite 41.)
- Underwood, G., Chapman, P., Brocklehurst, N., Underwood, J., und Crundall, D. (2003), „Visual attention while driving: sequences of eye fixations made by experienced and novice drivers“, *Ergonomics*, 46/6, 629–646. (Zitiert auf Seite 34.)
- Velichkovsky, B. M. und Hansen, J. P. (1996), „New Technological Windows into Mind: There is More in Eyes and Brains for Human-computer Interaction“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96 (New York, NY, USA: ACM), 496–503. (Zitiert auf Seite 23.)
- Ware, C. (2008), *Visual Thinking: for Design* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.). (Zitiert auf Seite 6.)
- Ware, C. (2013), *Information Visualization: Perception for Design* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), 3 edition. (Zitiert auf Seite 6, 9, and 11.)
- Weibel, N., Fouse, A., Emmenegger, C., Kimmich, S., und Hutchins, E. (2012), „Let's look at the Cockpit: Exploring Mobile Eye-Tracking for Observational Research on the Flight Deck“, in *Proceedings of the 2012 Symposium on Eye Tracking Research & Applications*, ETRA '12 (New York, NY, USA: ACM), 107–114. (Zitiert auf Seite 24 and 25.)
- Wertheimer, M. (1923), „Untersuchungen zur Lehre von der Gestalt“, *Psychologische Forschung*, 4, 301–350. (Zitiert auf Seite 125.)
- West, J. M., Haake, A. R., Rozanski, E. P., und Karn, K. S. (2006), „eyePatterns: software for identifying patterns and similarities across fixation sequences“, in *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, ETRA '06 (New York, NY, USA: ACM), 149–154. (Zitiert auf Seite 26.)

- Wickens, C. D. (1992), *Engineering Psychology and Human Performance* (New York: HarperCollins), 2 edition. (Zitiert auf Seite 10.)
- Williams, L. J. (1985), „Tunnel Vision Induced by a Foveal Load Manipulation“, *Human Factors*, 27/2, 221–227. (Zitiert auf Seite 10.)
- Wolfe, J. M. (1994), „Guided Search 2.0: A revised model of visual search“, *Psychonomic Bulletin and Review*, 1/2, 202–238. (Zitiert auf Seite 42.)
- Wolfe, J. M. (2007), „Guided Search 4.0: Current Progress with a model of visual search“, in Gray, W., Herausgeber, *Integrated Models of Cognitive Systems* (New York: Oxford), 99–119. (Zitiert auf Seite 42.)
- Wolfe, J. M. und Gancarz, G. (1997), „Guided Search 3.0: a model of visual search catches up with Jay Enoch 40 years later“, in Lakshminarayanan, V., Herausgeber, *Basic and Clinical Applications of Vision Science* (Dordrecht, Netherlands: Kluwer Academic Publishers), 189–192. (Zitiert auf Seite 42.)
- Wooding, D. S. (2002), „Fixation maps: quantifying eye-movement traces“, in *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02 (New York, NY, USA: ACM), 31–36. (Zitiert auf Seite 23.)
- Yarbus, A. L. (1967), *Eye Movements and Vision* (Plenum Press). (Zitiert auf Seite 24.)
- Zimbardo, P. G. und Gerrig, R. J. (2008), *Psychologie* (Pearson Studium), 18., aktualisierte auflage edition. (Zitiert auf Seite 9.)