

Visualisierungsinstitut der Universität Stuttgart
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3343

Dynamische Annotation mit Eyetracking

Giuseppina Maringolo

Studiengang: Informatik
Prüfer: Prof. Dr. Daniel Weiskopf
Betreuer: Dipl.-Inf. Julian Heinrich

begonnen am: 01. Juni 2012
beendet am: 01. Dezember 2012

CR-Klassifikation: H.5.2, I.3.6, I.3.3

Kurzfassung

Im Bereich der Mensch-Computer-Interaktion gewinnt Eyetracking immer mehr an Bedeutung. Dabei lag der Fokus zu Beginn auf motorisch eingeschränkten Nutzern, um ihnen den Zugang zu verschiedenen Applikationen zu erleichtern. Mittlerweile richten sich Forschungsarbeiten aber auch auf blickbasierte Schnittstellen, die sich nicht ausschließlich auf körperlich beeinträchtigte Personen orientieren.

In der vorliegenden Diplomarbeit wird eine neue Eyetracking-Technik zur Selektion diskreter Punktobjekte in Scatterplots umgesetzt, welche Beschriftungen von Punktobjekten innerhalb der Fokussierung dynamisch einblendet. Die neue Technik wird durch eine Studie mit zwei bestehenden Techniken verglichen. Die erste bestehende Technik stellt im Gegensatz zur neuen Eyetracking-Technik die Beschriftungen statisch dar, während die zweite Technik die Mauskoordinaten zur dynamischen Auslösung der Beschriftungen verwendet. Eine kritische Betrachtung erfolgt nun darüber, ob die neue Technik gewisse Vorteile für das Lokalisierungsproblem in Scatterplots in Bezug zur Performanz und zur Genauigkeit liefert.

Diese Arbeit beschäftigt sich mit Eyetracking sowie mit den wesentlichen Kriterien und Problemstellungen aus der Geoinformatik und der automatisierten Textplatzierung. Zudem werden die eingesetzten Algorithmen für die statische und dynamische Beschriftung von Informationsobjekten sowie deren Implementierung vorgestellt. Gründe für den Einsatz von Eyetracking zur Interaktion statt der Maus werden erläutert. Die Durchführung einer Eyetracking-Studie zum Vergleich der Techniken bildet einen weiteren Abschnitt der Arbeit. Die Auswertung und die Präsentation der Ergebnisse der Studie schließen diese Arbeit ab.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Einführung	9
1.2	Motivation und Aufgabenstellung	9
1.3	Gliederung	10
2	Eyetracking	11
2.1	Grundlagen	11
2.2	Visuelles System des Menschen	12
2.3	Geschichte	15
2.4	Eyetracking-Techniken	16
2.4.1	Videobasierte Eyetracker	17
2.4.2	Funktionsweise eines ferngesteuerten Eyetrackers	18
2.5	Anwendungsgebiete	19
3	Labeling	23
3.1	Kriterien zur Platzierung von Labels	23
3.2	Labeling von Punktobjekten	24
3.3	Problemstellungen beim Labeling	25
3.4	Scatterplots	29
4	Automatische Platzierung von Labels	33
4.1	Statisches Labeling	33
4.1.1	Erschöpfende Suche	36
4.1.2	Discrete Gradient Descent	36
4.1.3	Algorithmus von Hirsch	37
4.1.4	Simulated Annealing	40
4.2	Dynamisches Labeling	40
4.2.1	One at a time-Techniken	42
4.2.2	Änderung der Gesamtansicht	43
4.2.3	Übersicht & Detail-Darstellung	43
4.2.4	Fokus- & Kontext-Darstellung	44
4.2.5	Panning	46
4.2.6	Dynamic Sampling	47
4.2.7	Excentric Labeling	47
5	Dynamisches Labeling mit Eyetracking	51
5.1	Auswahl eines Labeling-Algorithmus	51

5.2	Eyetracking zur Interaktion	52
5.3	Performanz von Eyetracking und Maus	53
5.4	Hypothesen	55
5.5	Verwandte Arbeiten	55
6	Implementierung	59
6.1	Anforderungen	59
6.2	Verwendete Technologien	60
6.2.1	Processing	60
6.2.2	Processing Easy Eye Tracker Plugin	61
6.2.3	gwoptics	62
6.3	Einfaches Scatterplot	62
6.4	Dynamisches Labeling mit Excentric Labeling	64
6.5	Statisches Labeling mit Simulated Annealing	67
7	Eyetracking-Studie	71
7.1	Studienkonzept	71
7.2	Testdaten und Aufgabenstellungen	72
7.3	Pilotstudie	73
7.4	Finale Studie	74
7.5	Statistische Auswertung	77
8	Schlussfolgerung	81
8.1	Überprüfung der Hypothesen	81
8.2	Diskussion über die Ergebnisse der Studie	82
9	Zusammenfassung und Ausblick	85
	Literaturverzeichnis	87

Abbildungsverzeichnis

2.1	Aufbau des Auges	12
2.2	Blickbewegungen beim Lesen	14
2.3	EyeTracker-System Tobii T60 XL	18
3.1	Eindeutigkeit zwischen Label und Objekt	24
3.2	Regeln von Imhof	25
3.3	Regeln von Yoeli	25
3.4	Festpositionenmodelle	26
3.5	Elastisches Rechteck beim Elastic Labeling Problem	27
3.6	Anordnung der Rechtecke beim Rectangle Perimeter Problem	28
3.7	Schieber-Modelle	28
3.8	Einfaches Scatterplot	30
3.9	Scatterplot-Matrix	31
3.10	3D-Scatterplot	32
4.1	Beschriftungspositionen mit Prioritäten	36
4.2	Lokales Minimum beim Discrete Gradient Descent Algorithmus	37
4.3	Potentielle Beschriftungspositionen	38
4.4	Lokales Minimum beim Algorithmus von Hirsch	39
4.5	Ergebnis der statischen Algorithmen	41
4.6	Übersicht & Detail-Darstellung	44
4.7	Fisheye View für Fokus & Kontext-Darstellung	45
4.8	Space-Scales-Diagramm	46
4.9	Vertically Coherent Labeling	48
4.10	Non-Crossing Lines Labeling	48
4.11	Horizontally Coherent Labeling	49
5.1	EyeWindows	56
5.2	Text 2.0	57
5.3	Subtle Gaze Direction	58
6.1	Koordinatensystem von Processing	61
6.2	Werte auf den Scatterplot-Achsen	63
6.3	Scatterplot mit dynamischer Beschriftung	66
6.4	Dynamische Beschriftung für ein Punktobjekt im Fokus	66
6.5	Scatterplot mit statischer Beschriftung	70

7.1	Korrelation, Ausreißer und Cluster in Scatterplots	76
7.2	Reihenfolge der Techniken nach dem Latin Square	76
7.3	Q-Q-Plot für Antwortzeiten	78
7.4	Boxplots für Antwortzeiten pro Datensatz	79
7.5	Q-Q-Plot für Genauigkeitswerte	80
7.6	Boxplots für Genauigkeitswerte pro Technik	80
8.1	Scatterplot für statische Technik und Eyetracking-Technik	82
8.2	Boxplot für Probanden mit Eyetracking-Erfahrung	83

Verzeichnis der Listings

6.1	Processing Hauptfunktionen	60
6.2	PEEP für Processing	61
6.3	Timer in Scatterplot	64
6.4	Processing loop() und noloop()	67
6.5	Überprüfung auf Überlappungen in Processing	69

Verzeichnis der Algorithmen

4.1	Discrete Gradient Descent	37
6.1	Excentric Labeling	65
6.2	Simulated Annealing	68

1 Einleitung

1.1 Einführung

Die Informationsvisualisierung ist ein Forschungsgebiet, das sich mit der grafischen Repräsentation von Daten befasst, um strukturelle Zusammenhänge und relevante Eigenschaften zu erlangen und enorme Informationsbestände zu explorieren. Mittlerweile existieren in der Informationsvisualisierung auch Ansätze, die sich auf Informationsdarstellungen im Raum, wie auf geografische Karten, und auf die Visualisierung von zeitlichen Veränderungen konzentrieren [FS04].

Die Informationsvisualisierung befasst sich unter anderem mit platzsparenden Anzeigetechniken, da eine begrenzte Auflösung der Ausgabegeräte auch eine Begrenzung der darstellbaren Informationen verursacht. Zu diesen Techniken zählen *Übersicht-* und *Detail-* sowie *Fokus-* und *Kontext-Darstellungen* [SM04]. Die Übersicht- und Detail-Darstellung trennt Überblicks- und Zusatzinformationen in zwei Ansichten. Die Fokus- und Kontext-Darstellung stellt dagegen die Information direkt an den Stellen dar, die sich im Fokus des Interesses befinden. Um Objekte in der Informationsvisualisierung identifizieren zu können, ist eine automatisierte Platzierung ihrer Beschriftungen bzw. Annotationen (engl. Labels) eine notwendige Voraussetzung. Die automatische Platzierung von Labels ist bereits aus der Kartografie bekannt. Für die Lösung unterschiedlicher Labeling-Probleme existieren bereits verschiedene Heuristiken. Diese liefern jedoch nicht immer optimale Ergebnisse, vor allem bei größeren Datensätzen auf einer begrenzten Fläche, wenn Labels durch Überlappungen unlesbar werden oder Objekte mit ihren Labels nicht eindeutig zugeordnet werden können. Zudem werden auch Objekte beschriftet, die nicht von Interesse sind, also nicht im Fokus des Betrachters liegen. Diese stören jedoch die Wahrnehmung.

1.2 Motivation und Aufgabenstellung

Das genannte Problem ist nicht nur bei geografischen Karten bekannt, sondern auch bei statistischen Diagrammen, wie zum Beispiel Streudiagrammen (engl. Scatterplots), welche auch überschneidende Datenpunkte enthalten können [LAE⁺10]. Hier besteht ebenfalls das Problem, dass Labels sich überlappen und somit unlesbar werden, falls Datenpunkte dicht beieinander liegen. Erwünscht ist jedoch eine Anordnung der Labels, die sich nicht überlappen und dadurch lesbar werden. Weiterhin sollen Labels mit den entsprechenden Datenpunkten eindeutig in Verbindung gebracht werden können.

Ziel dieser Diplomarbeit ist die Implementierung einer neuen dynamischen Technik mit Hilfe von Eyetracking-Daten, die das Problem der Überlappung von Labels sowie die Assoziation der Labels mit ihren Datenpunkten in Scatterplots angeht. Da bekannt ist, dass Eyetracking in manchen Fällen schneller bei der Selektion von groben Bereichen ist als die Maus, wird für die neue Technik ein Eyetracker zur Interaktion eingesetzt [Vero8]. Über die Blicksteuerung können somit nur die Labels einzelner Informationsobjekte dargestellt werden, die im Fokus des Betrachters liegen und somit von Interesse sind. Dadurch kann außerhalb der Fokussierung der Platz zur Darstellung aller Labels innerhalb des Blickwinkels genutzt werden, ohne unlesbare Labels zu erzeugen. Die neue dynamische Technik soll zudem mit zwei bereits bestehenden Techniken durch eine Eyetracking-Studie bezüglich der Zeit und der Genauigkeit bei der Auslösung von Labels verglichen werden. Die erste bestehende Technik stellt eine statische Technik dar, bei der alle Labels immer im Scatterplot eingeblendet sind. Für die zweite bestehende Technik werden Maus- statt Augkoordinaten zur Auslösung der Labels im Scatterplot verwendet.

1.3 Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Eyetracking stellt eine Einführung in das Thema Eyetracking dar.

Kapitel 3 – Labeling befasst sich mit der Beschriftung von Objekten in der Kartografie, die für diese Arbeit relevant sind (insbesondere für Kapitel 4 und 6). Wichtige Kriterien zur Textplatzierung sowie Problemstellungen bei Punktbeschriftungen werden erläutert. Zudem werden unterschiedliche Klassen von Scatterplots behandelt, die eine Informationsdarstellung von Punktobjekten ermöglichen.

Kapitel 4 – Automatische Platzierung von Labels stellt mögliche Algorithmen und Techniken vor, die Beschriftungen automatisch platzieren. Hierbei wird zwischen statischer und dynamischer Beschriftung unterschieden.

Kapitel 5 – Dynamisches Labeling mit Eyetracking begründet den Einsatz von Eyetracking für die neue Technik zur dynamischen Beschriftung und stellt Hypothesen für die neue Technik auf. Zudem werden verwandte Arbeiten aufgeführt, die als Motivation für den Einsatz von Eyetracking in dieser Arbeit dienen.

Kapitel 6 – Implementierung zeigt, wie die ausgewählten Algorithmen aus Kapitel 4 praktisch umgesetzt werden. Es wird auf die Implementierung von Scatterplots und auf die Implementierung von statischen und dynamischen Beschriftungen mit der Programmiersprache *Processing* eingegangen.

Kapitel 7 – Eyetracking-Studie beschreibt das Vorgehen während der Eyetracking-Studie.

Kapitel 8 – Schlussfolgerung erläutert, wie die Ergebnisse der Studie interpretiert werden.

Kapitel 9 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2 Eyetracking

Dieses Kapitel stellt eine Einführung in das Thema Eyetracking dar und behandelt verschiedene Bereiche von Eyetracking. Seit mehr als 100 Jahren ist Eyetracking ein Forschungsgebiet [PB05]. Das Forschungsgebiet erstreckt sich mittlerweile von der Medizin über die Usability bis hin zur Mensch-Computer-Interaktion [Nau12].

2.1 Grundlagen

Unter *Eyetracking* (*Blickerfassung* bzw. *Okulographie*) wird das Aufzeichnen von Blickbewegungen einer Person bei der Betrachtung von Stimuli verstanden. Somit kann die visuelle Erfassung eines Objekts erkannt werden. Das Messgerät, das die Augenbewegungen aufzeichnet, wird als *Eyetracker* bezeichnet. Dabei werden verschiedene Arten von Aufzeichnungsgeräten unterschieden, die in Kapitel 2.4 näher beschrieben werden. Des Weiteren existieren mehrere Anwendungsgebiete von Eyetracking, die in Kapitel 2.5 erläutert werden [Nau12].

Es gibt einige Gründe für den Einsatz von Eyetracking-Methoden [HV03]:

Es kann festgestellt werden, ob eine Person einen Text liest oder ihn nur überfliegt. Bildfixierungen einer Person werden erkannt (z.B. Werbebanner). Die Reihenfolge der Bildfixierungen, die sich aus der Abfolge von Fixationen (genau Punkt Betrachtungen) und Sakkaden (schnelle Augenbewegungen) ergeben, können verfolgt werden. Es kann gemessen werden, wie lange eine Person eine *Region of Interest* (ROI) betrachtet. Die Suchstrategie beim Lösen von Aufgaben kann ermittelt werden. Andere Methoden, wie Interviews oder Fragebögen, können ergänzt werden.

In der Praxis fallen jedoch auch Schwierigkeiten mit solchen Methoden an [HV03]:

Ein hoher technischer Aufwand und hohe finanzielle Kosten können entstehen. Außerdem ist die Analyse und Interpretation von Eyetracking-Daten aufgrund von fehlenden Standards schwierig. Es ist nicht messbar, ob eine Person tatsächlich etwas wahrgenommen hat oder nicht, da über die Fixationsdauer des Auges kein Aufschluss über die Wahrnehmung gegeben werden kann. Die Aufmerksamkeit wird selektiert, sodass irrelevante Informationen ignoriert werden, da nicht alle Umgebungsreize parallel aufgenommen werden können. Nicht alle Eyetracking-Methoden können die Blicke von Brillenträgern erfassen. Manche Verfahren erfordern den Einsatz von Kontaktlinsen. Die Verarbeitung von Information erfolgt auch außerhalb des fixierten Objekts durch peripheres Sehen, die durch Eyetracking nicht aufgenommen werden können, da Eyetracking nur foveale Daten erfasst. Ein Eyetracker

zeichnet also nur den Blickfokus auf und nicht die Informationen aus dem peripheren Blickfeld.

Beim Eyetracking ist die Bewegung des Auges von großer Bedeutung. Aufgrund dessen wird auf die Anatomie des Auges sowie auf die verschiedenen Arten von Augen- und Blickbewegungen im nächsten Abschnitt eingegangen.

2.2 Visuelles System des Menschen

Die visuelle Wahrnehmung beim Menschen wird aufgrund der Rezeptoren und Nervenbahnen durch das Auge bestimmt, sodass es einen hohen Informationsfluss besitzt. Das Auge besteht aus mehreren Komponenten, die im Folgenden beschrieben werden.

Aufbau des Auges

Die Augen befinden sich in der sogenannten Augenhöhle des Kopfes, sind kugelförmig und haben einen Durchmesser von 24 Millimeter [Küo4]. Abbildung 2.1 zeigt den Aufbau des menschlichen Auges. Die Cornea (dt.: Hornhaut) stellt eine lichtdurchlässige, gewölbte Augenhaut dar, die als Schutz für die Pupille, Iris und Linse dient. Die Iris, auch Regenbogenhaut genannt, ist dafür zuständig, die Größe der Pupille zu regulieren und somit die Menge des durchdringenden Lichtes zu steuern. Bei Dunkelheit, also bei einem geringen Lichtverhältnis, öffnet die Iris die Pupille weiter. Bei Helligkeit wird die Pupille verengt. Solch eine Anpassung nennt man auch *Hell-Dunkel-Adaption*. Die Ziliarmuskeln können die

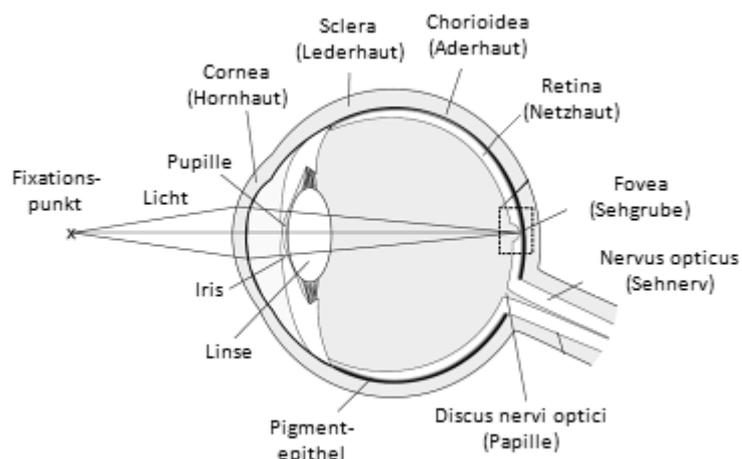


Abbildung 2.1: Aufbau des Auges (siehe [PBM09, Seite 149])

Wölbung der Augenlinse verformen. Die Linse bündelt das Licht im hinteren Bereich des Auges, das durch die Pupille durchdringt. Durch die Ziliarmuskeln können ferne Objekte scharf gesehen werden, wenn die Linse flach ist. Wenn die Linse gewölbter ist, dann können nahe Objekte scharf betrachtet werden. Das automatische Verändern der Linsenwölbung wird *Akkommodation* genannt. Der Glaskörper bildet das Innere des Augapfels und enthält eine gelartige, durchsichtige Substanz. Er sorgt für einen Abstand zwischen Netzhaut und Augenlinse und leitet das Licht auf die Retina weiter. Die Retina (dt.: Netzhaut) enthält Fotorezeptoren, die sogenannten Stäbchen und Zäpfchen. Zäpfchen sind farbempfindlich und dienen für das *fotopische Sehen* (Tagessehen), während Stäbchen für die Grauwerte und somit für das *skotopische Sehen* (Nachtsehen) zuständig sind. Auf der Retina existieren zwei besondere Bereiche: Die *Fovea Centralis*, auch *gelber Fleck* genannt, ist aufgrund der höchsten Zapfendichte der Ort des besonders scharfen Sehens. Der *blinde Fleck* ist frei von Fotorezeptoren, da an diesem Ort der Sehnerv austritt. An dieser Stelle ist das Auge blind, deshalb auch „blinder Fleck“. Die visuelle Information auf den Fotorezeptoren wird vom Opticus (dt.: Sehnerv) zum Gehirn weitergegeben und dort als Bild zusammengestellt. Die Chorioidea (dt.: Aderhaut) hinter der Retina dient als Versorgungssystem der Netzhaut. Die Sclera (dt.: Lederhaut) schützt das Auge von außen [Nau12].

Durch diese Komponenten hat das menschliche Auge ein breites Farbspektrum. Das Auge kann sichtbares Licht von rot bis blau wahrnehmen, das in Wellenlängen gemessen wird und sich von ca. 380-780 nm erstreckt. Kürzere Wellenlängen (ultraviolett) und längere Wellenlängen (infrarot) sind für das Auge jedoch unsichtbar [Nau12].

Augen- und Blickbewegungen

Durch die Augenmuskeln kann das Auge eine Vielzahl von unterschiedlichen Bewegungen ausführen. Man unterscheidet dabei zwischen Augen- und Blickbewegungen. Augenbewegungen stehen für die alleinige Bewegung des Augapfels. Blickbewegungen sind Bewegungen des Auges, die mit dem Wahrnehmungsprozess verknüpft werden.

Blickbewegungen lassen sich anhand ihrer Charakteristik in mehrere Kategorien unterteilen (vgl. [HKMS11], [SGSo8], [Nau12]):

Eine *Fixation* ist definiert als der Zustand, bei dem das Auge sich bezüglich eines Objekts in relativem Stillstand befindet. Grundsätzlich werden dabei die aufgenommenen Informationen in die Fovea Centralis gerückt und anschließend vom Gehirn verarbeitet. Dabei stellen 90 Prozent der Blickbewegungen Fixierungen dar. Die Dauer einer Fixierung beträgt ca. 150 bis 600 Millisekunden. Eine visuelle Wahrnehmung kann nur während einer Fixierung stattfinden. *Sakkaden* stellen ruckartige Bewegungen des Auges von einer Fixation zur anderen dar, die dem Erfassen eines neuen Fixpunktes dienen. Sie treten beim Betrachten von Objekten oder beim Lesen auf. Pro Sekunde kommt es etwa zu drei bis vier Sakkaden. Sakkaden haben eine Dauer von 10 bis 100 Millisekunden, sodass die visuelle Wahrnehmung eingeschränkt wird und deshalb die Aufnahme von Information verhindert wird, das als *sakkadische Suppression* bekannt ist. *Regressionen* sind Rücksprünge des Auges beim Lesen, wenn bereits Punkte fixiert wurden. Bei *kontinuierlichen Folgebewegungen* verfolgt das Auge

ein sich bewegendes Objekt. *Mikrosakkaden* treten innerhalb von Fixationen auf, wenn ein unbewegter Gegenstand angestarrt wird. Das Auge steht in diesem Fall nicht still, sondern bewegt sich minimal und springt ruckartig herum. Solche geringen Bewegungen werden vom Gehirn nicht wahrgenommen, sind jedoch notwendig, damit die Rezeptoren im Auge ständig stimuliert werden, da ansonsten das Bild langsam verschwinden würde. *Nystagmen* sind schnelle Augenbewegungen, die bei Kopfbewegungen oder beim Verfolgen von sich langsam bewegenden Objekten auftreten. *Vergenzbewegungen* ändern die Stellung der Augen zueinander, um ein Objekt zu fixieren, sodass ein scharfes Bild auf der Retina entsteht. Bei *torsionalen Augenbewegungen* handelt es sich um reflektorische Augenbewegungen, die durch die Bewegungen des Kopfes ausgelöst werden. Sie dienen zum Ausgleich von Rotationen des Kopfes. Fixationen und Sakkaden stellen die häufigsten Blickbewegungen dar.

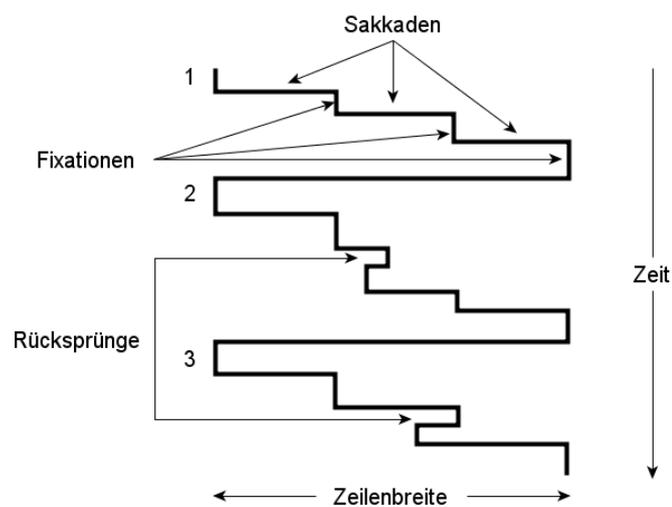


Abbildung 2.2: Blickbewegungen beim Lesen von drei Zeilen Text (eigene Darstellung in Anlehnung an [Gus96, Kapitel 4.1.2])

Beim Lesen ändern sich die Fixationen in kleine Sakkaden entsprechend der Leserichtung. Die Länge der Sprünge hängt dabei von der Größe des Textes und vom Verständnis ab. Rücksprünge können bei Texten auftreten, die schwer zu lesen sind. Ein Zeilenwechsel erfolgt oft über eine einzige Sakkade. Abbildung 2.2 zeigt die schematische Darstellung von Sakkaden und Fixationen beim Lesen von Text [Gus96].

Periphere und foveale Wahrnehmung

Der Mensch nimmt seine Umgebung durch Fixationen wahr, die durch Sakkaden und Regressionen unterbrochen werden. Dabei werden auch grobe Strukturen außerhalb der Fixation wahrgenommen, nämlich im peripheren Sichtfeld. Das menschliche Auge entspricht somit einem „Doppelsystem“, da durch foveales Sehen eine maximale Sehschärfe erreicht

wird und im peripheren Sichtfeld die Stimuli unscharf und verzerrt aufgenommen werden. Der Mensch ist somit in der Lage, nicht nur direkt fixierte Punkte wahrzunehmen, sondern auch grobe Strukturen außerhalb einer Fixation. Im peripheren Sichtfeld reagiert die menschliche Wahrnehmung sehr empfindlich auf Bewegungen und Veränderungen. Das zeitliche Auflösungsvermögen ist in der Peripherie am höchsten. Das bedeutet, schnelle Bewegungen im peripheren Sichtfeld führen dazu, dass die Aufmerksamkeit dorthin gelenkt wird und somit Fixationen neben den foveal einfallenden Stimuli auch durch sakkadische Zielsprünge zu Fixationsorten im peripheren Blickfeld gesteuert werden. Bei der Nutzung von Webseiten führt dies jedoch zum sogenannten *Vampir-Effekt*, da die Nutzer durch Animationen von relevanten Informationen abgelenkt werden [Nau12].

2.3 Geschichte

Die Untersuchung von Augenbewegungen ist schon seit dem 19. Jahrhundert bekannt. Der Franzose Émile Javal [Jav78] war einer der Ersten, der die Augenbewegungen beim Lesen direkt beobachtete. Seit dieser Zeit wurden einige unterschiedliche Methoden entwickelt. Es entstanden erste mechanische Geräte zur Aufzeichnung von Blicken, die anfänglich sehr unangenehm waren, da man sich eher invasiver Methoden bediente, die direkten Kontakt mit der Hornhaut hatten. Dodge und Cline [DCo1] entwickelten 1901 eine erste berührungsfreie Technik, die Reflexionen der Hornhaut auf fotografischen Platten speicherte. Im Jahr 1905 nutzten Judd et al. [JMS05] Filmaufnahmen, um den zeitlichen Verlauf der Augenbewegungen zu analysieren. In den 50er Jahren hat Fitts [FJL50] mit seinen Kollegen die ersten Blickpfade von Piloten während des Landeanflugs aufgenommen. Sie führten die erste Benutzerstudie - bekannt als *Usability-Engineering* - ein. Das Ziel dieser Studie war es, die Interaktion mit Produkten und ihr Produkt-Design zu verbessern. Später entdeckte Yarbus [Yar67] zudem, dass die Augenbewegungen beim Betrachten von Bildern je nach Aufgabenstellung unterschiedliche Verläufe aufweisen. Im Laufe der 70er Jahre lag der Schwerpunkt auf der Verbindung der Augenbewegungen mit der Wahrnehmung, indem die Blickpfade mit psychologischen Theorien verknüpft wurden. Weiterhin konnten zu dieser Zeit durch den technischen Fortschritt Eyetracking-Daten erstmals in Echtzeit genutzt werden, was zuvor nur in Retrospektive erfolgte. Außerdem wurde die Genauigkeit der ermittelten Daten erhöht, da erkannt wurde, dass durch Reflexionen von Licht auf der Hornhaut die Richtung des Blickes berechnet werden kann und somit auch die Augenbewegungen, unabhängig von der Kopfbewegung, analysiert werden können. In den 80er Jahren kam zum ersten Mal der Gedanke, die Augenbewegungen zur Interaktion zwischen Mensch und Maschine zu nutzen. Heutzutage ist Eyetracking eine unterstützende Technologie im Bereich der Mensch-Computer-Interaktion und dient hauptsächlich dazu, die Usability von Benutzerschnittstellen zu validieren [HRDo3].

Nicht nur das Eyetracking selbst hat sich weiterentwickelt, sondern auch die Techniken, welche im nächsten Abschnitt erläutert werden.

2.4 Eyetracking-Techniken

In diesem Abschnitt werden vier unterschiedliche Technologien für das Eyetracking vorgestellt, um zu zeigen, wie Augenbewegungen registriert und Eyetracking-Daten erfasst werden (vgl. [Duc07] und [Nau12]).

Elektro-Okulographie

Bei der Elektro-Okulographie werden mehrere Elektroden um das Auge herum platziert, die eine elektrische Spannung zwischen Hornhaut und Netzhaut messen. Die Elektroden können dabei die Blickrichtung der Augen feststellen, da sich die Netzhaut der einen Elektrode und die Hornhaut der gegenüberliegenden Elektrode annähert, wenn sich die Augenposition ändert. Das bedeutet, die Elektroden weisen Unterschiede im elektrischen Potenzial auf, sodass sich mit Hilfe der Spannungsdifferenzen die Richtung des Blickes berechnen lässt. Dabei ist zu beachten, dass die Augenbewegungen relativ zur Position des Kopfes gemessen werden, sodass auch die Kopfbewegung betrachtet werden muss. Vorteil dieser Methode ist, dass sich die Blickrichtungen bzw. die Augenbewegungen sogar mit geschlossenen Augen messen lassen. Diese Methode wird deshalb häufig in der Schlafforschung eingesetzt. Nachteil dieser Methode ist jedoch, dass die Ergebnisse relativ unpräzise sind.

Magneto-Okulographie

Bei der Magneto-Okulographie lassen sich die Augenbewegungen durch das Ändern von Magnetfeldern erkennen. Man unterscheidet hierbei zwischen zwei Techniken:

- Sclerale Kontaktlinsen-Methode
- Search-Coil-Methode

Bei der ersten Methode werden Kontaktlinsen eingesetzt, die verspiegelt sind, wodurch das einfallende Licht reflektiert und über Kameras aufgezeichnet wird. Bei der zweiten Methode werden um den Probanden herum Spulen aufgebaut, sodass Magnetfelder erzeugt werden. Die Probanden tragen hier ebenfalls Kontaktlinsen, die aber mit Sensoren und Spulen versehen sind. Sobald sich die Augenposition ändert, entstehen Magnetfeldveränderungen. Dadurch können die Augenbewegungen analysiert werden. Beide Methoden sind sehr unangenehm und werden nur unter ärztlicher Aufsicht durchgeführt.

Infrarot-Okulographie

Bei der Infrarot-Okulographie, auch *Cornea-Reflex-Methode* genannt, wird das Auge mit Infrarotlicht beleuchtet. Da Infrarot vom menschlichen Auge nicht erfasst werden kann, wird das Licht von der Hornhaut reflektiert und vom Glaskörper absorbiert. Das hat zur Folge, dass kein Infrarotlicht auf die Netzhaut trifft. Durch die Reflexion des infraroten

Lichts entsteht ein kleiner heller Punkt unter- oder innerhalb der Pupille und kann von einer Kamera wahrgenommen werden. Diese Reflexion liegt etwa 3,5 Millimeter unterhalb der Augenoberfläche. Wenn sich das Auge bewegt, ändern sich die reflektierenden Lichtstrahlen. Aus dem Winkel zwischen Pupille und der Reflexion kann die Blickrichtung exakt bestimmt werden.

Video-Okulographie

Die Funktionsweise der Video-Okulographie ähnelt der Infrarot-Okulographie, außer dass kein Infrarotlicht und somit auch kein Cornea-Reflex zum Einsatz kommt. Die Augenbewegungen werden ausschließlich von einer Kamera erfasst. Die Stellung der Augen wird direkt durch die Bildverarbeitung bestimmt. Moderne Systeme kombinieren jedoch beide Verfahren, um genauere Ergebnisse zu erhalten.

Bei der Untersuchung von Augenbewegungen werden meistens videobasierte Eyetracker eingesetzt, die im nächsten Abschnitt näher betrachtet werden.

2.4.1 Videobasierte Eyetracker

Bei videobasierten Eyetracker können generell zwei Arten von Aufzeichnungsgeräten unterschieden werden (vgl. [Rey09] und [Nau12]).

Überkopfsysteme

Überkopfsysteme, auch *Head-Mounted Eyetracker* genannt, stellen kopfgetragene Systeme dar und zeichnen die Augenbewegungen der Probanden auf. Durch eine Szenekamera, die im vorderen Bereich des Geräts integriert ist, wird zusätzlich das Sichtfeld der Versuchsperson aufgenommen. Der Proband kann sich somit frei bewegen und die Blickpfade können gleichzeitig aufgezeichnet werden. Um die Eyetracking-Daten speichern zu können, werden die Überkopfsysteme oft mit einem Notebook verbunden. Vorteil solcher Systeme ist die Mobilität, sodass Untersuchungen und Studien auch außerhalb von Laboren durchgeführt werden können. Nachteil solcher Systeme ist, dass eine statistische Auswertung der gewonnenen Eyetracking-Daten aufwendig ist. Der Grund hierfür ist, dass eine Parametrisierung der Daten aufgrund der Videoaufzeichnung nicht möglich ist und deshalb die Analyse der Aufzeichnungen manuell erfolgt.

Ferngesteuerte Systeme

Bei ferngesteuerten Systemen, auch *Remote Eyetracker* genannt, werden die Probanden mit dem Eyetracker nicht physisch verbunden. Die Aufzeichnung der Augen findet also berührungsfrei statt. Dafür werden meistens Infrarotkameras verwendet, die im Monitor des Computers integriert sind. Ferngesteuerte Systeme müssen zu Beginn kalibriert werden.



Abbildung 2.3: In dieser Arbeit eingesetzter ferngesteuerter Eyetracker Tobii T60 XL ^a

^a<http://www.tobii.com/>

Anschließend werden die Augenbewegungen aufgenommen und die Probanden können für einen bestimmten Bereich den Kopf frei bewegen. Vorteil dieser Systeme ist, dass im Gegensatz zu den Überkopfsystemen die Eyetracking-Daten für eine statistische Auswertung eindeutig parametrisiert werden können. Mit solchen Systemen kann genau gezeigt werden, wo und wann ein Proband den Bildschirm fixiert hat.

Abbildung 2.3 zeigt das ferngesteuerte Eyetracking System Tobii T60 XL, das in dieser Arbeit eingesetzt wurde. Dieser Eyetracker enthält zwei Infrarot-LEDs im unteren Bereich des Monitors. Die Augenbewegungen werden von einer Videokamera aufgezeichnet und mit Hilfe der Infrarot-LEDs wird die Position der Pupillen erkannt. Wie ein ferngesteuerter Eyetracker genau funktioniert, wird im nächsten Abschnitt erläutert.

2.4.2 Funktionsweise eines ferngesteuerten Eyetrackers

Im Rahmen dieser Arbeit kommt ein ferngesteuerter Eyetracker Tobii T60 XL zum Einsatz. Aufgrund dessen wird kurz auf die Funktionsweise eines ferngesteuerten Eyetrackers eingegangen.

Die Infrarot-LEDs, die im Monitor integriert sind, dienen als Lichtquelle und erzeugen Reflexionen auf der Hornhaut, was bereits bei der Infrarot-Okulographie erläutert wurde. Durch ein oder mehrere Cornea-Reflexionen und mit Hilfe der Koordinaten des Zentrums der Pupille kann die Augenposition im Video aufgenommen werden. Für die Bestimmung der Fixationen auf dem Bildschirm, ist eine Messung der Augen-Parameter und System-Parameter notwendig. Die Augenparameter ändern sich von Proband zu Proband und müssen deshalb mit Hilfe einer Kalibrierung bei jeder Untersuchung gemessen werden. Eine Kalibrierung wird durchgeführt, damit bestimmte Parameter bei der Blickpunktberechnung

das Ergebnis nicht verfälschen. Dabei wird beispielsweise der Abstand zwischen dem Zentrum der Pupillen jeder Person und die relative Position des Kopfes zum Monitor berechnet. Die System-Parameter werden nur einmal zu Beginn bestimmt. Dazu gehören beispielsweise die Parameter der Kamera und die Position der Infrarot-LEDs [GE06].

Während der Kalibrierung erscheinen auf dem Bildschirm Punkte, die vom Probanden verfolgt werden sollen. Dabei werden für jeden fixierten Punkt etwa 100 Schätzungen der Positionen der Reflexionen und Pupillenzentren gemacht. Anschließend werden für diese Positionen die Durchschnittskordinaten berechnet. Die Durchschnittskordinaten werden danach auf die Bilder der Kamera übertragen und in Weltkordinaten transformiert. Mit Hilfe von mathematischen Algorithmen lassen sich anschließend die Blickpunkte berechnen [GE06].

2.5 Anwendungsgebiete

Das Eyetracking hat sich in den letzten Jahren so weit entwickelt, dass es in verschiedenen Gebieten Anwendung findet. Anfänglich wurde es meistens in der Usability, im Marketing und in der Medizin genutzt. Mittlerweile hat das Eyetracking auch im Gebiet der Mensch-Computer-Interaktion an Bedeutung gewonnen. Im Folgenden werden unterschiedliche Anwendungsgebiete von Eyetracking vorgestellt.

Einsatz in der Neuropsychologie

In der Neuropsychologie verwenden Neurowissenschaftler Eyetracking, um den Zusammenhang zwischen physiologischen mit kognitiven und emotionalen Prozessen sowie der Informationsverarbeitung im Gehirn zu untersuchen. Es wird also versucht, die Funktionsweise des Auges zu erforschen und ein Zusammenhang zwischen Augenbewegung und Bildwahrnehmung festzustellen. Dem Menschen ist es möglich, einen Punkt zu fixieren ohne ihn wahrzunehmen. Stattdessen können in derselben Zeit andere Objekte wahrgenommen werden [Nau12].

Einsatz in der Medizin

Mit Hilfe des Eyetrackers werden in diesem Gebiet verschiedene Krankheiten oder Störungen erkannt und nach psychiatrischer, neurologischer oder ophthalmologischer Art differenziert. In der Ophthalmologie wird Eyetracking genutzt, um okulomotorisches Verhalten und Sehschwächen zu analysieren, sodass Informationen über Sakkaden und Reaktionen des peripheren Sehens durch blinkende Objekte untersucht werden können. Anhand der Blickbewegungen können dadurch beispielsweise Strabismus¹ festgestellt werden [Nau12].

¹Unkontrolliertes Innen- oder Außenschielen

Einsatz im Marketing

Auf diesem Anwendungsgebiet wird mit Hilfe von Eyetracking-Studien versucht, die psychologischen Auswirkungen unterschiedlicher Medienformate und Werbebotschaften von Web-Shops zu messen. Beispielsweise wird analysiert, ob ein Nutzer das Firmenlogo oder die Werbebotschaft wahrnimmt. Durch die Augenbewegungen können visuelle Elemente mit höchster oder geringster Aufmerksamkeit erkannt werden. Dadurch kann festgelegt werden, ob Elemente mit einer geringeren Aufmerksamkeit beibehalten oder an einer anderen Stelle platziert werden sollen. Wichtige Inhalte können bzw. sollten an den Stellen platziert werden, bei denen die Aufmerksamkeit hoch ist [Nau12].

Einsatz in der Usability

In der Usability werden Benutzerstudien vor allem für die Benutzerfreundlichkeit von Anwendersoftware genutzt, da mit Hilfe eines Eyetrackers die Aufmerksamkeitsverteilung der Probanden auf der Benutzeroberfläche gezeigt werden kann. Dabei wird versucht, Schwachstellen der Seiten durch Blickbewegungen der Probanden aufzudecken und zu verbessern. Außerdem kann festgestellt werden, ob die Seite intuitiv gestaltet ist oder nicht.

Das Optimieren von Anwendersoftware mittels Eyetracking wird in drei Bereiche untergliedert (vgl. [Nau12]):

- **Orientierung (Spontaneous Looking)**

Bei dieser Optimierung handelt es sich um die Verbesserung der Layout-Entwürfe. Dabei werden die erste Blicke der Nutzer auf der Seite betrachtet, da sie zu Beginn unwillkürlich einen Scanpfad festlegen und diesen bei späteren Aufrufen wieder folgen. Diese Phase nennt sich *Orientierungsphase*, da der Nutzer sich einen Überblick über die Software verschafft. Durch gelernte Schemata und Gestaltungselemente wird dabei die Festlegung des Scanpfads beeinflusst. Diese Aspekte werden deshalb bei der Gestaltung des Layouts ebenfalls betrachtet, um dem Nutzer eine intuitive Orientierung zu gewährleisten. Je besser das Layout, desto kürzer der Scanpfad.

- **Informationssuche (Task-Oriented Looking)**

In diesem Bereich werden sämtliche Aktivitäten der Probanden aufgezeichnet und analysiert. Dabei werden den Probanden bestimmte Aufgaben („*Use Cases*“) vorgegeben. Während des Lösen der Aufgaben werden Blickverläufe, Scrolling-Aktivitäten und Klicks parallel aufgenommen, wodurch ein ganzheitliches Bild von Such- und Aufmerksamkeitsprozessen gewonnen werden kann. Dadurch kann beispielsweise erkannt werden, an welchen Stellen die Probanden Probleme hatten, da Informationen an den falschen Stellen erwartet oder auch ganz übersehen wurden.

- **Informationsverarbeitung (Information-Oriented Looking)**

Bei der Informationsverarbeitung geht es um die Verbesserung der Content-Usability. Damit ist der Aufbau der Inhalte der Software gemeint. Durch die Augenbewegungen werden wichtige Daten für die Optimierung von Inhalten geliefert. Um eine hohe Nutzerfreundlichkeit zu erreichen, sollten Inhalte ästhetisch, intuitiv erfassbar und leicht lesbar sein.

Einsatz in der Mensch-Computer-Interaktion

Das Eyetracking spielt bei der Kommunikation zwischen Mensch und Maschine eine immer wichtigere Rolle. Dabei existieren in diesem relativ jungen Forschungsfeld hauptsächlich zwei Schwerpunkte: *Blickbedienung* und *Attentive User Interfaces (AUIs)* [Poi11]. Im Rahmen dieser Diplomarbeit wird das Eyetracking als Blickbedienung genutzt.

Im Bereich der Blickbedienung lag der Fokus zu Beginn vor allem darauf, motorisch eingeschränkten Nutzern den Zugang zu Computersystemen zu erleichtern. Die Blickposition wird dabei als Musersatz zur Interaktion genutzt. Mittlerweile liegt ein Schwerpunkt auch darauf, die Augen als ein weiteres Eingabemedium für nicht körperlich beeinträchtigte Menschen einzusetzen [Poi11].

Jacob [Jac90] hat bereits 1990 gezeigt, dass einige Funktionen der Maus durch Blickbewegungen ersetzt werden können. Seiner Ansicht nach, war es jedoch ein naiver Ansatz, die Augen zur direkten Manipulation zu ersetzen. Auf den ersten Blick scheint das Ersetzen der Maus durch Augenbewegungen eine gute Methode zu sein. In der Praxis ergibt sich jedoch das sogenannte *Midas Touch Problem*: In Anlehnung an den König Midas, der alles in Gold verwandelte, was er anfasste, bedeutet dies im übertragenen Sinne, dass die Nutzer alles anklicken würden, was sie anschauen. Die Nutzer sind es nicht gewohnt, Eingaben durch ihre Blicke zu tätigen. Objekte, wie beispielsweise Schaltflächen, werden oft betrachtet, ohne gleich mit ihnen interagieren zu wollen. Die Maus stellt dagegen zwei Funktionen zur Verfügung: Ein Objekt selektieren und dieses durch einen Mausklick initiieren. Mit dem Auge fällt die Funktion des Anklickens weg. Wenn beim einfachen Betrachten des Bildschirms ständig Aktionen ausgelöst werden, entsteht das Midas Touch Problem. Die einfachste Möglichkeit zur Vermeidung dieses Problems ist die Nutzung einer Verweildauer („*dwell time*“). Die Auslösung wird dabei verzögert oder abgebrochen, sobald der Nutzer seinen Fokus ändert. Bei der Selektion eines Objekts können hierbei verschiedene Latenzzeiten unterschieden werden. Diese einfache Methode hat jedoch Nachteile: Bei einer kurzen Verweildauer werden schnell falsche Selektionen getroffen. Bei einer langen Verweildauer wird der Nutzer ungeduldig und schweift mit seinem Blick vor der Selektion ab. Eine andere Möglichkeit zur Auslösung einer Interaktion, wie beispielsweise einem Mausklick, bietet die Kombination mit einem Tastendruck oder bei Menschen mit einer körperlichen Beeinträchtigung die Kombination mit einem Lidschlag eine Variante ohne Verweildauer [Nau12].

Im Bereich der Blickbedienung für motorisch eingeschränkte Menschen brachten Hornof und Cavender [HC05] im Jahr 2005 eine zweite Version der Software *EyeDraw* heraus, die

es Kindern mit einer körperlichen Beeinträchtigung ermöglicht, Bilder mit ihren Augenbewegungen zu zeichnen, damit sie ebenfalls kreative Erfahrungen sammeln können. Die Grundidee von EyeDraw ist, dass Blickpunkte, die als Mauscursor interpretiert werden, drei unterschiedliche Zustände haben können:

- Grün: Schauen
- Gelb: Bereitstellung zum Zeichnen
- Rot: Zeichnen

Durch diese Zustände wird das Midas Touch Problem gelöst und die Nutzer können selbst bestimmen, wann sie zeichnen möchten. Das Ändern der Zustände erfolgt durch die Unterscheidung zwischen Sakkaden und Fixationen: Solange der Nutzer seine Augen bewegt, bleibt der Zustand grün. Wenn ein Bereich für mehr als 500 Mikrosekunden fixiert wird, ändert sich der Zustand auf gelb. Wenn derselbe Bereich nochmals für über 500 Mikrosekunden fixiert wird, dann ändert sich der Zustand auf rot und der Nutzer kann zeichnen. Falls derselbe Bereich nicht ein zweites Mal für über 500 Mikrosekunden fixiert wird, dann wird der Zustand auf grün zurückgesetzt. Die Verweildauer kann in den Einstellungen auch geändert werden.

Attentive User Interfaces (AUIs) sind Anwendungen, welche die visuelle Aufmerksamkeit eines Benutzers auswerten und sich daran anpassen. In diesem Bereich ist das Eyetracking eine große Hilfe. AUIs interagieren mit den Nutzern und stellen ihnen Informationen bereit, indem sie durch die visuelle Aufmerksamkeit auf die Intention des Nutzers schließen. Konkret bedeutet dies, dass AUIs das Benutzerverhalten überwachen und Interessen oder Bedürfnisse des Nutzers angeben. In Kombination mit Eyetrackern gibt es eine Reihe an Anwendungsfeldern für AUIs. Ein Beispiel stellt *iDict* dar, das den Nutzern beim Lesen eines fremdsprachigen elektronischen Textes unterstützt. Dazu wird die Blickrichtung während des Lesens ständig analysiert. Ändert sich das Leseverhalten vom typischen Verhalten, erkennt das System, dass der Nutzer Probleme beim Lesen hat und übersetzt das Wort in die Muttersprache [Poi11].

3 Labeling

Dieses Kapitel befasst sich mit der Beschriftung von Objekten in der Kartografie, die zum Verständnis dieser Arbeit notwendig sind. Wichtige Grundregeln für die Platzierung von Beschriftungen werden erläutert. Anschließend wird auf die Beschriftung von Punktobjekten und auf unterschiedliche Problemstellungen beim Labeling eingegangen. Zudem werden verschiedene Klassen von Scatterplots vorgestellt.

3.1 Kriterien zur Platzierung von Labels

In der Geoinformatik und der automatisierten Textplatzierung ist das Einfügen von Text in Karten eine große Herausforderung. Die Positionierung der Beschriftung sollte klar und harmonisch zum Informationsobjekt sein. Überlappungen von Text und Informationsobjekten sollten vermieden werden, damit Beschriftungen lesbar und Assoziationen zwischen Informationsobjekten mit entsprechenden Beschriftungen eindeutig sind [YCL05].

Eduard Imhof [Imh62] veröffentlichte erstmals in seinem Aufsatz von 1962 wichtige Grundregeln für die Kartenschrift, die zeigen, wie Karten richtig zu beschriften sind und wie gute von schlechten Beschriftungen unterschieden werden können:

- Die Zuordnung zwischen Schrift und zugehörigem Informationsobjekt ist eindeutig
- Die Beschriftung überlappt keine anderen Objekte
- Die Beschriftung soll lesbar, unterscheidbar und leicht auffindbar sein
- Die Beschriftung soll nicht außerhalb der Karte erscheinen

Die zu beschriftenden Informationsobjekte lassen sich in *Punkt-*, *Linien-* und *Flächenobjekte* unterteilen. Dabei ist zu beachten, dass sich die Objektarten bei einem Maßstabswechsel ändern können.

Abbildung 3.1 zeigt sowohl ein Beispiel für eine schlechte, irreführende Zuordnung der Beschriftung als auch ein Beispiel für eine klare, eindeutige Zuordnung der Beschriftung. Sie verdeutlicht, dass eine räumliche Nähe zum Punktobjekt zwar notwendig ist, jedoch eine Nähe zu anderen Punktobjekten oder anderen Beschriftungen sich negativ auswirken kann. Wenn aber Beschriftungen andere grafische Objekte verdecken, reduziert sich der Informationsgehalt. Deshalb ist eine Überlappungsfreiheit der eindeutigen Zuordnung vorzuziehen, falls nicht beide Fälle eingehalten werden können. Eine Beschriftung kann dabei entweder ein Punktobjekt oder eine andere Beschriftung überdecken. Bei ersterem bewirkt

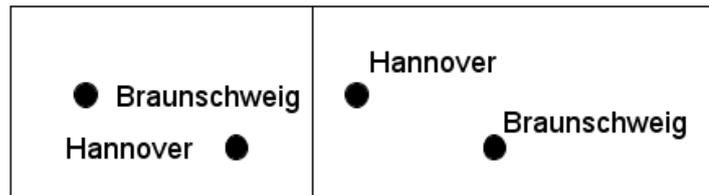


Abbildung 3.1: Irreführende Zuordnung der Beschriftung auf der linken Seite und eindeutige Zuordnung der Beschriftung auf der rechten Seite (Eigene Darstellung in Anlehnung an [Imh62, Seite 141])

das nicht nur eine schlechte Lesbarkeit, sondern führt auch zum Verlust der Information des Punktobjektes.

Eine gute Lesbarkeit der Beschriftungen hängt laut Imhof [Imh62], außer von der Position und der Anordnung von anderen Beschriftungen, auch von der Schriftart, der Schriftgröße und der Schriftfarbe ab. Dabei stellt eine größere Schrift eine höhere Priorität oder Quantität und eine kleinere Schrift eine niedrigere Priorität oder Quantität dar. Analog gelten diese Eigenschaften auch für fettgedruckte Schriftarten.

Der nächste Abschnitt befasst sich mit Punktobjekten, da Linien- und Flächenobjekte in dieser Arbeit nicht relevant sind.

3.2 Labeling von Punktobjekten

Laut Imhof [Imh62] werden Beschriftungen von Punktobjekten als Etiketten (Labels) entweder links, rechts, oben oder unten vom Punktobjekt angeordnet. Das bedeutet, alle Positionen um das Punktobjekt herum sind grundsätzlich möglich, haben aber unterschiedliche Prioritäten. Außerdem sollen Labels in Karten mit rechtwinkligen Koordinatenachsen waagrecht und Labels in Karten mit geografischen Koordinatennetzen parallel zu den Breitenkreisen angeordnet werden. Die Abbildung 3.2 zeigt, welche Positionen für Labels in einer Karte mit rechtwinkligen Koordinatenachsen bevorzugt werden. Idealerweise wird das Label rechts neben dem Punktobjekt etwas gehoben platziert. Falls rechts das Label sich mit einem anderen Label überlappt oder kein Platz mehr vorhanden ist, dann ist die Position links vom Objekt geeigneter als die „Überschrift“ oder „Unterschrift“. Wenn ein Label doch ober- oder unterhalb vom Punktobjekt platziert werden muss, ist die Überschrift der Unterschrift vorzuziehen. Das lateinische Alphabet, das mehr Kleinbuchstaben mit Oberlängen statt Unterlängen enthält, ist der Grund dafür, dass Überschriften weniger weggedrängt wirken als Unterschriften. Yoeli [Yoe72] hat in den 70er Jahren die bevorzugten Positionen der Labels verfeinert (siehe Abbildung 3.3).

Imhof [Imh62] beschreibt weiterhin, dass die Distanz zwischen einem Label und einem Punktobjekt von der Größe des Punktobjekts und von der Höhe des Labels abhängt. Die

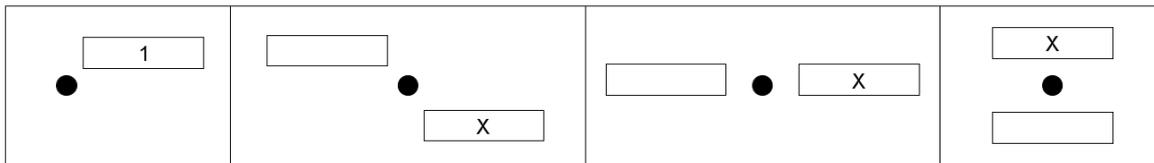


Abbildung 3.2: Imhof's bevorzugte Platzierung der Labels: 1 steht für eine ideale Platzierung; X zeigt die bevorzugte Platzierung (Eigene Darstellung in Anlehnung an [Hir82, Seite 6])

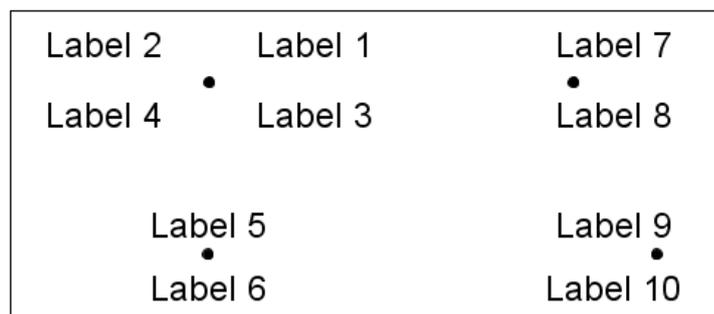


Abbildung 3.3: Yoeli's bevorzugte Positionen der Labels: Label 1 steht für eine bevorzugte Platzierung; Label 10 für die am wenigsten bevorzugte Platzierung (Eigene Darstellung in Anlehnung an [Hir82, Seite 6])

Distanz sollte zum einen nicht zu groß sein, da ansonsten die Zugehörigkeit zwischen Label und Punktopjekt nicht mehr eindeutig ist. Zum anderen sollte die Distanz nicht zu klein sein, da sonst der Abstand zu gedrängt wirkt und das Label schlechter lesbar ist.

3.3 Problemstellungen beim Labeling

Bei der Platzierung von Labels existieren verschiedene Problemstellungen, da es schwierig ist, allgemeine Kriterien oder Regeln zu gestalten. Im Folgenden werden verschiedene Problemstellungen vorgestellt, die sich auf Punktopjekte richten. Allerdings werden erst in Kapitel 4.1 verschiedene Ansätze vorgestellt, die das Problem der automatischen Platzierung von statischen Labels angehen.

Allgemein gilt für die vorgestellten Problemstellungen, dass ein Punktopjekt als x- und y-Koordinate auf einer Karte definiert wird, das beschriftet werden soll. Beschriftungen repräsentieren den Text eines Punktopjekts. Es wird dabei angenommen, dass alle Beschriftungen rechteckig und achsenparallel bzw. waagrecht sind und jedem Punktopjekt genau

eine Beschriftung zugeordnet wird. Jedes Punktobjekt hat entweder eine oder mehrere Positionen, auf die seine Beschriftung platziert werden kann.

Point Feature Label Placement Problem

Das diskrete *Point Feature Label Placement Problem (PFLP)* stellt eine Problemstellung bei der automatischen Platzierung von Labels dar. Beim PFLP wird angenommen, dass alle Labels gleich groß sind. Die Anzahl an Rechteckpositionen für jedes Label ist eine endliche Zahl. Dabei werden ein, zwei oder vier Rechteckpositionen gewählt (siehe Abbildung 3.4). Das *4-Positionen-Modell* entspricht den vier möglichen Rechteckpositionen für die Beschriftung eines Punktobjekts. Weiterhin wird davon ausgegangen, dass sich das Punktobjekt in der Nähe einer Ecke des Rechtecks des Labels befindet. Da es nicht immer möglich ist, alle Punktobjekte und Beschriftungen ohne Überlappung anzuordnen, wird beim PFLP versucht, die Anzahl an Überlappungen zu minimieren, indem für alle Beschriftungen genau die Rechteckpositionen gewählt werden, die nicht zur Überlappung mit anderen Beschriftungen oder Punktobjekten führen. Die Wahl aller Rechteckpositionen wird also so kombiniert, dass in der Gesamteinheit wenige Überdeckungen entstehen. Das PFLP funktioniert besonders gut, wenn wenige Punktobjekte dicht beieinander liegen [KM03], [IL99].

In anderen Fällen werden beim PFLP nur die Beschriftungen abgebildet, die sich mit keiner anderen überlappen, auch bekannt als *Beschriftungs-Anzahl-Maximierung* (genauere Details im nächsten Kapitel). Das bedeutet, falls dichte Bereiche mit Punktobjekten existieren, werden überlappende Beschriftungen nicht dargestellt [KM03]. Wenn beim PFLP zudem die Größe der Rechtecke für die Beschriftungen skaliert wird, dann handelt es sich um die nächste vorgestellte Problemstellung.

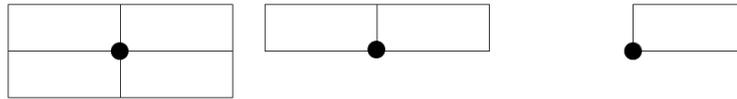


Abbildung 3.4: Festpositionenmodelle: vier-, zwei- und ein-Positionen-Modell (Eigene Darstellung in Anlehnung an [KM03, Seite 4])

Elastic Labeling Problem

Das *Elastic Labeling Problem* ist eine Alternative des PFLP. Hierbei werden Überlappungen nicht durch geeignete Rechteckposition vermieden, sondern durch Ändern der Form der Rechtecke, indem die Höhen und Breiten der Rechtecke variiert werden. Dabei wird angenommen, dass die Rechtecke der Labels einen fixen Bereich haben. Eine Ecke eines Rechteckes wird als Aufhängungspunkt genutzt, sodass trotz einer Formänderung eine konstante Position garantiert werden kann [IL99]. Abbildung 3.5 (a) veranschaulicht das.

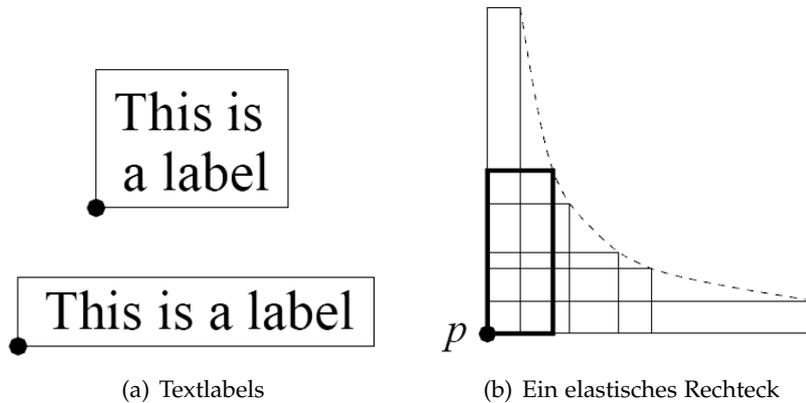


Abbildung 3.5: Mögliche Formen eines Rechtecks beim Elastic Labeling Problem (siehe [IL99, Seite 307 und 308])

Definition 3.1 Ein elastisches Rechteck ε ist eine Familie aus Rechtecken gegeben durch ein Quintupel (p, α, H, W, Q) , wobei nach [IL99] gilt:

- p ist ein gemeinsamer Punkt aus allen Rechtecke in ε
- α ist die Fläche jedes Rechtecks in ε
- $H = [h^{\min}, h^{\max}]$ ist die Spannweite der Höhen der Rechtecke
- $W = [w^{\min}, w^{\max}]$ ist die Spannweite der Breiten der Rechtecke
- $Q \subseteq \{1, 2, 3, 4\}$ ist eine Menge an möglichen Positionen von p , die in der Familie erlaubt sind. Der Wert 1 bedeutet, dass sich p in der unteren linken Ecke befindet, bei 2 in der oberen linken Ecke, bei 3 in der oberen rechten Ecke und bei 4 in der unteren rechten Ecke.

Abbildung 3.5 (b) zeigt ein elastisches Rechteck mit einer festen Anzahl an verschiedenen Formen. Punkt p stellt den Aufhängungspunkt aller möglichen Rechtecke aus der Familie ε dar. Da p sich in der unteren linken Ecke befindet, ist in diesem Beispiel $Q = 1$. Für jedes Element aus Q wird die Fläche α , in der die Familie der Rechtecke liegt, durch ein Segment der Hyperbel festgelegt, das sich Gegenüber von p befindet. In Abbildung 3.5 (b) wird ein Segment der Hyperbel für $Q = 1$ als gestrichelte Kurve gezeigt. Um ein einziges Rechteck aus der Familie zu erhalten, muss eine gültige Höhe H und Breite W des Rechtecks gewählt werden.

Das Elastic Labeling Problem kann mit dem PFLP kombiniert werden, sodass mehrere Positionen elastische Rechtecke benutzen, die wiederum eine bestimmte Anzahl an verschiedenen Formen eines elastischen Rechtecks verwenden.

Rectangle Perimeter Labeling Problem

Beim *Rectangle Perimeter Labeling Problem* werden alle Labels am Rand einer rechteckigen Fläche ohne Überlappungen positioniert. Die Punktobjekte liegen dabei auf dem Rand der Karte und jede Beschriftung wird einer Seite der rechteckigen Fläche zugeordnet. Das bedeutet, dass die Position der Labels von der Zuordnung der Seite abhängt. Um eine Überlappungsfreiheit zu ermöglichen, werden elastische Rechtecke benutzt [IL99] (siehe Abbildung 3.6).

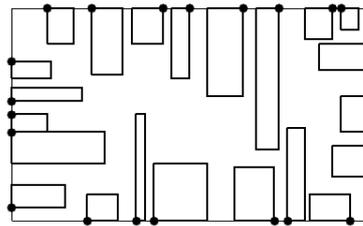


Abbildung 3.6: Lösung für ein Rectangle Perimeter Problem (siehe [IL99, Seite 307])

Sliding Labels Problem

Beim *Sliding Labels Problem* wird das PFLP verallgemeinert, da jede beliebige Position für ein Label erlaubt wird, sobald das Punktobjekt, das beschriftet werden soll, eine Kante des Labels berührt (siehe Abbildung 3.7). Somit sind unendlich viele Positionen möglich. Meist wird das 4-Schieber-Modell verwendet. In diesem Fall liegt das Punktobjekt irgendwo am Rand der Labelbox [KSW99]. Dieses Problem wurde bereits von Hirsch [Hir82] erwähnt. Bei überlappenden Labels werden Überlappungs- und Aggregatvektoren definiert, sodass die Labels um diese Vektoren entgegengesetzt verschoben werden. Dieses Prinzip stammt aus der Physik, dass gleiche magnetische Pole abstößt (genauere Details im nächsten Kapitel).

Es wurde gezeigt, dass Schieber-Modelle bei einer gegebenen Menge von n Punkten mindestens die Hälfte der Labels ohne Überlappungen darstellen, unabhängig davon, ob es sich um ein ein-, zwei- oder vier-Schieber-Modell handelt [EKW04].

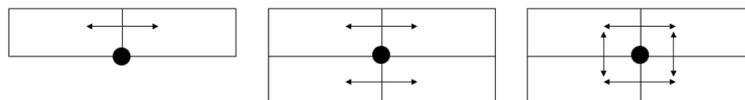


Abbildung 3.7: Ein-, zwei- und vier-Schieber-Modelle (Eigene Darstellung in Anlehnung an [KSW99, Seite 4])

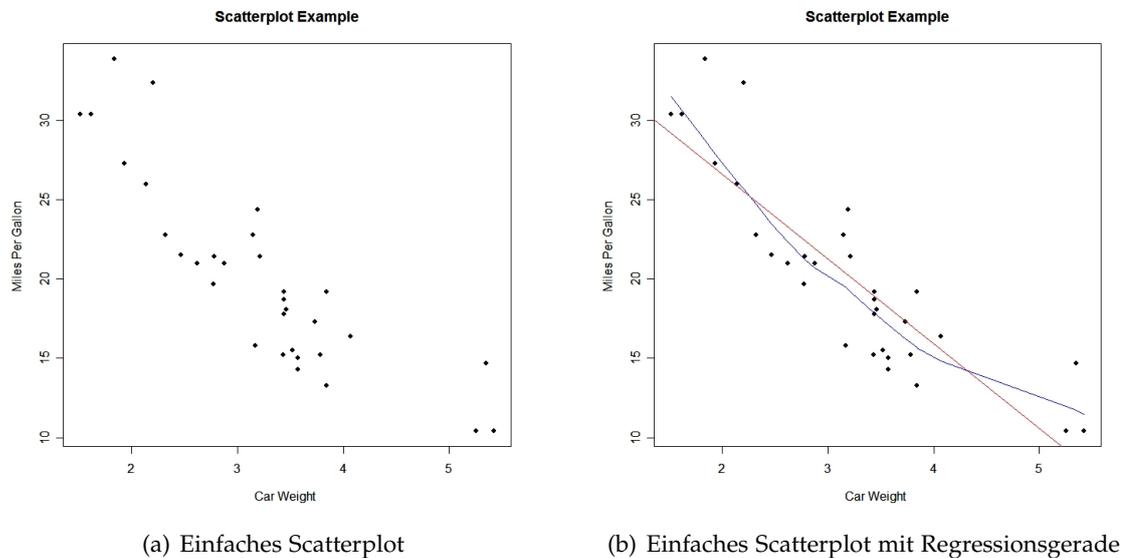
3.4 Scatterplots

In diesem Abschnitt werden verschiedene Klassen von Scatterplots (dt.: Streudiagramme) vorgestellt, die eine Möglichkeit für die Darstellung von Punktobjekten mit zugehörigen Labels bieten.

Ein Scatterplot ist eine grafische Darstellungsform, die eine gemeinsame Verteilung von Werten zweier oder dreier Variablen in Form von Punkten in einem kartesischen Koordinatensystem visualisiert [NBOH96]. Je nach Bearbeitungsziel lassen sich nach Schumann und Müller [SMoo] verschiedene Informationen über die Abhängigkeitsstruktur der Variablen erkennen, die sich in unterschiedlichen Problemstellungen unterteilen lassen:

Das *Identifikationsproblem* liefert die Werte von Punkten in einem bestimmten Bereich im Scatterplot zurück. Beim *Lokalisierungsproblem* werden die Punkte für bestimmte Werte der Variablen zurückgegeben. Das *Korrelationsproblem* zeigt Zusammenhänge zwischen den Variablen. Eine Korrelation stellt den linearen Zusammenhang zwischen zwei metrischen Variablen dar. Man unterscheidet zwischen einer positiven und einer negativen Korrelation. Eine positive Korrelation liegt vor, wenn eine positive lineare Beziehung zwischen den Variablen vorliegt. Das bedeutet, dass tendenziell hohe Werte der einen Variablen mit hohen Werten in der anderen Variablen auftreten. Eine negative Korrelation liegt vor, wenn eine negative lineare Beziehung zwischen den Variablen vorliegt. In diesem Fall treten tendenziell hohe Werte der einen Variablen mit niedrigen Werten in der anderen Variablen auf. Durch eine Regressionsgerade kann der lineare Zusammenhang einer Korrelation verdeutlicht werden. Diese Gerade liegt mitten im Verlauf der Punktwolke, sodass die Abstände zwischen der Geraden und den Punkten minimiert werden [NBOH96]. Beim *Gruppierungsproblem* werden Punkte mit gleichen Eigenschaften zu Clustern zusammengefasst. Bei einem Cluster handelt es sich um die Zusammenfassung von Punkten in möglichst homogenen Gruppen. Dabei werden Punkte mit ähnlichen bzw. korrelierten Variablenwerten einer Gruppe zugeordnet. Die Clusteranalyse ist für das Finden von Clustern und für die Bildung von homogenen Clustern zuständig [JLog]. Es soll also eine möglichst hohe Ähnlichkeit der Variablenwerte in den Gruppen und eine möglichst hohe Unähnlichkeit der Variablenwerte zwischen den Gruppen erreicht werden. Durch die Cluster entsteht somit eine heterogene Gesamtmenge mit homogenen Teilmengen an Punkten [BEPWoo]. Für die Gruppenbildungen existieren verschiedene Algorithmen, die allerdings nicht für diese Arbeit relevant sind. Das *Verteilungsproblem* sucht nach Extremwerten und Ausreißern und analysiert Muster der Punkte. Ein Ausreißer ist ein Punktobjekt, das auffällig weit von der Masse entfernt ist. Beim *Vergleichsproblem* wird analysiert, wie die Punkte sich in bestimmten Bereichen unterscheiden. Das *Häufigkeitsproblem* untersucht, welche Werte für Punkte besonders häufig auftreten und das *Kategorisierungsproblem* klassifiziert bestimmte Wertebereiche aufgrund von verschiedenen Eigenschaften.

Es existieren unterschiedliche Klassen von Scatterplots, die im Folgenden erläutert werden.



(a) Einfaches Scatterplot

(b) Einfaches Scatterplot mit Regressionsgerade

Abbildung 3.8: Grafische Darstellung von einfachen Scatterplots (siehe [Kab12])

Einfaches Scatterplot

Ein einfaches Scatterplot stellt die gemeinsame Verteilung zweier metrischer Variablen in einem zweidimensionalen Koordinatensystem dar. Dabei werden die Werte der einen Variable auf der Abszissenachse und die Werte der anderen Variable auf der Ordinatenachse abgetragen. Mit Hilfe der beiden Achsen kann anschließend die Lage der Wertepaare im Koordinatensystem durch ein Punkt oder ein anderes Symbol eingefügt werden. Das bedeutet, dass jeder Punkt ein Wertepaar beider Variablen repräsentiert [NBOH96]. Eventuell fallen mehrere Punkte auf dieselbe Position. Eine weitere kategorische Variable kann zusätzlich für die Punktbeschriftung integriert werden, um mit Hilfe dieser Information ein besseres Verständnis der Punkte zu erlangen oder wenn eine nicht-metrische Variable durch metrische Variablen repräsentiert werden soll. Die Punktbeschriftungen können entweder statisch oder dynamisch dargestellt werden. Im nächsten Kapitel wird vertieft auf die verschiedenen Methoden der Punktbeschriftung eingegangen. Abbildung 3.8 (a) stellt ein Beispiel für ein einfaches Scatterplot mit einer Korrelation dar. In diesem Beispiel ist eine negative Korrelation abgebildet. Denn je höher das Gewicht der Autos ist, desto niedriger ist die zurückgelegte Strecke in Meilen, für die eine Gallone Kraftstoff verbraucht. Abbildung 3.8 (b) verdeutlicht diese Korrelation mit Hilfe einer roten Regressionsgerade. Im Rahmen dieser Diplomarbeit werden einfache Scatterplots mit Punktbeschriftungen eingesetzt.

Einfaches Scatterplot in Matrix-Darstellung

Wenn mehr als zwei oder drei metrische Variablen betrachtet werden, dann kann eine Scatterplot-Matrix eingesetzt werden. Die Scatterplot-Matrix stellt mehrere einfache Scatter-

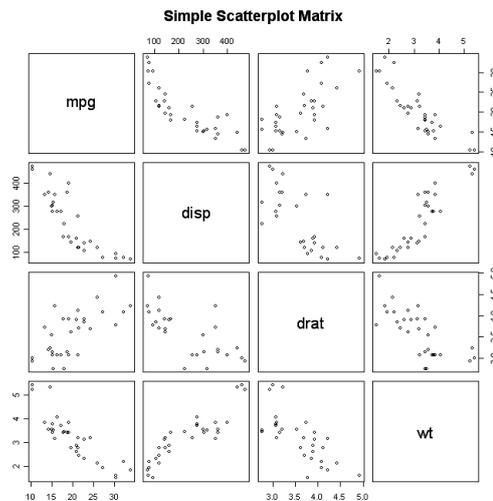


Abbildung 3.9: Grafische Darstellung einer Scatterplot-Matrix (siehe [Kab12])

plots für verschiedene Variablen in Form einer Matrix gegenüber, um sich einen Überblick über alle denkbaren Variablenpaare zu verschaffen. Jedes Scatterplot erscheint in der Matrix einmal oberhalb und einmal unterhalb der Hauptdiagonalen, wobei die Achsen auf den entsprechenden Seiten vertauscht sind. Jede Zeile enthält die Wertekombinationen einer Variablen mit allen anderen Variablen. Anstelle der Kombination mit sich selbst stehen in der Hauptdiagonalen die Bezeichnungen der entsprechenden Variablen. Insgesamt werden für p zu untersuchenden Variablen $p(p-1)/2$ Scatterplots jeweils zweimal symmetrisch erzeugt. Wie bei einfachen Scatterplots können auch hier Regressionsgeraden hinzugefügt werden [NBOH96]. Abbildung 3.9 zeigt ein Beispiel für eine Scatterplot-Matrix. Die Variablen stehen auf der Hauptdiagonalen der Matrix, wodurch alle Kombinationsmöglichkeiten zwischen den Variablen einen Scatterplot erzeugen. Die Anzahl an Variablen kann hier beliebig gewählt werden. Vorteil von Scatterplot-Matrizen ist, dass Korrelationen zwischen zwei Variablen gut erkennbar sind.

3D-Scatterplot

Wenn drei metrische Variablen betrachtet werden, dann kommt das 3D-Scatterplot zum Einsatz, das drei Variablen räumlich darstellt (siehe Abbildung 3.10). Auch hier kann eine weitere Variable zur Punktbeschriftung hinzugefügt werden. Für jede der drei Variablen wird eine Achse im dreidimensionalen Raum zugeordnet, sodass eine Punktwolke entsteht. In dieser Darstellung treten Zusammenhänge stärker hervor, wenn das Diagramm gedreht wird und somit von mehreren Seiten betrachtet wird [Mar03].

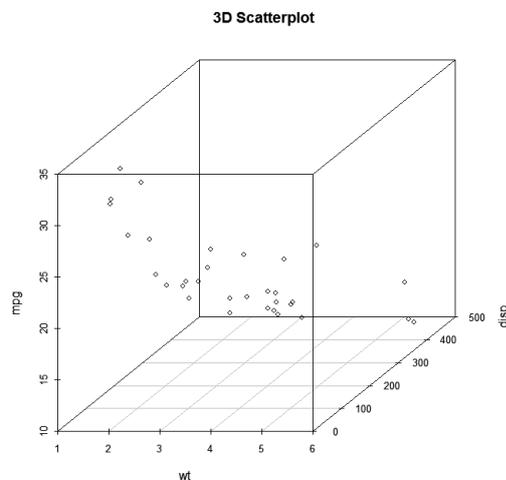


Abbildung 3.10: Grafische Darstellung eines 3D-Scatterplots (siehe [Kab12])

Vor- und Nachteile

Wesentliche Vorteile von Scatterplots sind die Veranschaulichung von Zusammenhängen und Unterschieden zwischen den Merkmalsvariablen. Scatterplots liefern eine hohe Anschaulichkeit der Daten. Dabei können große Mengen ohne hohen Aufwand analysiert werden. Durch die grafische Repräsentation mehrerer Merkmale können erste Trends der Datensätze erkannt werden. Scatterplots visualisieren Muster der Datenpunkte und ermöglichen Entdeckungen von auffälligen Punktwerten. Zudem können Ausreißer interpretiert werden.

Ein Nachteil von Scatterplots sind die sogenannten *Visual Clutter*, die dann auftreten, wenn zu viele Punkte auf zu engem Platz liegen. Dadurch können Verdeckungen von Punkten sowie Informationsverluste entstehen [FTSS09]. Um dieses Problem zu verringern, werden verschiedene Metriken verwendet. Scatterplots arbeiten beispielsweise mit den Metriken *Overplotting* oder *Visual Density*.

Das Overplotting beschäftigt sich mit mehreren aufeinander liegenden Punkten, da in diesem Fall die genaue Anzahl an Punkten nicht sichtbar ist. Das Overplotting geht mit Hilfe der Dichteschätzung dieses Problem an, indem die Luminanz geringer wird, je mehr Punkte aufeinander liegen. Über die Prozentangabe pro Pixel kann die Luminanz bestimmt werden. Beispielsweise steht die Prozentangabe 0 für keinen Overplot. Allgemein gilt, je höher die Prozentangabe, desto mehr Punkte liegen aufeinander [ED06]. In einem Scatterplot entspricht die Visual Density dem durchschnittlichen Verhältnis der Anzahl an Punkten innerhalb eines Clusters zur Cluster-Fläche. Dabei wird die konvexe Hülle des Clusters bestimmt und die Dichte durch verschiedene Farbstufen dargestellt [FTSS09].

4 Automatische Platzierung von Labels

In diesem Kapitel werden verschiedene Algorithmen und Techniken für eine automatische Platzierung von Labels vorgestellt, die in Scatterplots für eine statische und dynamische Beschriftung der Punktobjekte eingesetzt werden können. Zudem wird verglichen, welche Techniken für den Einsatz in Scatterplots und für die Umsetzung der drei Techniken geeigneter sind.

4.1 Statisches Labeling

Das statische Labeling unterscheidet sich vom dynamischen dadurch, dass die Positionierung der Labels nicht zeitabhängig ist. Das Ziel ist, eine optimale Platzierung der Labels festzulegen. Bei dieser Art von Textplatzierung handelt es sich um ein Optimierungsproblem, das zwei Aspekte betrachtet: Einen *Search Space* und eine *Objective Function* [CMS95].

Search Space

Ein Element des Search Spaces stellt eine Labeling-Funktion $f : P \rightarrow N$ dar, mit einer Menge an Punktobjekten $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ und einer Menge $N = \{n_1, \dots, n_i\} \subset \mathbb{N}$ an möglichen Beschriftungspositionen. Die Menge an potentiellen Beschriftungspositionen für jedes Punktobjekt charakterisiert somit den Search Space. Abbildung 4.1 zeigt eine typische Menge an acht möglichen Beschriftungspositionen für ein Punktobjekt. Jede Box entspricht dabei einer Region, auf die eine Beschriftung platziert werden kann. Alternativ kann auch eine kontinuierliche Platzierung genutzt werden, indem ein Kreis um das Punktobjekt herum spezifiziert wird, sodass die Beschriftungen diesen Kreis ohne Überschneidung berühren.

Objective Function

Die Objective Function repräsentiert die Qualität des Labelings, die von vielen Faktoren abhängt. Das Ziel der Objective Function ist, aus der Menge der möglichen Beschriftungspositionen genau die auszuwählen, die keine Überlappung erzeugt. Falls solch eine Auswahl nicht existiert, wird die Auswahl der Beschriftungspositionen so gewählt, dass möglichst wenige Überlappungen entstehen. Zur Vereinfachung wird angenommen, dass alle Beschriftungen dieselbe Größe haben. Für die Objective Function gilt:

• **Überlappung**

Die Anzahl an Überlappungen zwischen den Beschriftungen sowie zwischen Beschriftungen und Punktobjekten ist zu minimieren, das in [MS93] folgendermaßen definiert wird:

Gegeben ist eine Menge an Punktobjekten $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, eine Menge $N = \{n_1, \dots, n_i\} \subset \mathbb{N}$ an möglichen Beschriftungspositionen sowie die Kostenfunktionen $c_{pl} : P \times N \rightarrow \mathbb{Z}^+$ und $c_{ll} : P \times N \times P \times N \rightarrow \mathbb{Z}^+$ mit:

$$c_{pl}(p, n) = \begin{cases} 1, & \text{wenn ein Punkt } p' \in P \text{ existiert, sodass } p' \in \text{region}(p, n) \\ 0, & \text{sonst} \end{cases}$$

$$c_{ll}(p, n, p', n') = \begin{cases} 1, & \text{wenn } \text{region}(p, n) \cap \text{region}(p', n') \neq \emptyset \\ 0, & \text{sonst} \end{cases}$$

$\text{region}(p, n)$ gibt die überlappende Position n für das Punktobjekt p an. Folgende Frage soll beantwortet werden: Existiert eine Schriftplatzierung $g : P \rightarrow N$, welche die Kostenfunktion

$$\sum_{p \in P} c_{pl}(p, g(p)) + \sum_{p, q \in P} c_{ll}(p, g(p), p, g(q))$$

minimiert?

• **Beschriftungs-Anzahl-Maximierung**

In manchen Fällen werden nicht alle Punktobjekte beschriftet. In Abhängigkeit der Anzahl an Punktobjekten und Überlappungen wird das Auslassen von Beschriftungen bewertet. Unter dem Problem der *Beschriftungs-Anzahl-Maximierung (BAM)* versteht man nach [BHH⁺02]:

Zur Punktmenge $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ und gegebenen Menge B der Beschriftungskandidaten finde eine Teilmenge $L = l_1, \dots, l_m \subset B$, so dass

- $l_i \cap l_j = \emptyset \quad \forall i \neq j$
- jeder Punkt höchstens eine Beschriftung erhält und
- $|L|$ maximal ist

• **Bevorzugte Platzierung**

Die Positionen um das Punktobjekt herum können verschiedene Prioritäten haben. Das bedeutet, nicht alle Positionen werden gleich behandelt. Abbildung 4.1 zeigt verschiedene Prioritäten für ein Punktobjekt. Für eine Bewertung der Beschriftungspositionen werden Konstanten eingesetzt. Konstanten mit einem höheren Wert werden weniger bevorzugt als Konstanten mit einem niedrigeren Wert.

- **Distanz zwischen den Beschriftungen**

Um Visual Clutter zu reduzieren, sollten Beschriftungen, falls möglich, weit entfernt voneinander positioniert werden. Dabei wird überprüft, ob zwei nicht-überlappende Beschriftungen nahe beieinander liegen. Beschriftungen, die näher beieinander liegen, erhalten einen höheren Bewertungswert.

Objective Function bei linearer Programmierung

Bei der linearen Programmierung wird versucht, die Objective Function in Gleichungen auszudrücken. Zoraster [Zor86] definiert das Optimierungsproblem für das PFLP Problem folgendermaßen:

Gegeben sind k Beschriftungen und N_k potentielle Beschriftungspositionen für eine Beschriftung. Jede Beschriftungsposition wird durch eine Variable $X_{i,k}$ mit $1 \leq i \leq N_k$ und $1 \leq k \leq K$ repräsentiert mit:

$$X_{i,k} = \begin{cases} 1, & \text{Beschriftungsposition ist selektiert} \\ 0, & \text{Beschriftungsposition ist nicht selektiert} \end{cases}$$

Damit jedes Punktobjekt genau einmal beschriftet wird, muss für $1 \leq k \leq K$ folgende Bedingung erfüllt sein:

$$\sum_{i=1}^{N_k} X_{i,k} = 1$$

Gegeben sind Q paarweise Überlappungen mit $1 \leq q \leq Q$. Damit sich zwei Beschriftungspositionen r_q und r'_q für zwei Punktobjekte s_q und s'_q nicht überlappen, gilt folgende Bedingung:

$$X_{r_q, s_q} + X_{r'_q, s'_q} \leq 1$$

Die Bewertung der möglichen Positionen anhand ihrer Priorität erfolgt mit $W_{i,k} \geq 0$. Ein niedriger Wert stellt eine höhere Priorität dar. Das Textplatzierungsproblem lässt sich durch folgende Objective Function ausdrücken, die minimiert werden soll:

$$\sum_{i=1}^k \sum_{i=1}^{N_k} W_{i,k} X_{i,k}$$

Verschiedene Algorithmen können eingesetzt werden, um diese Summe zu minimieren.

Die vorgestellten Algorithmen gehen diesem Optimierungsproblem nach. Dabei wird die Objective Function entweder mit Hilfe von lokalen Minima oder durch eine vollständige globale Suche minimiert [CMS95]. Bei ersterem werden deutlich geringere Laufzeiten bei jedoch nicht optimalen Ergebnissen erreicht. Das lässt sich dadurch erklären, da nicht alle Kombinationen der Beschriftungspositionen getestet werden. Die Objekte werden nur lokal

optimal beschriftet. Solche Heuristiken sind nur dann geeignet, wenn wenig oder keine Überlappungen auftreten. Bei zweiterem werden zwar schlechtere Laufzeiten erreicht, dafür werden häufig optimale Ergebnisse geliefert. Grund hierfür ist, dass zwischenzeitlich auch Verschlechterungen akzeptiert werden.

4.1.1 Erschöpfende Suche

Bei der *erschöpfenden Suche* werden alle Positionsmöglichkeiten der Beschriftungen durchgegangen und bewertet. Die Kombination, die den kleinsten Wert für die Objektive Funktion liefert, wird anschließend ausgewählt.

Die Auswahl der Beschriftungsposition erfolgt durch eine vorgegebene Reihenfolge, die in Abbildung 4.1 beispielhaft gezeigt wird. Ein Rechteck entspricht dabei einem Bereich, in dem eine Beschriftung platziert werden kann. Falls ein Objekt nicht beschriftet werden kann, weil ansonsten eine Überlappung entsteht oder alle möglichen Positionen bereits ausprobiert wurden, dann kehrt der Algorithmus zum letzten beschrifteten Punkt zurück und betrachtet die nächste verfügbare Position. Der Algorithmus wiederholt diesen Schritt solange, bis eine überlappungsfreie Position gefunden wurde oder alle Beschriftungspositionen durchlaufen wurden [CMS95].

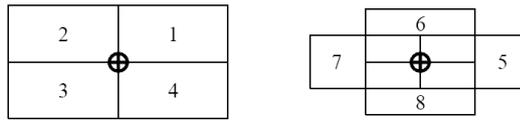


Abbildung 4.1: Mögliche Beschriftungspositionen und ihre Prioritäten: Niedrige Werte stellen bevorzugte Beschriftungspositionen dar (siehe [CMS95, Seite 204])

Ein Vorteil von diesem Algorithmus ist, dass dieser eine globale Suche durchführt. Ein Nachteil ist seine hohe Komplexität. Denn wenn zum Beispiel vier mögliche Beschriftungspositionen existieren und 50 Punkte beschriftet werden sollen, dann müssen 4^{50} Kombinationen getestet werden.

4.1.2 Discrete Gradient Descent

Beim *Discrete Gradient Descent Algorithmus* handelt es sich um einen Greedy Algorithmus, der versucht, einen Ausgleich zwischen Laufzeit und Beschriftungsqualität zu erzielen. Im Gegensatz zur erschöpfenden Suche wird beim Discrete Gradient Descent Algorithmus über lokale Minima ein globales Minimum gesucht. Algorithmus 4.1 auf der nächsten Seite zeigt den Discrete Gradient Descent Algorithmus [CMS95].

Der Algorithmus basiert darauf, eine Position für die Beschriftung zu wählen, die augenblicklich eine globale Verbesserung erzeugt. Die Verbesserung wird anhand der Objective Function berechnet.

Algorithmus 4.1 Discrete Gradient Descent

```

procedure DGD(Features, Mögliche Positionen des Features)
  for all Features do
    Platziere die Beschriftung zufällig auf eine mögliche Position des Features
    Berechne die aktuelle Objective Function
  end for
  repeat
    Teste alle alternativen Beschriftungspositionen des Features
    Berechne die neue Objective Function
    if Verbesserung then
      Platziere die Beschriftung an die Position mit der größten Verbesserung neu
    end if
  until Keine Verbesserung
end procedure

```

Der größte Nachteil dieser Methode ist, dass sie in lokale Minima verfällt. Abbildung 4.2 zeigt ein typisches Beispiel für ein lokales Minimum. Die grauen Flächen stellen jeweils die gewählte Position für ein Punktobjekt dar, in der die Beschriftung liegt. Die weißen Flächen zeigen die Positionen, die nicht gewählt wurden. In diesem Fall kann die Überlappung gelöst werden, wenn die Beschriftung des unteren Punktobjekts auf die untere linke Position und die Beschriftung des oberen Punktobjekts auf die obere rechte Position verschoben wird. Da der Algorithmus nur *eine* Beschriftung repositioniert, wird die Überlappung für alle vier möglichen Positionen nicht gelöst und der Wert der Objective Function ändert sich nicht. Vorteil des Discrete Gradient Descent Algorithmus ist, dass seine Laufzeit geringer ist als die der erschöpfenden Suche [CMS95].

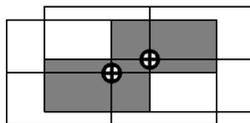


Abbildung 4.2: Darstellung eines lokalen Minimums für den Discrete Gradient Descent Algorithmus (siehe [CMS95, Seite 208])

4.1.3 Algorithmus von Hirsch

Der *Algorithmus von Hirsch* [Hir82] unterscheidet sich von den bisherigen Algorithmen dadurch, dass die Beschriftungen eines Punktobjekts nun unendlich viele Positionen annehmen können (siehe Sliding Labels Problem, Kapitel 3.3). Die einzige Bedingung ist, dass die achsenparallelen Beschriftungsboxen einen Kreis, um das Punktobjekt herum, tangieren müssen (siehe Abbildung 4.3 a). Wenn die Beschriftungsboxen entweder auf der höchsten,

der niedrigsten, der äußerst linken oder der äußerst rechten Position des Kreises liegen, dann befinden sie sich in einer sogenannten *Special Zone*. In diesen Zonen dürfen die Beschriftungsboxen auch nach oben und nach unten bzw. nach rechts und nach links verschoben werden (siehe Abbildung 4.3 b).

Zu Beginn befinden sich alle Beschriftungen auf der äußerst rechten Position. Anschließend wird für jede Beschriftung getestet, ob Überlappungen mit anderen Beschriftungen oder Punktobjekten existieren. Falls es zu Überlappungen kommt, werden *Überlappungsvektoren* gebildet. Diese Vektoren repräsentieren die Euklidische Distanz, die benötigt wird, um eine Überlappung zu lösen. Falls eine Beschriftung mehrere Überlappungen enthält, wird ein *Aggregatvektor* gebildet. Dieser stellt die Summe der Überlappungsvektoren innerhalb einer Beschriftung dar. Das Prinzip des Algorithmus stammt aus der Physik. Die Überlappungsvektoren entsprechen magnetischen Polen, die sich gegenseitig abstoßen. In Abbildung 4.3 c sind die Überlappungsvektoren grau und die Aggregatvektoren schwarz gefärbt. Falls die Beschriftungen nur in einem Konflikt involviert sind, ist der Überlappungsvektor gleich dem Aggregatvektor.

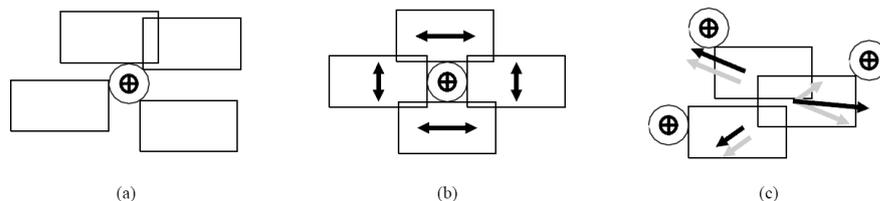


Abbildung 4.3: Potentielle Beschriftungspositionen (siehe [CMS95, Seite 209])

Nachdem die Überlappungs- und Aggregatvektoren für alle Beschriftungen innerhalb eines Konflikts berechnet wurden, werden sie um diese Vektoren verschoben, um die Anzahl an Überlappungen zu reduzieren. Man unterscheidet dabei zwischen zwei Verschiebungsmethoden:

- Inkrementelle Verschiebung
- Absolute Verschiebung

Die *inkrementelle Verschiebung* nutzt mehrere heuristische Regeln, um eine Beschriftung zu verschieben. Wenn sich zum Beispiel eine Beschriftung in einer *Special Zone* befindet, werden die x und y Komponenten der Vektoren genutzt. Das bedeutet, wenn sich die Beschriftung beispielsweise äußerst links oder äußerst rechts befindet, wird nur die y Komponente betrachtet. In diesem Fall wird die Beschriftungsbox in Abhängigkeit der y Komponente entweder nach unten oder nach oben geschoben. Die *absolute Verschiebung* platziert die Beschriftungen direkt auf die Positionen, die von den Überlappungs- bzw. Aggregatvektoren vorgegeben werden. Wenn beispielsweise der Überlappungsvektor einen Winkel von 110 Grad hat, wird die Beschriftung exakt auf die Position des Kreises platziert. Dabei spielt die aktuelle Position der Beschriftung keine Rolle. Nach Hirsch [Hir82] sollten beide Methoden abwechselnd eingesetzt werden.

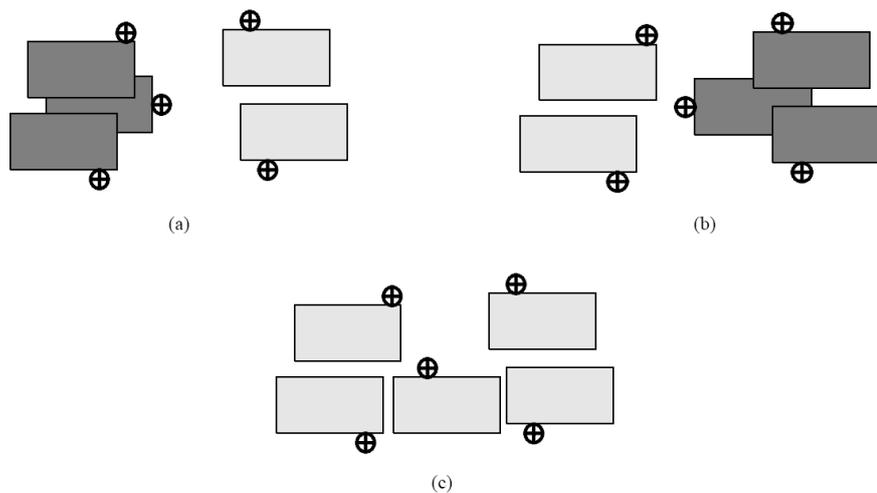


Abbildung 4.4: Algorithmus von Hirsch: Ein lokales Minimum mit inkrementeller (a), absoluter (b) und gewünschter (c) Verschiebung (siehe [CMS95, Seite 210])

Nachteil des Algorithmus ist, dass durch die Überlappungs- bzw. Aggregatvektoren nicht immer eine Verbesserung erzeugt wird. Das Problem liegt an der Addition der Überlappungsvektoren. Wenn mehrere Beschriftungen eine einzige Beschriftung überlappen, wird oft ein zu langer Aggregatvektor gebildet. Für die inkrementelle Methode sind zu lange Vektoren eher problematisch. Außerdem verfügt jeder Überlappungsvektor über zwei Freiheitsgrade, obwohl sie eigentlich den Kreis ihres Punktoobjekts tangieren müssen. Weiterhin verfällt dieser Algorithmus, wie auch der Discrete Gradient Descent Algorithmus, in lokale Minima. Abbildung 4.4 zeigt dafür eine problematische Konfiguration. Falls eine inkrementelle Verschiebung durchgeführt wird (siehe Abbildung 4.4 a), wird die Beschriftung etwas nach unten und nach oben verschoben. In diesem Fall wird damit jedoch der Konflikt nicht gelöst. Falls eine absolute Verschiebung durchgeführt wird (siehe Abbildung 4.4 b), erzeugt der Aggregatvektor eine Position auf der gegenüberliegenden Seite. In diesem Fall wird wieder aufgrund der anderen Beschriftungen der Konflikt nicht gelöst und die Beschriftung kann sich nicht zwischen der linken und rechten Seite entscheiden. Ein weiterer Nachteil sind die virtuellen Kreise um die Punktoobjekte herum. Denn je größer der Radius des Kreises ist, desto mehr Platz wird verschwendet. Dadurch ist die Wahrscheinlichkeit höher, dass sich Beschriftungen überlappen. Dieses Problem lässt sich aber schnell lösen, indem der Radius einfach auf Null gesetzt wird [CMS95].

Vorteil des Algorithmus ist, dass im Gegensatz zu Algorithmen, die feste Positionen für ihre Beschriftung verwenden, der Raum für die Platzierung vollständig ausgenutzt wird [CMS95].

4.1.4 Simulated Annealing

Simulated Annealing ist ein stochastischer Optimierungsalgorithmus, der durch die Simulation natürlicher Vorgänge versucht, ein optimales Arrangement der Beschriftungen zu erreichen. Die Grundidee des Algorithmus stammt aus der Physik, da das langsame Abkühlen einer Molekülmenge auf den Algorithmus projiziert wird. Voraussetzung für die Anwendung von Simulated Annealing ist, dass Festpositionenmodelle als Basis für den Algorithmus verwendet werden, wie etwa das 4-Positionen-Modell [BHH⁺02].

Der Algorithmus läuft folgendermaßen ab: Zu Beginn werden alle Beschriftungen zufällig auf eine mögliche Position platziert. Anschließend wird ein beliebiges Punktobjekt aus der Gesamtmenge herausgegriffen und in einer zufälligen Position neu beschriftet. Falls die neue Position eine Verbesserung zur alten aufweist, wird diese sofort übernommen. Die Verbesserung bzw. Verschlechterung wird mit Hilfe der Objective Function bestimmt. Sinkt die Objective Function, so tritt eine Verbesserung ein. Steigt sie oder bleibt konstant, so tritt eine Verschlechterung ein. Wenn die neue Position eine Verschlechterung aufweist, wird die neue Position nur mit einer bestimmten Wahrscheinlichkeit übernommen. Diese Wahrscheinlichkeit hängt von der Temperatur ab, die eine Kontrollgröße darstellt. Mit abnehmender Temperatur nimmt auch die Wahrscheinlichkeit ab. Das bedeutet, dass zu Beginn Überlappungen noch mit einer höheren Wahrscheinlichkeit zugelassen werden. Mit sinkender Temperatur nimmt dies jedoch ab. Ab einem gewissen Zeitpunkt werden keine Verschlechterungen mehr zugelassen. Wenn die Temperatur so gering ist, dass sich die Position der Beschriftungen nicht mehr ändert, bricht der Algorithmus ab [CMS95].

Die Intensität der Temperaturabkühlung kann die Endqualität und die Laufzeit beeinflussen. Je stärker die Abkühlung, desto kürzer ist die Laufzeit, aber desto größer ist die Wahrscheinlichkeit, ein schlechtes Endergebnis zu erhalten [BHH⁺02].

Ein großer Vorteil von Simulated Annealing ist, dass dieser Algorithmus nicht so schnell in lokale Minima verfällt, da er zwischenzeitlich auch Verschlechterungen zulässt. Simulated Annealing ist also in der Lage, ein globales Minimum zu finden. Aufgrund dessen wurde in dieser Diplomarbeit Simulated Annealing für die statische Beschriftung umgesetzt. Abbildung 4.5 zeigt deutlich die Vorteile von Simulated Annealing in der Qualität gegenüber den anderen vorgestellten Algorithmen. In dieser Abbildung wird beispielhaft die Beschriftung von 750 Punktobjekten mit den vorgestellten Algorithmen veranschaulicht. Die dunkelgrauen Rechtecke stellen dabei die Beschriftungsboxen mit Überlappungen und die hellgrauen Rechtecke die ohne Überlappungen dar [CMS95].

In Kapitel 6.5 werden die genaueren Details und die Implementierung von Simulated Annealing vorgestellt.

4.2 Dynamisches Labeling

Eine möglichst optimale statische Beschriftung zu erzeugen, ist das Ziel der bisher vorgestellten Beschriftungsalgorithmen. Diese liefern allerdings nicht immer ein überlappungsfreies

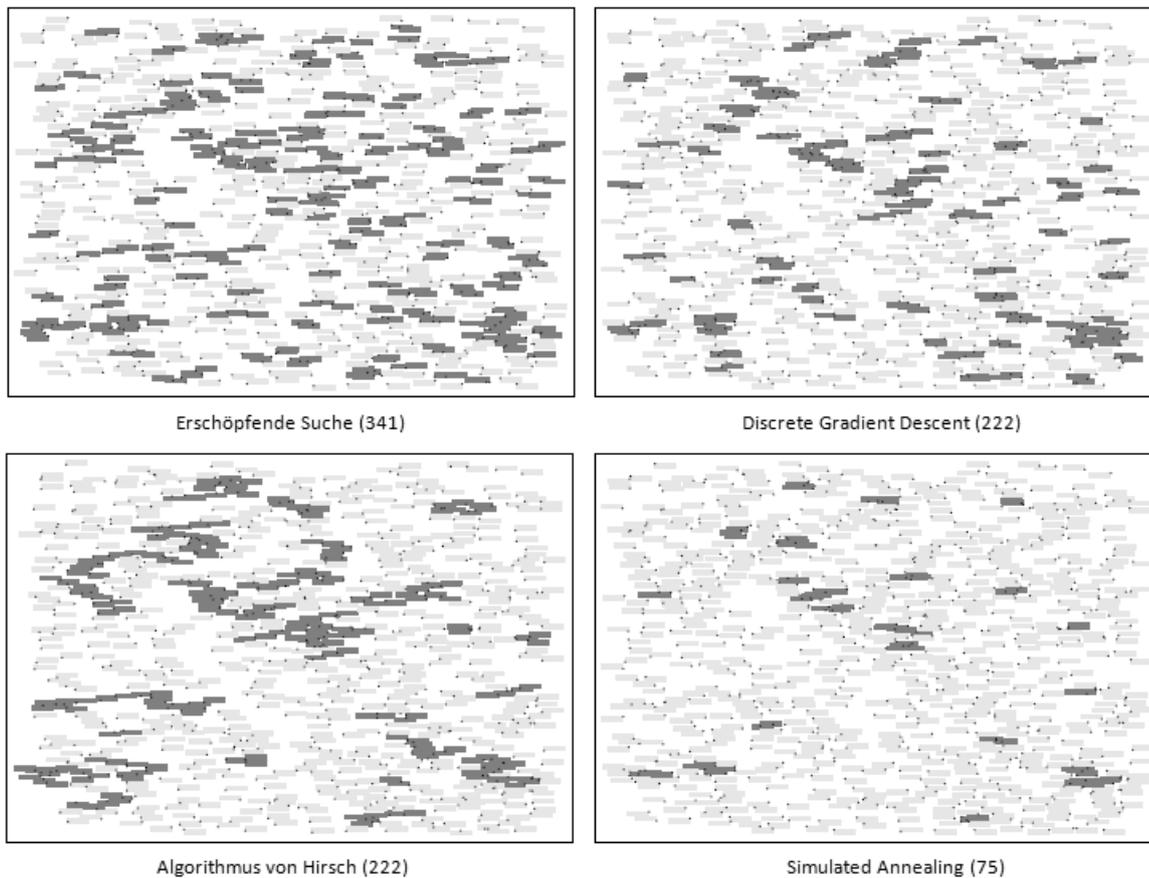


Abbildung 4.5: Vergleich zwischen den vorgestellten statischen Algorithmen (siehe [CMS95, Seite 217])

Ergebnis, das von der Anzahl der zu beschrifteten Objekten und der begrenzten Auflösung der Ausgabegeräte abhängt. Beim dynamischen Labeling werden andere Techniken angeboten, die dieses Problem angehen. Dynamische Techniken unterstützen eine sich ständig ändernde Darstellung in Echtzeit und damit auch eine sich ständig ändernde Position der Beschriftung, die durch die Interaktivität eines Benutzers verursacht wird. Interaktivität und Dynamik spielen somit eine große Rolle. Dynamische Techniken erfüllen nach [SM04] folgende Eigenschaften:

- Bilder werden ständig aktualisiert, um verschiedene visuelle Sichten auf große Objektmengen zuzulassen.
- Intuitive Interaktionstechniken werden zum Wechseln der Sichten genutzt.
- Effektive Berechnungsalgorithmen werden für das Auffinden von Details von Interesse und für das Ausblenden nicht-relevanter Beschriftungen eingesetzt.

Generell versuchen dynamische Beschriftungsalgorithmen ein ausgewogenes Verhältnis zwischen Beschriftungsqualität und der dazu benötigten Laufzeit zu finden. Unterschiedliche platzsparende Techniken zur dynamischen Beschriftung wurden bereits entwickelt, die im Folgenden vorgestellt werden.

4.2.1 One at a time-Techniken

Zu diesen Techniken zählen nach [FKCP99] Infotips, mauszeigersensitive Beschriftungen am Seitenfenster und das Temporal Brushing.

Infotip

Beim klassischen *Infotip* erscheint eine Beschriftung nur dann, wenn der Mauszeiger sich über das Objekt hinweg bewegt und anschließend wieder verschwindet. Die Beschriftung wird dabei direkt neben dem Objekt angezeigt.

Vorteil dieser Technik ist, dass sie einfach zu realisieren ist und sie wenig Rechenaufwand benötigt. Nachteil ist, dass immer nur eine Beschriftung angezeigt werden kann. Das bedeutet, falls alle Beschriftungen gelesen werden möchten, muss jedes einzelne Objekt selektiert werden. Außerdem können verdeckte Objekte nicht beschriftet werden.

Mauszeigersensitive Beschriftungen am Seitenfenster

Die mauszeigersensitiven Beschriftungen am Seitenfenster ähneln dem klassischen Infotip stark. Sobald der Mauszeiger sich über ein Objekt hinweg bewegt, wird die Beschriftung nicht direkt neben dem Objekt angezeigt, sondern separat in einem Seitenfenster. Diese Technik eignet sich vor allem dann, wenn die Beschriftungen sehr lang oder strukturiert sind.

Vorteil dieser Technik ist, dass andere Objekte nicht verdeckt werden. Ein Nachteil ist allerdings, dass die ständige Hin- und Herbewegung der Augen anstrengend werden kann.

Temporal Brushing

Beim *Temporal Brushing* erscheinen die Beschriftungen dann, wenn der Mauszeiger sich über ein Objekt hinweg bewegt. Der Unterschied zum Infotip ist aber der, dass die Beschriftungen auf dem Bildschirm erhalten bleiben. Dabei überdecken die neu angezeigten Beschriftungen einfach die alten.

Wesentlicher Vorteil gegenüber dem Infotip ist, dass mehrere Beschriftungen gleichzeitig angezeigt werden. Daraus ergibt sich jedoch der Nachteil, dass Überschneidungen von Beschriftungen nun möglich sind.

4.2.2 Änderung der Gesamtansicht

Bei der Änderung der Gesamtansicht wird mittels Zoomen und Filtern versucht, dynamische Beschriftungen zu erzeugen. Hierfür kann das *Mantra der Informationsvisualisierung* eingesetzt werden [Shn96]:

„Overview first, Zoom and Filter, then Detail-on-Demand.“

Ausgangspunkt ist ein Überblicksbild, das die Eigenschaft der Objekte repräsentiert und der Orientierung dient. Mittels Zoomen kann der Fokus auf einen Teilbereich der Objektmenge gelenkt werden. Dabei können weniger relevante Bereiche vereinfacht dargestellt werden und interessante Bereiche vergrößert werden. Es wird solange in einen Bereich hinein gezoomt, bis genug Platz vorhanden ist, um die gewünschten Beschriftungen ohne Überlappung einzublenden. Diese Technik kann gut mit statischen Techniken, wie etwa dem 4-Positionen-Modell, kombiniert werden, um stufenweise mehr Detail darstellen zu können. Durch Anwendung von Filteroperationen lassen sich Beschriftungen ausblenden, die im Kontext einer konkreten Aufgabe nicht zur Problemlösung beitragen oder nicht von Interesse sind. Ebenfalls denkbar ist eine „Alles oder Nichts“-Beschriftung, in der Beschriftungen genau dann angezeigt werden, wenn die Anzahl aller Beschriftungen bzw. Objekte unter ein bestimmtes Limit fällt. Dies ist jedoch nur dann akzeptabel, wenn die Daten einfach und sinnvoll in kleine Untermengen gefiltert werden können, was nicht immer der Fall ist. Schließlich werden nach dem Zoomen und Filtern beliebige Details von Interesse (Detail-on-Demand) erzeugt.

Basierend auf dem Mantra der Informationsvisualisierung wurden weitere Anzeigetechniken entworfen, die von einer Übersichtsdarstellung ausgehen und zusätzliche Informationen erst bei Bedarf einblenden. Zu diesen Techniken gehören Übersicht- & Detail, Fokus- & Kontext-Darstellungen und Panning-Techniken.

4.2.3 Übersicht & Detail-Darstellung

Die *Übersicht- und Detail-Darstellung* wird in zwei Ansichten unterteilt. Eine Ansicht stellt die Übersicht und die andere die Zusatzinformation dar. Abbildung 4.6 zeigt beispielhaft eine Google Maps Übersicht- und Detail-Darstellung. Die Übersicht-Darstellung wird in der unteren rechten Ecke der Detail-Darstellung eingefügt. Diese beinhaltet ein interaktives Rechteck, dessen Position entsprechend dem ausgewählten Bereich in der Detail-Ansicht aktualisiert wird. Das bedeutet, Bildverschiebungen in der Detail-Ansicht werden unmittelbar in der Übersicht angepasst. Die Detail-Ansicht stellt einen subregionalen Bereich der Übersicht dar und wird in höherer Auflösung mit zusätzlichen Informationen abgebildet. Bildverschiebungen in der Übersicht werden allerdings nur dann in der Detail-Ansicht reflektiert, wenn der Nutzer die Maustaste loslässt. Dadurch kann der Nutzer die Übersicht erkunden, ohne die dargestellte Information in der Detail-Ansicht zu verändern. [CKB09].

4 Automatische Platzierung von Labels



Abbildung 4.6: Beispielhafte Google Maps Übersicht & Detail-Darstellung (siehe [CKBo9, Seite 2:3])

Nachteil dieser Technik ist, dass eine mentale Verknüpfung zwischen Übersicht- und Detaildarstellung schwer sein kann. Aufgrund der Trennung in zwei Ansichten müssen die Nutzer separat mit diesen Ansichten interagieren.

4.2.4 Fokus- & Kontext-Darstellung

Bei der *Fokus- und Kontext-Darstellung* wird versucht, detaillierte Informationen (wie Beschriftungen) direkt an den Stellen anzureichern, die sich im Fokus des Interesses befinden. Das bedeutet, gewisse Bereiche werden im Fokus hervorgehoben. Damit ein Bezug zum Rest erhalten bleibt, wird nur ein Teil der Umgebung im Kontext dargestellt. Dadurch kann die Wahrnehmung des Menschen ausgenutzt werden, da die Aufmerksamkeit auf wichtige Bereiche gelenkt wird. Im Gegensatz zur Übersicht- und Detail-Darstellung werden Fokus und Kontext nicht getrennt voneinander auf dem Bildschirm angezeigt. Grundsätzlich lassen sich verschiedene Fokus- und Kontext-Techniken unterscheiden. Im Folgenden wird eine Fokus- und Kontext-Technik vorgestellt, die für eine dynamische Beschriftung eingesetzt werden kann. Für eine vollständige Beschreibung aller Techniken siehe [CKBo9].

Linsen stellen ein Konzept zur lokalen Anpassung dar, die interaktiv durch den Nutzer positioniert werden, wodurch die visuelle Repräsentation in diesem Bereich verändert wird. Typischerweise verzerren Linsen lokale Bereiche in einem Darstellungsraum. Für die Darstellung von dynamischen Beschriftungen können Linsen als Operatoren zur Anpassung des Detailgrades eingesetzt werden, um zusätzliche Aspekte der gegebenen Objekte im Linsenbereich zu visualisieren. Die *Fisheye View* stellt eine Linse dar, die geeignete Verzerrungen des Bildes einsetzt, sodass der periphere Kontextbereich verkleinert und der zentrale Fokusbereich vergrößert wird. Abbildung 4.7 stellt solch eine Verzerrung dar. Der Fokus ist vergrößert und wird innerhalb seines Kontextes abgebildet. Der zentrale Bereich im

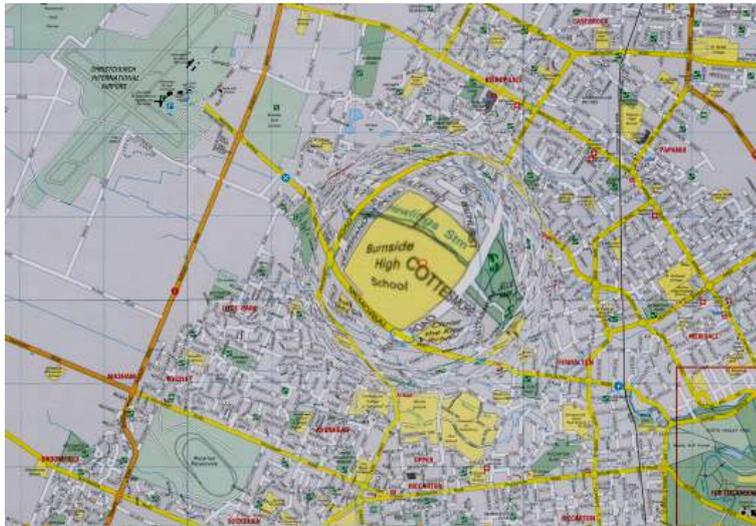


Abbildung 4.7: Fisheye View für Fokus & Kontext-Darstellung (siehe [CKBo9, Seite 2:3])

Fokus wird vergrößert und der weniger interessante Kontextbereich wird ausgehend vom Fokusbereich stetig in nicht-linearer Form verkleinert. Dadurch wird ein kontinuierlicher Übergang von Fokus- zu Kontextbereich erreicht. Für solch eine Umsetzung wird eine *Degree of Interest (DOI)* Funktion über die Objekte definiert [CKBo9]:

$$DOI(x | y) = - \text{Dist}(x,y)$$

Das x stellt ein Objekt (z.B. ein Punktobjekt mit Beschriftung) dar und das y den aktuellen Fokus. Dist steht für die Distanz eines Objekts zum Zentrum des Fokus. Das bedeutet, der Grad des Fokusbereichs nimmt mit der Entfernung ab. Die Entfernung wird mit Hilfe der Euklidischen Distanz zwischen den Objekten und dem Blickpunkt festgelegt. Die Wahl der Entfernung entscheidet dabei, wie stark der Fokusbereich hervorgehoben und ausgebreitet wird. Über die Entfernung lässt sich zudem der Übergang vom Fokus zum Kontext definieren. Der Übergang erfolgt über eine geometrische Transformationsfunktion, wie beispielsweise über eine lineare, hemisphäre, trigonometrische oder hyperbolische Funktion. Je nach Funktion kann der Übergang abrupt oder weicher definiert werden. Demzufolge wird über die Transformationsfunktion die Größe, die Position und der Detaillierungsgrad jedes Objektes bestimmt.

Der größte Nachteil der Fisheye View ist, dass durch eine Verzerrung des Fokusbereichs die Position der Objekte verändert wird. Dadurch ist die Lokalisierung der Objekte nicht mehr eindeutig. Ein weiterer Nachteil ist, dass durch die Verzerrung nicht unbedingt die Beschriftungen der Objekte überlappungsfrei dargestellt werden.

4.2.5 Panning

Das Zooming kann auch als Fokus- und Kontext-Technik verstanden werden, da temporär ein animierter Wechsel zwischen Fokus und Kontext in einer graphischen Darstellung erreicht werden kann. Die Nutzer vergrößern (hineinzoomen für Fokus-Ansicht) oder verkleinern (herauszoomen für Kontext-Ansicht) die Ansichten, ohne diese parallel betrachten zu können. Je weiter hineingezoomt wird, desto mehr Beschriftungen werden überlappungsfrei eingeblendet. Das Zooming kann mit der Übersicht- und Detail-Darstellung kombiniert werden. Neuere Arbeiten auf diesem Gebiet beschäftigen sich mit der Schaffung effektiver Übergänge von Fokus zu Kontext, Kontext zu Fokus und auch von Fokus zu neuem Fokus, das auch als *Panning* bezeichnet wird [CKB09]. *Space-Scales-Diagramme* bilden die Grundlage, um solche Pan/Zoom-Übergänge auf verschiedene Arten zu lösen (siehe Abbildung 4.8). Das Pan steht für die Verschiebung und das Zoom für das Vergrößern oder Verkleinern der Bildschirmanzeige. Für das Space-Scales-Diagramm werden auf mehreren horizontalen Ebenen immer kleiner werdende Bilder übereinander sortiert. Die Navigation erfolgt mittels einem Fenster fixer Größe (Viewing Window). Das Fenster entspricht dem Fokus und wird auf dem Bildschirm projiziert. Eine Vergrößerung der Bildschirmanzeige erfolgt durch das Verschieben des Fensters auf eine obere Ebene. Das horizontale Verschieben auf derselben Ebene entspricht einem Pan. Die x- und z-Koordinaten entsprechen dem Space und die y-Koordinate dem Scale, also dem Vergrößerungsfaktor. Wird das Bildschirmfenster entlang der y-Koordinate vertikal verschoben, dann entspricht dies einem Zoomen ohne Pan [CKB09].

Der Nachteil einer Kombination von Panning und Zooming ist, dass das Viewing Window durch das gleichzeitige Verschieben und Vergrößern bzw. Verkleinern auf verschiedene

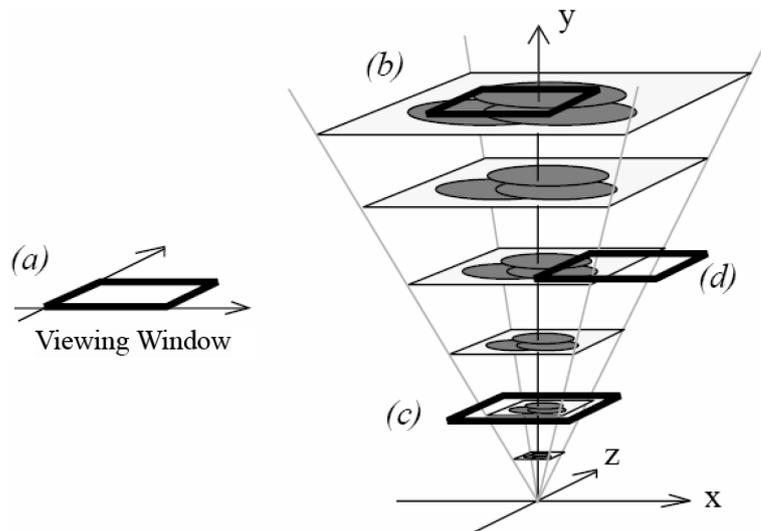


Abbildung 4.8: Space-Scales-Diagramm (siehe [CKB09, Seite 2:9])

Bereiche des Bildschirms gesetzt werden kann, sodass das Raumverständnis für die Nutzer erschwert wird.

4.2.6 Dynamic Sampling

Chalmers et al. [CIP96] nutzen das *Dynamic Sampling*, das versucht, eine exzessive Anzahl an Beschriftungen von Punktobjekten zu vermeiden. Dafür wird nur eine geringe Anzahl an Beschriftungen parallel eingeblendet, indem nur ein bis drei Beschriftungen gleichzeitig dargestellt werden. Diese sind abhängig von der Aktivität des Nutzers. Dabei wird jede Sekunde die zuletzt aktivierte Beschriftung entfernt, sodass neue Beschriftungen eingeblendet werden können. Durch das Selektieren eines Punktobjektes mit dem Mauszeiger, werden sukzessiv auch die Beschriftungen (maximal drei) in der Umgebung angezeigt. Dabei werden nur die Beschriftungen in der Umgebung eingeblendet, die sich nicht überlappen.

Nachteil von Dynamic Sampling ist jedoch, dass nur wenige Beschriftungen gleichzeitig eingeblendet werden.

4.2.7 Excentric Labeling

Fekete und Plaisant [FKCP99] führten das *Excentric Labeling* ein, das die Idee des Infotips um mehrere, parallel angezeigte Infotips erweitert, da Informationsobjekte um den Mauszeiger herum ohne Überlappungen beschriftet werden. Der Fokus wird demnach durch den Mauszeiger gesteuert. Sobald der Mauszeiger über eine Sekunde auf einer Stelle verweilt, werden alle Beschriftungen um den Mauszeiger herum eingeblendet. Die Umgebung wird dabei durch einen Kreis mit einem bestimmten Radius festgelegt. Dieser Radius kann durch das Scrollrad vergrößert oder verkleinert werden. Die Beschriftungen werden durch Linien mit ihren zugehörigen Punktobjekten verbunden. Jedes Informationsobjekt hat dabei dieselbe Farbe wie die Verbindungslinie zur entsprechenden Beschriftung und die zugehörige Beschriftungsbox. Abbildung 4.9 zeigt, wie Beschriftungen mit ihren Objekten verbunden werden. Wenn Beschriftungen einmal eingeblendet sind, kann der Nutzer den Mauszeiger über dem Bildschirm bewegen, wodurch die Beschriftungen der neu ausgewählten Objekte dynamisch aktualisiert werden. Das Beenden der dynamischen Aktualisierung erfolgt durch eine Interaktion des Nutzers, wie etwa einem Mausklick.

Gerade bei Visual Clutter in Scatterplots ist das Excentric Labeling besonders gut geeignet. Ein großer Vorteil gegenüber den anderen vorgestellten Techniken ist, dass garantiert keine Überlappungen zwischen den Beschriftungen entstehen. Außerdem arbeitet das Excentric Labeling bei einer Selektion eines neuen Fokus schnell und die Beschriftungen sind gut lesbar. Zudem werden alle Beschriftungen (und nicht nur eine Teilmenge) innerhalb der Fokussierung eingeblendet. Im Falle von zu vielen Informationsobjekten innerhalb der Fokussierung kann der Algorithmus erweitert werden, sodass die Anzahl an Labels angezeigt wird und nur zufällig ausgewählte Labels eingeblendet werden. Ein weiterer wichtiger Vorteil ist, dass Excentric Labeling ein Overplotting von Punktobjekten (z. B. in Scatterplots)

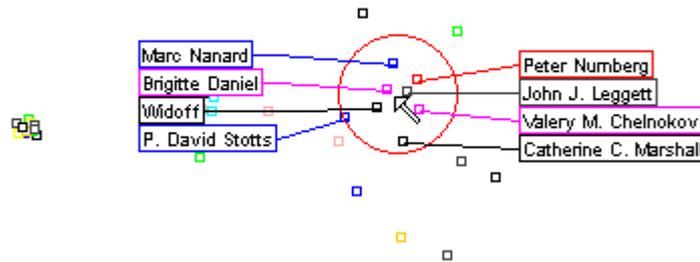


Abbildung 4.9: Excentric Labeling: Beschriftungen von Objekten im Fokusbereich werden außerhalb vom Kreis eingeblendet und mit ihren entsprechenden Objekten durch eine Linie verbunden (siehe [FKCP99, Seite 512])

berücksichtigt, da in solchen Fällen mehrere Verbindungslinien aus den aufeinanderliegenden Punkten ausgehen und die Beschriftungen untereinander aufgelistet werden.

Für das Layout der Linien zwischen den Objekten und den Beschriftungen haben Fekete und Plaisant [FKCP99] verschiedene Variationen vorgeschlagen, die im Folgenden erläutert werden.

1. Non-Crossing Lines Labeling – Radial Labeling

Das *Non-Crossing Lines Labeling* vermeidet Kreuzungen zwischen den Verbindungslinien (siehe Abbildung 4.10). Dafür werden Beschriftungen linksbündig auf zwei Seiten aufgelistet und jeweils auf der linken und rechten Seite ein Abstand zu den Beschriftungsblöcken gehalten. Für Beschriftungen auf der linken Seite werden drei Linien verwendet, um sie mit ihrem Objekt zu verbinden: Eine Linie wird ausgehend vom Objekt bis zum Kreis gesetzt. Diese Linien können immer so gesetzt werden, dass sie sich nicht kreuzen. Die nächste Linie wird vom Kreis zum linken Rand gezeichnet. Die letzte Linie wird vom linken Rand bis zur rechten Seite der Beschriftungsbox gesetzt und sollte möglichst kurz sein. Für Beschriftungen auf der rechten Seite werden

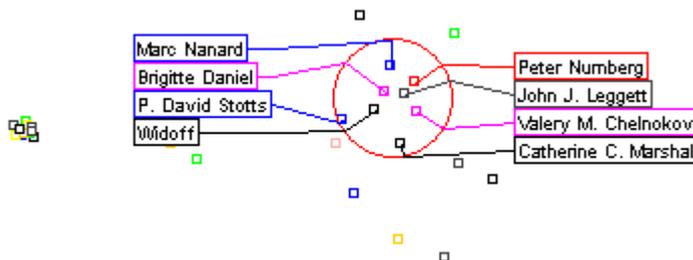


Abbildung 4.10: Non-Crossing Lines Labeling (siehe [FKCP99, Seite 515])

nur zwei Linien benötigt: Die erste Linie wird vom Objekt bis zum Kreis gesetzt. Die zweite Linie wird vom Kreis bis zur Beschriftungsbox gesetzt. Die Reihenfolge der Beschriftungen auf beiden Seiten hängt von der Linie ab, die sich zwischen einem Objekt und dem Kreis befindet.

Diese Technik vereinfacht die Zuordnung der Beschriftung zum entsprechenden Objekt. Das Non-Crossing Lines Labeling kann in kartenartigen Anwendungen genutzt werden, in denen die Reihenfolge der Beschriftungen unwichtig ist.

2. Vertically Coherent Labeling

Falls die vertikale Reihenfolge der Objekte wichtig ist, kann das *Vertically Coherent Labeling* benutzt werden. In diesem Fall werden allerdings Kreuzungen zwischen den Linien nicht aufgehoben. Dafür wird eine vertikale Reihenfolge der Objekte bewahrt. Sowohl auf der linken als auch auf der rechten Seite werden Linien zwischen den Objekten und ihren zugehörigen Beschriftungen direkt gesetzt. Die Reihenfolge der Objekte wird mit Hilfe der y-Komponente eines Objekts bestimmt. Fekete und Plaisant [FKCP99] beschreiben diesen Algorithmus als die beste und geeignetste Variante für die Visualisierungen von Daten, da die vertikale Reihenfolge der Beschriftungen die Zuordnung zu den Punktobjekten besser verdeutlicht. Abbildung 4.9 entspricht dem Vertically Coherent Labeling.

3. Horizontally Coherent Labeling

Wenn eine horizontale Reihenfolge der Objekte wichtig ist, kann das *Horizontally Coherent Labeling* benutzt werden. Der Unterschied zu den beiden bisherigen Algorithmen ist, dass die Beschriftungen nun nicht mehr linksbündig aufgelistet werden. Die Beschriftungen werden in Abhängigkeit der x-Komponente eines Objekts aufgelistet. Abbildung 4.11 illustriert solch eine Anordnung der Beschriftungen. Auf der linken Seite werden zwei Linien als Verbindung benutzt: Die erste Linie wird zwischen dem Ende der Beschriftungsbox und dem festgelegten Rand gesetzt. Die zweite Linie wird zwischen dem Rand und dem Objekt gesetzt. Die Position des Randes variiert allerdings von Beschriftung zu Beschriftung, da der Rand von der Position der Beschriftung abhängt. Auf der rechten Seite wird dagegen nur eine direkte Linie zwischen einem Objekt und seiner entsprechenden Beschriftungsbox gesetzt.

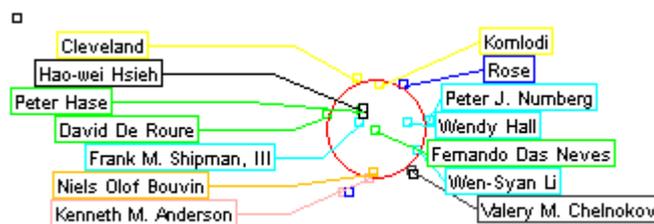


Abbildung 4.11: Horizontally Coherent Labeling (siehe [FKCP99, Seite 515])

5 Dynamisches Labeling mit Eyetracking

Das Ziel der neuen Technik ist, dass Punktobjekte in Scatterplots nur innerhalb der Fokussierung mit Hilfe von Eyetracking-Daten dynamisch beschriftet werden. Im letzten Kapitel wurden verschiedene Techniken und Algorithmen zur Beschriftung von Informationsobjekten vorgestellt. In diesem Kapitel wird nun erläutert, wieso die neue Technik die Labels dynamisch anzeigt, welcher dynamische Algorithmus für die neue Technik am geeignetsten ist und weshalb das Eyetracking für die Auslösung der Labels eingesetzt wird. Anschließend werden verwandte Arbeiten vorgestellt, die als Motivation für den Einsatz von Eyetracking dienen.

5.1 Auswahl eines Labeling-Algorithmus

Die größte Herausforderung bei der statischen Beschriftung von Objekten stellt die begrenzte Auflösung der Ausgabegeräte dar, da diese eine Begrenzung der darstellbaren Information verursacht. Das hat zur Folge, dass entweder einzelne Beschriftungen ausgeblendet werden, Überlappungen zwischen den Beschriftungen entstehen oder Beschriftungen Objekte überdecken und dadurch ihre Information verloren geht. In anderen Fällen werden die Beschriftungen so weit ausgebreitet, dass die Beziehung zwischen Objekt und Beschriftung nicht mehr eindeutig ist. Diese Probleme verstärken sich, wenn die Datendichte steigt oder die Beschriftungen sehr lang sind [FKCP99].

Dynamische Interaktionstechniken erlauben eine höhere Genauigkeit durch das Filtern von Objekten und durch die Selektion von beliebigen Objekten von Interesse. Der Nutzer kann mit dynamischen Techniken interaktiv die für ihn relevanten Objekte einblenden, sodass keine Information verloren geht. Der Nutzer bestimmt somit seinen Fokus selbst, sodass nur Details für selektierte Objekte eingeblendet werden. Überlappungen zwischen Beschriftungen können verhindert werden, da der Platz außerhalb der Fokussierung für Details ausgenutzt werden kann. Außerdem ist das Einblenden von Details in der Fokussierung sinnvoll, da Details außerhalb der Fokussierung die Wahrnehmung stören.

Im letzten Kapitel wurden bereits verschiedene dynamische Interaktionstechniken vorgestellt. Für die Wahl einer dynamischen Interaktionstechnik muss allerdings bedacht werden, welche dynamische Technik sich für das Eyetracking eignet. Da das Excentric Labeling mehr Beschriftungen gleichzeitig als der Infotip und das Dynamic Sampling anzeigt und keine Überschneidungen von Beschriftungen im Gegensatz zum Temporal Brushing erzeugt, wurde Excentric Labeling für die neue Technik eingesetzt. Außerdem ist das Excentric Labeling in Kombination mit einem Eyetracker besser geeignet als der Infotip, der eine

sehr präzise Selektion der Objekte benötigt. Der Grund hierfür ist, dass der Eyetracker schlechter bei einer präzisen Selektion arbeitet, dafür aber in groben Bereichen schneller ist [Vero8]. Da im Rahmen dieser Diplomarbeit der Eyetracker nicht zur direkten Manipulation genutzt wird, kommt eine Änderung der Gesamtansicht mittels Zooming nicht in Frage. Eine Trennung der Übersicht und des Details ist nicht erwünscht. Die Fisheye View ändert durch eine Verzerrung des Darstellungsraums die Position der fokussierten Objekte. Das hat zur Folge, dass eine Lokalisierung von Punktobjekten im fokussierten Bereich nicht mehr möglich.

Excentric Labeling ist ein Algorithmus in Kombination mit der Maus. Deshalb musste eine Veränderung vorgenommen werden, um eine Kombination mit einem Eyetracker zu ermöglichen: Der ursprüngliche Algorithmus löst Beschriftungen aus, die sich um den Mauszeiger herum befinden. Für die neue Technik werden nicht die Mauskoordinaten aufgenommen, sondern die Augkoordinaten, um den Fokus des Nutzers zu ermitteln. Da jedoch Beschriftungen außerhalb der Fokussierung erscheinen, um Überlappungen zu vermeiden, muss verhindert werden, dass sich der Fokus beim Lesen ändert. Dies kann erreicht werden, indem der Nutzer selbst bestimmt, wann das Bild „eingefroren“ wird. Die beste Möglichkeit dafür ist das Betätigen einer beliebigen Taste. Für die neue Technik wurde die Leertaste zum Einfrieren des Bildes eingesetzt, sodass die Beschriftungen, die direkt außerhalb der Fokussierung dargestellt werden, gelesen werden können.

In Kapitel 6.4 werden die genaueren Details und die Implementierung von Excentric Labeling vorgestellt.

5.2 Eyetracking zur Interaktion

Eine Kombination von dynamischem Labeling mit Eyetracking ist bis heute aus Veröffentlichungen nicht bekannt. Im Rahmen dieser Diplomarbeit wird diese Kombination umgesetzt, um zu testen, ob die Auslösung der Beschriftungen von Punktobjekten schneller ist als die Maus.

Die neue Technik soll Beschriftungen von Punktobjekten dynamisch in einfachen Scatterplots auslösen. Dabei sollen nur die Beschriftungen eingeblendet werden, die im Fokus des Betrachters liegen und somit von Interesse sind. Beschriftungen im peripheren Sichtfeld sind für den Nutzer uninteressant und sollen nicht eingeblendet werden, denn außerhalb der Fovea Centralis liegt die Sehschärfe nur noch bei 15 bis 50 Prozent [Jac95], sodass im peripheren Sichtfeld Objekte nicht mehr klar aufgenommen werden können. Um ein Objekt eindeutig und scharf zu erkennen, muss es fixiert werden. Demnach bietet der Eyetracker einen wesentlichen Vorteil, da er nur den Fokus des Nutzers betrachtet. Mit dieser Information können somit Beschriftungen innerhalb der Fokussierung dynamisch ein- und ausgeblendet werden.

Vertegaal beschreibt in [Vero8], dass der Eyetracker aufgrund des Midas Touch Problems (siehe Kapitel 2.5) ein Zeigegerät, wie die Maus, nicht einfach ersetzen kann. Der Eyetracker eignet sich eher zur Auswahl diskreter Objekte, wie zum Beispiel Buttons. In dieser Arbeit

wird diese Idee umgesetzt: Der Eyetracker wird nicht zur direkten Manipulation eingesetzt, sondern nur zur Selektion von Punkten in Scatterplots. Der Eyetracker wird somit nicht als Mausersatz eingesetzt, um Objekte anzuklicken und an eine andere Stelle zu bewegen oder um ein Button anzuklicken. Die Eyetracking-Daten werden ausschließlich dafür genutzt, um die Information zu erhalten, wann und wo der Nutzer den Bildschirm betrachtet, indem die Augenkoordinaten in Echtzeit zurückgeliefert werden [FM02]. Diese Art von Blickbedienung wird in [Jac95] als *Non Command-Based User Interfaces* bezeichnet, die eine andere Eingabe für den Nutzer repräsentiert, indem bestimmte Kommandos passiv ausgelöst werden. Das System beobachtet den Nutzer und antwortet entsprechend, ohne auf einen Kommando des Nutzers zu warten. Non Command-Based User Interfaces können mit expliziten Kommandos kombiniert werden, wie Maus- oder Tastatur-Events, um einen höheren Grad an Interaktivität zwischen Mensch und Computer zu erlangen.

Allgemeine Vorteile für den Einsatz von Eyetracking anstelle der Maus hat Vertegaal [Vero8] zusammenfassend dargestellt:

Wenn die Hände aufgrund motorischer Einschränkungen nicht zur Verfügung stehen oder anderweitig eingesetzt werden, dann kann stattdessen der Blick zur Interaktion genutzt werden. Augen enthalten die schnellsten Muskeln im menschlichen Körper. Das impliziert, dass die Blickbedienung die schnellste Eingabemethode ermöglicht. Nutzer können tausende Augenbewegungen durchführen, ohne schnell zu ermüden. Außerdem wird das Risiko einer Verletzung durch ständig wiederholte Bewegungsabläufe verringert, da viele manuelle Handlungen wegfallen. Der Blick als Eingabe muss nicht extra erlernt werden, da dies eine intuitive Handlung ist. Weiterhin tendieren Menschen dazu, Zielobjekte zunächst mit den Augen zu selektieren, bevor sie eine manuelle Aktion durchführen. Die Augen geben somit vorher einen Hinweis auf das Zielobjekt, bevor es mit einem Zeigergerät selektiert wird.

5.3 Performanz von Eyetracking und Maus

Der Hauptaspekt für den Einsatz von Eyetracking anstelle der Maus für die neue Technik ist, dass die Augen die schnellsten Muskeln im menschlichen Körper haben und sie vorher ein Zielobjekt fixieren, bevor sie es auswählen. Demzufolge wird erwartet, dass der Eyetracker eine schnellere Eingabemethode darstellt als die Maus. Vergleiche zwischen Eyetracking und Maus hinsichtlich der Zeit zur Selektion von Zielobjekten wurden bereits durchgeführt.

Vertegaal [Vero8] hat die Performanz zwischen Eyetracking, Maus und Eingabestift mit Hilfe des Fitts' Gesetzes verglichen. Für das Eyetracking wurden zwei verschiedene Techniken genutzt:

- **Maus**
Für diese Technik wurde eine Maus mit absoluten Koordinaten eingesetzt. Eine Maustaste wurde zum Anklicken einer Schaltfläche genutzt.
- **Eingabestift**
Für diese Technik wurde ein Eingabestift zum Anklicken einer Schaltfläche genutzt.

- **Eyetracking mit manuellem Klick**
Mit dieser Technik wurden Blickpunkte als Mauscursor interpretiert und eine Maustaste zum Anklicken einer Schaltfläche genutzt.
- **Eyetracking mit Dwell Time Klick**
Mit dieser Technik wurden Blickpunkte als Mauscursor interpretiert und das Anklicken einer Schaltfläche wurde automatisch bei einer Fixierung von 100 ms ausgelöst.

Die Ergebnisse haben gezeigt, dass das Eyetracking mit Dwell Time Klick schneller als das Eyetracking mit manuellem Klick, dem Eingabestift und der Maus ist. Das Eyetracking mit manuellem Klick ist dabei um 16% schneller als die Maus. Eingabestift und Maus haben fast dieselbe Performanz. Nachteil der Eyetracking-Techniken ist jedoch, dass beide eine hohe Fehlerrate besitzen. Eyetracking mit manuellem Anklicken unterstützt einen besten Trade-off zwischen Geschwindigkeit und Genauigkeit [Vero8].

Ware and Mikaelian [WM87] führten ebenfalls ein Experiment durch, wobei die Zeit für das Selektieren eines Objekts gemessen wurde. Dafür wurden drei verschiedene Methoden eingesetzt:

- **Dwell Time Button**
Mit dieser Methode wurde ein Objekt selektiert, wenn dieser für eine bestimmte Verweildauer fixiert wurde.
- **Screen Button**
Bei dieser Methode wurde auf den Bildschirm eine große rechteckige Fläche als Button festgelegt. Um ein Objekt zu selektieren, wurde dieses erstmals betrachtet und anschließend musste der Button fixiert werden.
- **Hardware Button**
Mit dieser Methode wurde ein Objekt selektiert, wenn der Nutzer eine Taste gedrückt hielt, während er das Objekt fixierte.

Diese drei Methoden wurden mit der Maus bezüglich der Geschwindigkeit verglichen. Für die Evaluation wurde das Fitts' Gesetz modifiziert. Die Ergebnisse haben gezeigt, dass die drei Methoden wesentlich schneller sind als die Maus. Die Dwell Time Button- und Hardware Button-Methode haben sehr ähnliche Zeiten zurückgeliefert und sind schneller als die Screen Button-Methode. Laut ihrem Ergebnis können mit einem Eyetracker Objekte schnell selektiert werden, falls die Fläche der Zielobjekte nicht zu klein ist. Wenn das Eyetracking mit einer zusätzlichen Maus- oder Tastatureingabe kombiniert wird, können Objekte am schnellsten selektiert werden. Allerdings ist zu beachten, dass die Zielobjekte nicht zu klein sein dürfen, da sich ansonsten die Fehlerrate erhöht.

Silbert et al. [SJToo] haben unter den selben experimentellen Bedingungen auch die Performanz zwischen Eyetracking und der Maus verglichen. Ihre Ergebnisse haben ebenfalls gezeigt, dass die Auswahl von Zielobjekten mit einem Eyetracker schneller ist als mit der Maus, wenn der Eyetracker keine wesentlichen Verzögerungen aufweist. Sie beschreiben die Auswahl von Objekten mit einem Eyetracker als schnell und mühelos. Für das Messen der mittleren Bewegungszeit wurde das Fitts' Gesetz eingesetzt. Die Analyse mit Hilfe des Fitts' Gesetz hat gezeigt, je weiter das Zielobjekt entfernt war, desto schneller war der

Eyetracker, da seine Auswahlzeiten in allen Fällen eher konstant waren. Außerdem haben sie festgestellt, dass der Eyetracker überragend arbeitet, wenn entfernte Bereiche auf dem Bildschirm sprunghaft fokussiert werden.

5.4 Hypothesen

Auf Basis der erworbenen Kenntnisse der letzten Abschnitte aus diesem Kapitel, werden zwei Hypothesen für die neue Technik aufgestellt. Diese werden durch die Ergebnisse der Eyetracking-Studie in Kapitel 8 bewiesen oder widerlegt.

Folgende Hypothesen werden für die neue Technik aufgestellt:

- **Vergleich der Zeit**

Bei der Suche nach Annotationen von Punktobjekten in Scatterplots liefert die neue dynamische Eyetracking-Technik schnellere Ergebnisse als die Maus und die statische Technik.

- **Vergleich der Genauigkeit**

Bei der Suche nach Annotationen von Punktobjekten in Scatterplots liefert die neue dynamische Eyetracking-Technik eine höhere Genauigkeit als die Maus und die statische Technik.

5.5 Verwandte Arbeiten

Da Eyetracking im Bereich der Blickbedienung noch ein relativ junges Forschungsgebiet ist, wurden zu diesem Thema bisher wenige Arbeiten veröffentlicht. Die meisten Arbeiten richten sich auf motorisch eingeschränkte Nutzer, um ihnen eine Bedienung von Desktop-Anwendungen zu ermöglichen.

EyeWindows

Fono und Vertegaal [FV05] stellten 2005 die Anwendung *EyeWindows* für Fenster in Bildschirmen vor. Mit dieser Anwendung ist es möglich, nicht überlappende Fenster durch Fixationen auszuwählen. Das fokussierte Fenster vergrößert sich dabei automatisch, die restlichen Fenster werden verkleinert und treten in den Hintergrund. Der Platz auf dem Bildschirm wird dadurch hauptsächlich für das Fokussierte genutzt und das Umschalten mit der Tastatur oder der Maus entfällt. Abbildung 5.1 (a) zeigt einen Nutzer X in einer Instant Messaging Unterhaltung mit Nutzer Y (siehe großes Fenster in der Mitte in Abbildung 5.1 (a)). Der Nutzer X bemerkt in seinem peripheren Sichtfeld die Benachrichtigung, dass Nutzer Z online ist. Nutzer X schaut unmittelbar auf die Benachrichtigung und betätigt die Leertaste. Die Benachrichtigung verschwindet und ein neues Unterhaltungsfenster mit Nutzer Z

5 Dynamisches Labeling mit Eyetracking



(a) Fokus richtet sich auf linkes IM-Fenster

(b) Fokus richtet sich auf rechtes IM-Fenster

Abbildung 5.1: EyeWindows: Fokussiertes Fenster wird automatisch vergrößert (siehe [FV05, Seite 156])

öffnet sich (siehe kleines Fenster unten rechts in Abbildung 5.1 (a)). Anschließend richtet Nutzer X seine Augen wieder auf das Unterhaltungsfenster mit Nutzer Y und antwortet ihm unmittelbar. In der Peripherie bemerkt Nutzer X eine Nachricht von Nutzer Z. Nutzer X fokussiert das Unterhaltungsfenster mit Nutzer Z, betätigt parallel die Leertaste und das Unterhaltungsfenster mit Nutzer Z wird automatisch vergrößert (siehe Abbildung 5.1 (b)), sodass dieses Fenster selektiert ist und Nutzer X sofort antworten kann.

Eyetracking ist nicht dazu geeignet, die kompletten Aufgaben der Maus zu übernehmen, da sich die Augen unkontrolliert bewegen. Eyetracking arbeitet allerdings effektiv für grobe Bereiche. In ihrer Anwendung wird der Eyetracker nicht als Musersatz angewendet, sondern nur zur Bestimmung der ungefähren Blickposition. Für die Auswahl eines Fensters reichen ungenaue Messungen der Blickpositionen völlig aus. Aufgrund dessen eignet sich in diesem Fall Eyetracking besonders gut, weil die Geschwindigkeit der Augenmuskeln ausgenutzt werden kann. Zudem fällt die Funktion des Anklickens weg, sodass die Zeit zum Selektieren eines Fensters reduziert werden kann.

Fono und Vertegaal [FV05] evaluierten die Performanz für die Selektion von Fenstern mit zwei verschiedenen Eyetracking-Techniken. Einmal konnten die Fenster durch längere Fixierung eines Fensters ausgewählt werden, das andere Mal durch Betätigen der Leertaste, wobei das auszuwählende Fenster parallel fokussiert wird. Ihre Ergebnisse zeigten, dass Eyetracking mit einer automatischen Selektion durch die Fokussierung eines Fensters fast doppelt so schnell war, wie die Selektion eines Fensters mit der Maus oder mittels Hotkeys. Das Eyetracking mit der automatischen Selektion durch die Fokussierung eines Fensters hat im Vergleich zum Eyetracking mit Betätigung der Leertaste die schnelleren Ergebnisse hinsichtlich der Zeit geliefert. Die Methode mit der Betätigung der Leertaste wurde allerdings von den meisten Nutzern bevorzugt. Das lässt sich vermutlich damit begründen, da durch eine zusätzliche Tastatureingabe die Selektion eines Fensters kontrolliert werden kann und

die Augen sich weiterhin frei bewegen können. Somit kamen Fono und Vertegaal [FV05] zu dem Entschluss, dass eine Fensterauswahl mit Hilfe von Eyetracking eine effektive Alternative bietet.

EyeWindows zeigt einige Parallelitäten zur neuen Eyetracking-Technik dieser Arbeit auf und dient aufgrund der positiven Ergebnisse als größte Motivation. EyeWindows nutzt Eyetracking, um mehrere geöffnete Fenster auf dem Desktop schneller auswählen zu können. Die Anwendung nutzt die Auflösung des Bildschirms für das Fokussierte aus, indem das fixierte Fenster vergrößert wird, um eine bessere Wahrnehmung zu ermöglichen. Die neue Eyetracking-Technik dieser Arbeit folgt demselben Prinzip: Nur Beschriftungen von fokussierten Punktobjekten werden angezeigt, sodass der Platz um den Fokus herum für genauere Details ausgenutzt wird und uninteressante Details außerhalb der Fokussierung nicht eingeblendet werden. Des Weiteren wird für eine Eyetracking-Technik von EyeWindows durch das Betätigen der Leertaste die Auswahl des Fokus gesteuert. Die neue Technik dieser Arbeit nutzt ebenfalls einen Tastendruck, um den Fokus zu bestimmen.

Text 2.0

Das Deutsche Forschungszentrum für Künstliche Intelligenz entwickelte *Text 2.0*, das interaktives Lesen ermöglicht [BBS⁺10]. Mit Hilfe eines Eyetrackers wird dabei die Blickposition auf dem Bildschirm festgestellt. Dadurch kann erkannt werden, was eine Person momentan liest. Kontextbezogen spielt das System thematisch passende Multimedia-Effekte in Echtzeit ein, wie beispielsweise Bilder, Musik oder Geräusche, die unter dem Namen *Augmented Reading* zusammengefasst werden. Falls notwendig, kann Text 2.0 fremdsprachige Textstellen übersetzen, die genau dann eingeblendet werden, wenn sie gelesen werden. Nach Bedarf

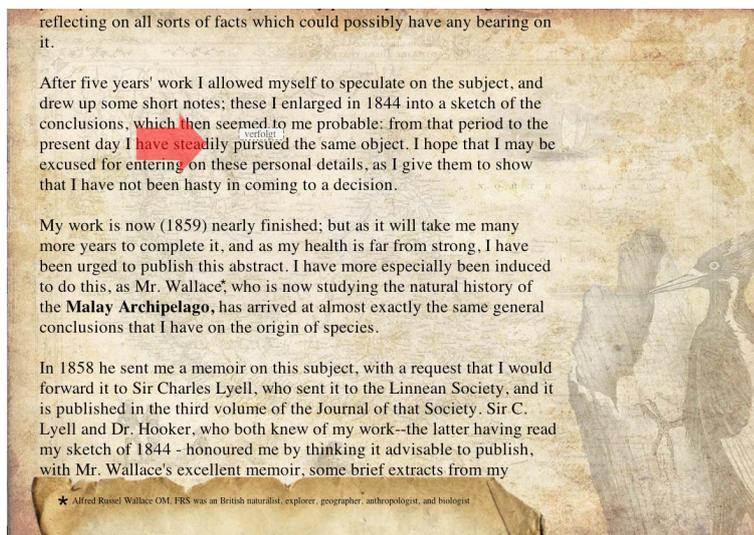


Abbildung 5.2: Text 2.0 ermöglicht Augmented Reading (siehe [BBS⁺10, Seite 4005])

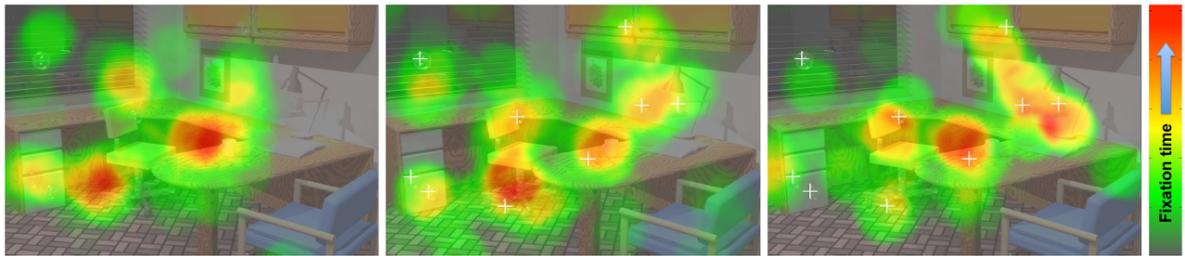


Abbildung 5.3: Verteilung der Fixationszeit mit und ohne SGD (siehe [MBGo8, Seite 51])

erfolgt eine Silbentrennung langer Wörter. Außerdem werden intelligente Fußnoten unterstützt und wenn beim Lesen der Anschluss verloren geht, markiert ein roter Pfeil die zuletzt betrachtete Stelle. Abbildung 5.2 zeigt beispielhaft eine Übersetzung sowie einen Hinweis mit Hilfe eines roten Pfeils, falls der Anschluss beim Lesen verloren geht.

Text 2.0 nutzt im Gegensatz zur neuen Eyetracking-Technik dieser Arbeit den Eyetracker nicht zur Auswahl diskreter Objekte, sondern um den Nutzer während des Lesens zu beobachten und zu unterstützen. Bei Text 2.0 handelt es sich somit um ein Attentive User Interface (siehe Kapitel 2.5). Die neue Eyetracking-Technik ist allerdings der Blickbedienung zuzuordnen. Text 2.0 konzentriert sich zudem nicht nur auf Fixationen eines Nutzers, sondern analysiert auch die Muster der Fixationen und Sakkaden. Anhand von Sakkaden zwischen zwei Fixationen wird bestimmt, ob ein Nutzer einen Text liest oder ihn überfliegt.

Improving Search Task Performance Using Subtle Gaze Direction

McNamara et al. [MBGo8] veröffentlichten 2008 eine neue Technik, genannt *Subtle Gaze Direction (SGD)*. Bei dieser Technik werden die Nutzer bei der Suche von Objekten auf bestimmte Bereiche eines Bildes gelenkt. SGD nutzt dafür eine Anpassung der Luminanz, um die Nutzer entlang einer Szene zu führen. Mit Hilfe eines Eyetrackers wird die Blickposition bestimmt und falls der Nutzer einen falschen Bereich betrachtet, wird er auf die gesuchten Objekte gelenkt, ohne spürbar auffällige Änderungen im Bild zu erzeugen. Dadurch soll die Performanz im Gegensatz zur herkömmlichen Suche verbessert werden, was auch in ihren Ergebnissen gezeigt wurde. Abbildung 5.3 illustriert die Fixationszeit mit Heat-Maps bei der Suche nach Zielobjekten. Auf der linken Seite wurde eine normale Suche vorgenommen, auf der rechten Seite und in der Mitte wurde die Suche durch Hinweise eines weißen Kreuzes unterstützt. Die Abbildung verdeutlicht durch dunklere rote Bereiche, dass in manchen Fällen für die Suche ohne SGD mehr Zeit benötigt wurde.

Subtle Gaze Direction verwendet im Vergleich zur neuen Eyetracking-Technik die Blickposition, um sie außerhalb der Fokussierung zu lenken, da im peripheren Sichtfeld die Verarbeitung von Stimuli schneller ist als in der Fovea Centralis.

6 Implementierung

Die theoretischen Erkenntnisse aus den letzten Kapiteln werden in diesem Kapitel von ihrer praktischen Seite vorgestellt. Zunächst wird auf die Anforderungen eingegangen, die an die drei Techniken gestellt werden. Anschließend wird ein allgemeiner Überblick über die verwendeten Technologien gegeben, bevor auf die Implementierung von Scatterplots und auf die Umsetzung einer statischen und dynamischen Beschriftung mit Hilfe der auserwählten Algorithmen aus Kapitel 4 eingegangen wird.

6.1 Anforderungen

In dieser Arbeit soll die Eignung von Eyetracking-Daten für die adaptive, dynamische Beschriftung in Scatterplots untersucht werden. Diese neue Technik soll anschließend mit zwei bestehenden Techniken durch eine Eyetracking-Studie verglichen werden. Bei der ersten bekannten Technik handelt es sich um eine statische Technik, die alle Beschriftungen im Scatterplot einblendet. Bei der zweiten bekannten Technik werden Maus- statt Augkoordinaten zur Auslösung der Beschriftungen im Scatterplot verwendet.

Folgende Anforderungen werden an die drei Techniken gestellt:

- Implementierung von einfachen Scatterplots mit *Processing*

- **Statische Annotation:**

Verwendung eines geeigneten statischen Algorithmus zur Beschriftung der Datenpunkte im Scatterplot.

- **Dynamische Annotation mit dem Eyetracker:**

Verwendung eines geeigneten dynamischen Algorithmus zur Beschriftung der Datenpunkte im Scatterplot, wobei die Beschriftung durch die Augkoordinaten ausgelöst wird. Zur Verwendung der Augkoordinaten wird das *Processing Easy Eye Tracker Plugin* eingesetzt.

- **Dynamische Annotation mit der Maus:**

Verwendung desselben dynamischen Algorithmus zur Beschriftung der Datenpunkte im Scatterplot, wobei die Beschriftung durch die Mauskoordinaten ausgelöst wird.

6.2 Verwendete Technologien

In diesem Abschnitt wird auf die eingesetzte Programmiersprache Processing und die mit ihr verbundenen Technologien für die Umsetzung der Anforderungen eingegangen.

6.2.1 Processing

Processing ist eine Open Source-Programmiersprache basierend auf Java. Mit ihr ist die Programmierung von Bildern, Animationen und Sound möglich. Somit ist sie besonders für Menschen mit visuellem Denken geeignet. Die Programme können anschließend als Java Applet exportiert werden. Processing wurde speziell für Studenten, Künstler, Designer und Architekten von Ben Fry und Casey Reas entwickelt und wird in einer Entwicklungsumgebung ausgeliefert [Shio8].

Die Entwicklungsumgebung setzt sich aus einem Programmfenster, einem Terminalfenster und einem Anwendungsfenster zusammen. Ein typisches Processing-Programm enthält zwei Funktionen: `setup()` und `draw()`, wobei der Programmcode in `setup()` nur einmal und der Programmcode in `draw()` kontinuierlich ausgeführt wird. Jeder Durchlauf von `draw()` zeichnet einen Frame in das Anwendungsfenster. Listing 6.1 zeigt, wie mit Processing ein Kreis auf einer bestimmten Position gezeichnet werden kann.

Listing 6.1 Processing Hauptfunktionen

```
void setup() {  
  size(400,400);  
}  
void draw() {  
  ellipse(25,25,50,50);  
}
```

Die `setup()`-Funktion legt mit `size()` die Größe des Anwendungsfensters fest und die `draw()`-Funktion zeichnet mit `ellipse()` einen Kreis im Anwendungsfenster. Dabei wird der Kreis ständig gezeichnet, weil die `draw()`-Funktion kontinuierlich ausgeführt wird. Die `draw()`-Funktion bildet die Basis, um Animationen zu erstellen.

Als Zeichengrundlage verwendet Processing das *Kartesische Koordinatensystem*, um geometrische Figuren wie Punkte, Geraden, Dreiecke, Vierecke oder Kreise einzeichnen zu können. Das Koordinatensystem von Processing ist so definiert, dass der Ursprung in der oberen linken Ecke liegt. Die positiven *x*-Werte werden somit nach rechts und die positiven *y*-Werte nach unten gezählt (siehe Abbildung 6.1). Der *x*-Wert steht dabei für die Anzahl der horizontalen Pixel und der *y*-Wert für die Anzahl der vertikalen Pixel. Ein Punkt ist also ein Pixel groß und seine Position wird durch den *x*- und *y*-Wert bestimmt. Punkte, die außerhalb des Anwendungsfensters liegen, werden nicht als Fehler behandelt, sondern einfach nicht angezeigt [Bar10].

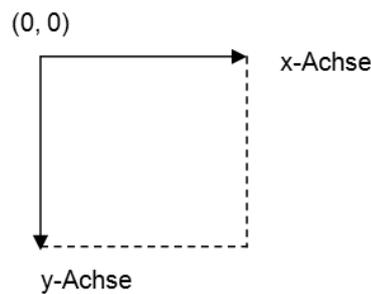


Abbildung 6.1: Koordinatensystem von Processing (Eigene Darstellung in Anlehnung an [Bar10, Seite 12])

6.2.2 Processing Easy Eye Tracker Plugin

Das *Processing Easy Eye Tracker Plugin (PEEP)*¹ ist ein Open Source-Plugin für Processing von den Entwicklern von Text 2.0 und ist Teil des Text 2.0 Frameworks. Mit PEEP werden Eyetracking-Daten aufgenommen, um mit Processing-Anwendungen interagieren zu können. PEEP ermöglicht eine Verarbeitung der Aug- und Kopfkoordinaten eines Nutzers, die von einem Eyetracker aufgenommen werden. Falls kein Eyetracker vorhanden ist, kann ein Simulator verwendet werden. Dieser Simulator kann jedoch nur die Augkoordinaten nachahmen, was mit Hilfe des Mauszeigers erfolgt. Das bedeutet, dass auf jeden beliebigen Rechner Eyetracking-Anwendungen entwickelt werden können. Wenn ein Eyetracker doch vorhanden ist, muss der Code auch nicht verändert werden. Listing 6.2 zeigt, wie mit Hilfe von PEEP die Augkoordinaten in einer Processing-Anwendung aufgenommen werden können. Vor der `setup()`-Funktion wird ein Eyetracking-Device deklariert und innerhalb der `setup()`-Funktion wird dieses geöffnet. In diesem Beispiel wird an den Stellen ein Rechteck gezeichnet, an denen der Nutzer seine Blickpunkte richtet. `et.x` und `et.y` stellen den aktuellen Blickpunkt des Nutzers dar, die als Koordinaten abgebildet werden.

Listing 6.2 PEEP für Processing

```
import eyetracking.*;

EyeTrackingDevice et;

void setup() {
    et = EyeTrackingDevice.open(this);
}

void draw() {
    rect(et.x, et.y, 30, 30);
}
```

¹<http://text20.net/>

Mit PEEP wird zudem ein *Tracking Server* und ein *Diagnosis Tool* mitgeliefert. Um PEEP ausführen zu können, muss zu Beginn der Tracking Server gestartet werden, um eine Verbindung mit dem Eyetracker herzustellen. Der Tracking Server kann auch im Simulationsmodus ausgeführt werden. Das Diagnosis Tool zeigt, ob eine Verbindung zum Eyetracker aufgebaut wurde. Mit dem Diagnosis Tool wird der Eyetracker kalibriert.

6.2.3 gwoptics

*gwoptics*² ist eine Programmbibliothek von Daniel Brown und Andreas Freise für die Entwicklungsumgebung Processing. Diese Bibliothek unterstützt eine Reihe an Tools, um zweidimensionale und dreidimensionale Graphen schnell mit Processing erstellen zu können. *gwoptics* wurde für wissenschaftliche Simulationen und für einfache Datenanzeigen entwickelt. Mit *gwoptics* ist eine grafische Darstellung von Funktionen und Daten möglich.

6.3 Einfaches Scatterplot

Ein einfaches Scatterplot (siehe Kapitel 3.4) bildet die Basis aller drei Techniken. Für die Implementierung von einfachen Scatterplots wurde die Programmbibliothek *gwoptics* eingesetzt. Diese enthält eine Klasse *Graph2D*, mit der zweidimensionale Graphen erstellt und auf verschiedene Methoden zugegriffen werden können, um die Achsen und das Layout der Graphen ändern zu können.

Da die Techniken auf verschiedene Datensätze angewendet werden, müssen die Achsenbeschriftungen und die Werte auf den Achsen automatisch an die Datensätze angepasst werden. Dafür wird ein Datensatz von einer csv-Datei in den Code geladen. Dieser besteht aus mehreren Zeilen und Spalten. In der ersten Zeile stehen die Achsenbeschriftungen eines Scatterplots. In der ersten Spalte stehen die y-Werte, in der zweiten Spalte die x-Werte und in der letzten Spalte die Beschriftungen der Punktobjekte. Um die Grenzen der beiden Achsen eines Scatterplots festzulegen, wird sowohl das Minimum als auch das Maximum aller x- und y-Werte der Punktobjekte bestimmt.

Nachdem die Grenzen der Achsen bestimmt wurden, müssen die Werte und die Abstände zwischen den Werten auf den Achsen berechnet werden. Abbildung 6.2 zeigt beispielhaft zwei x-Achsen von zwei verschiedenen Datensätzen. Erstmals wird die Differenz vom Minimum und Maximum des x-Wertes ermittelt. Dies gilt analog für den y-Wert. Anschließend wird die Anzahl an Ziffern *AZ* der Differenz gezählt. Wenn zum Beispiel der Wert des Minimums 10 und der Wert des Maximums 80 ist, beträgt der Wert der Differenz 70 und die Anzahl an Ziffern von 70 ist 2. Mit Hilfe der Potenzfunktion $10^{(AZ - 1)}$ werden die Werte und die Schritte auf den Achsen festgelegt.

²<http://www.gwoptics.org/>

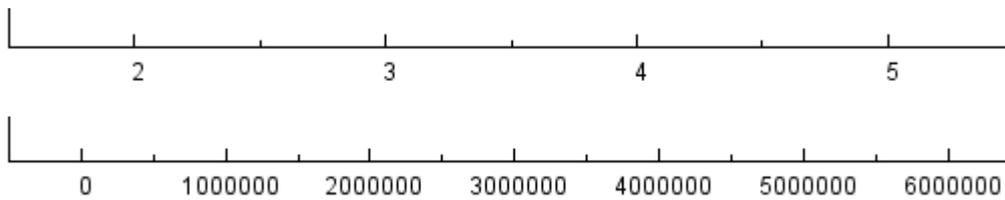


Abbildung 6.2: Werte und Abstand zwischen den Werten auf den Scatterplot-Achsen

Damit keine Punktobjekte direkt auf den Achsen liegen, werden die minimalen und maximalen Koordinaten erstmals auf ganze Zahlen ab- bzw. aufgerundet und anschließend um eine halbe Einheit verkleinert bzw. vergrößert. Mit Hilfe der Methoden von Graph2D werden unter anderem die Größe, die Position im Anwendungsfenster von Processing, die Werte auf den Achsen und die Achsenbeschriftungen des Scatterplots gesetzt. Da der Ursprung des Koordinatensystems von Processing in der oberen linken Ecke des Anwendungsfenster liegt und die Punkte in Pixel gezählt werden, müssen die Koordinaten der Punktobjekte transformiert werden, bevor sie abgebildet werden. Die Transformation der x- und y-Koordinaten erfolgt folgendermaßen:

$$x = x_{graph} + ((length(x_{graph}) / (x_{max} - x_{min})) * (x_{csv} - x_{min}))$$

$$y = y_{graph} + ((length(y_{graph}) / (y_{max} - y_{min})) * ((y_{max} - y_{min}) - (y_{csv} - y_{min})))$$

x_{graph} und y_{graph} stellen den oberen linken Pixel des Scatterplots dar. $length(x_{graph})$ und $length(y_{graph})$ stehen für die Breite und die Höhe des Scatterplots, die in Pixel angegeben werden. x_{max} , x_{min} , y_{max} und y_{min} entsprechen dem Maximum und Minimum der x- und y-Koordinaten des Scatterplots. x_{csv} und y_{csv} sind die Koordinaten der Punktobjekte aus der csv-Datei.

Damit die Antwortzeiten der Nutzer aufgenommen werden können, was für die Eyetracking-Studie geplant wurde, wurde zusätzlich ein Timer in die Scatterplots integriert. Dafür wird im Sketchbook-Ordner von Processing eine Textdatei *Zeitmessung* erstellt, die alle Zeiten in Millisekunden beinhaltet. Dabei wird eine Zeiteinheit in diese Datei eingetragen, sobald der Nutzer den Timer ausschaltet. Der Timer kann mit der Taste '1' ein- und ausgeschaltet werden. Listing 6.3 zeigt die Umsetzung des Timers. Sobald der Timer ausgeschaltet ist, erscheint ein weißes Bild. Wenn der Timer eingeschaltet wird, ist das Scatterplot wieder ersichtlich. Die Textdatei beinhaltet weiterhin einen Zeitstempel, wann der Timer eingeschaltet und zurückgestellt wurde.

Die Beschriftung der Punktobjekte kann statisch oder dynamisch angezeigt werden. In den nächsten Abschnitten werden beide Möglichkeiten vorgestellt.

Listing 6.3 Timer in Scatterplot

```
PrintWriter output_zeit;

void setup(){
  output_zeit = createWriter("Zeitmessung.txt");
}

void keyPressed() {
  if ((key == '1') && (button == true)) {
    passed_Counter = millis() - saved_Counter;
    output_zeit.println(passed_Counter);
    output_zeit.flush();
    button = false;
  }

  else if ((key == '1') && (button == false)) {
    saved_Counter = millis();
    output_zeit.println("Timer zurueckgesetzt.");
    button = true;
  }
}
```

6.4 Dynamisches Labeling mit Excentric Labeling

Die Vertically Coherent Labeling Variante des Excentric Labeling Algorithmus (siehe Kapitel 4.2.7) wurde sowohl für die Umsetzung der dynamischen Technik mit dem Eyetracker als auch für die Umsetzung der dynamischen Technik mit der Maus eingesetzt. Allerdings wurden teilweise Veränderungen vorgenommen. Um die Augkoordinaten aufzunehmen, wurde für die Eyetracker-Technik PEEP eingebunden. Das Einbinden wurde bereits in Kapitel 6.2.2 gezeigt.

Der Algorithmus wird in die draw()-Funktion von Processing eingebaut, da der Fokusbereich eines Nutzers ständig aktualisiert werden muss. Dadurch können dynamisch die Beschriftungen ein- und ausgeblendet werden, wenn der Fokus sich ändert. Algorithmus 6.1 zeigt die einzelnen Schritte von Excentric Labeling, die für jede Aktualisierung notwendig sind. Dieser Algorithmus orientiert sich allerdings an die Technik mit der Maus. Für die Eyetracking-Technik müssen mouseX und mouseY lediglich durch et.x und et.y ersetzt werden.

Zu Beginn werden alle Punktobjekte innerhalb der Fokussierung selektiert. Die Fokussierung stellt dabei einen Kreis mit einem Radius *radius* von 35 Pixel dar. Die Überprüfung, ob Punktobjekte mit den Koordinaten *x_pos* und *y_pos* im Kreis enthalten sind, erfolgt mit Hilfe des Euklidischen Abstands. Dabei wird kontrolliert, ob die Distanz *disX* und *disY* zwischen den Punktkoordinaten und den Mauskoordinaten mouseX und mouseY bzw. den Augkoordinaten et.x und et.y kleiner ist als der Radius. Damit die vertikale Reihenfolge erfüllt ist, werden die Punktobjekte nach ihrer y-Koordinate sortiert. Anschließend wird die Anzahl an Punktobjekten *count* innerhalb der Fokussierung gezählt. In Abhängigkeit von *count* werden die Punktobjekte entweder links, rechts oder oberhalb des Punktobjekts zugeordnet. Falls im

Algorithmus 6.1 Excentric Labeling

```

radius ← 35
disX ← x_pos - mouseX
disY ← y_pos - mouseY
draw_left ← 0
draw_right ← 0
for all Features do
  if sqrt(sq(disX) + sq(disY)) < radius then
    sort(y_pos of Features)
    count ← Number of Features
    if (x_pos < mouseX) && (count > 1) then
      draw_left ← draw_left + 1
      text(x_pos, y_pos * draw_left)
      line(x_pos, y_pos, x_pos, y_pos * draw_left)
    else if ((x_pos > mouseX) && (count > 1)) then
      draw_right ← draw_right + 1
      text(x_pos, y_pos * draw_right)
      line(x_pos, y_pos, x_pos, y_pos * draw_right)
    else if (count == 1) then
      text(x_pos, y_pos)
    end if
  end if
end for

```

Fokus mehr als ein Punktobjekt vorhanden ist, werden die fixierten Punkte auf der linken Seite des Mauszeigers bzw. der Augkoordinaten links zugeordnet, ansonsten rechts. Damit die Beschriftungen auf beiden Seiten unter- und nicht aufeinander erscheinen, werden zwei Zählervariablen *draw_left* und *draw_right* benutzt, welche die Höhe der einzelnen Beschriftungen auf jeweils beiden Seiten festlegt. Da die Beschriftungen links- bzw. rechtsbündig sein sollen, wird jeweils auf beiden Seiten die x-Koordinate aller Beschriftungen gleich gesetzt. Falls nur ein Punktobjekt im Fokus vorhanden ist, wird dieser keiner Seite zugeordnet und die Beschriftung wird direkt oberhalb des Punktes ohne Verbindungslinie gesetzt. Mit Hilfe einer *text()*-Funktion werden die Beschriftungen angezeigt. Eine *line()*-Funktion verbindet die Punktobjekte mit ihren entsprechenden Beschriftungen durch eine Linie.

Abbildung 6.3 zeigt ein endgültiges Ergebnis sowohl für die dynamische Beschriftung mit der Maus auf der linken Seite als auch für die dynamische Beschriftung mit dem Eyetracker auf der rechten Seite. Da Beschriftungen mit Hilfe von Verbindungslinien angezeigt werden, falls im Fokusbereich mehr als ein Punktobjekt vorhanden ist, entstehen keine Überlappungen. Abbildung 6.4 zeigt, wie Beschriftungen mit der Maus und mit dem Eyetracker ausgelöst werden, wenn nur ein Punktobjekt im Fokus liegt. Der Unterschied zwischen beiden Techniken ist, dass der Fokusbereich der einen Technik mit der Maus und der Fokusbereich der anderen Technik mit den Augen selektiert wird. Zudem wird für die Technik mit der Maus der Fokusbereich durch einen roten Kreis dargestellt. Die Technik mit dem Eyetracker

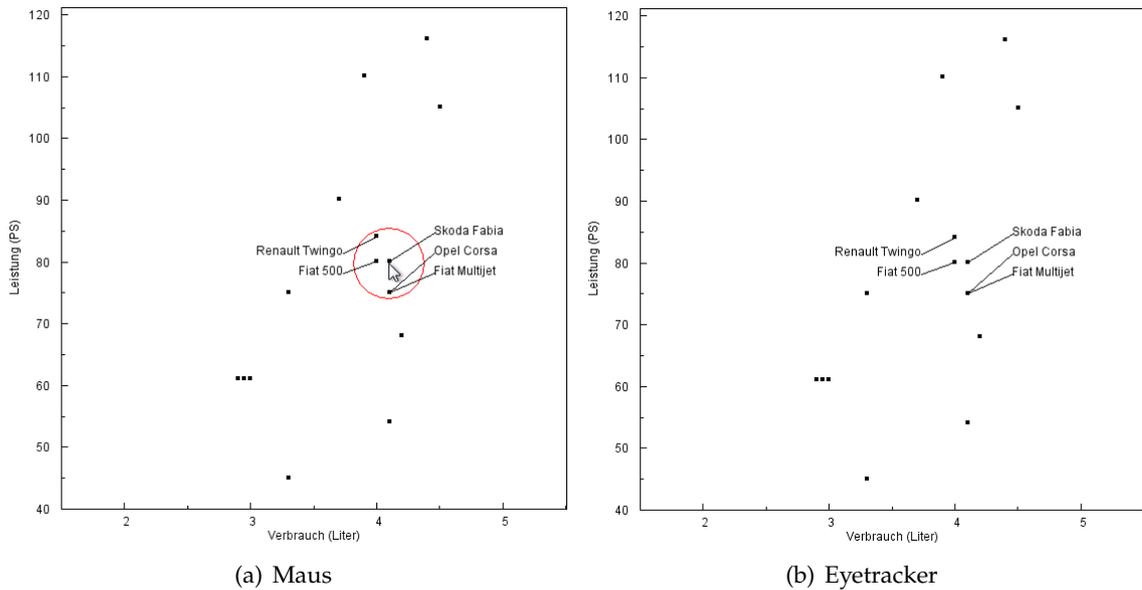


Abbildung 6.3: Scatterplot mit dynamischer Beschriftung

zeigt dagegen keinen Fokusbereich an. Ein weiterer wichtiger Unterschied ist, dass für die Eyetracking-Technik zusätzlich die Leertaste betätigt werden muss, um Beschriftungen lesen zu können. Denn während des Lesens bewegen sich die Augen in Richtung Beschriftung und ändern somit den Fokus. Durch das Drücken der Leertaste wird das Bild eingefroren, indem die `draw()`-Funktion von Processing durch ein `noloop()` nicht mehr durchlaufen wird. Sobald die Leertaste wieder losgelassen wird, ist das Bild durch ein `loop()` nicht mehr eingefroren. Die `draw()`-Funktion läuft dann wieder und aktualisiert ständig das Bild, sodass der Nutzer einen neuen Fokus wählen kann. Listing 6.4 zeigt, wie ein `loop()` und ein `noloop()` in Processing durch Betätigung der Leertaste gesteuert werden kann. Diese Funktion fällt für die Technik mit der Maus weg, da der Fokus der Augen und die Position der Maus verschieden sein können.



Abbildung 6.4: Dynamische Beschriftung für ein Punktobjekt im Fokus: Maus (links) und Eyetracker (rechts)

Listing 6.4 Processing loop() und noloop()

```
void keyPressed() {  
  if (key == ' ') {  
    noLoop();  
  }  
}  
  
void keyReleased() {  
  loop();  
}
```

6.5 Statisches Labeling mit Simulated Annealing

Für die Umsetzung der statischen Technik wurde der Simulated Annealing Algorithmus eingesetzt. Das 4-Positionen-Modell wurde dabei als Basis für die möglichen Positionen der Beschriftungen verwendet. Der Algorithmus kann in vier Komponenten gegliedert werden: Die Wahl einer Anfangskonfiguration, einer geeigneten Objective Function, einer Konfigurationsänderung und einem Annealing Schedule [CMS95].

Anfangskonfiguration

Bei der initialen Auswahl einer Beschriftungsposition werden alle Beschriftungen zufällig und ohne Rücksicht auf Überlappungen auf eine mögliche Position platziert. Die Piggyback-Methode bietet eine Alternative zur zufälligen Platzierung, wobei hier das Simulated Annealing als Nachbearbeitung für einen anderen Algorithmus eingesetzt wird. Die zweite Möglichkeit liefert jedoch keine bessere Qualität des Ergebnisses und auch keine schnellere Laufzeit des Algorithmus.

Objective Function

Die Wahl einer Objective Function beeinflusst die Ästhetik der Beschriftungsplatzierung, die Qualität des Ergebnisses und die Effizienz der Suche nach geeigneten Beschriftungspositionen. Da Simulated Annealing eine statistische Methode ist, die auf einer hohen Anzahl an Bewertungen beruht, sollte die Objective Function so gewählt werden, dass die Bewertung einer Verbesserung durch das Repositionieren einer Beschriftung an eine andere zufällige Position schnell erfolgt. In dieser Arbeit gibt die Objective Function die Anzahl an Überlappungen zwischen der Beschriftung eines Punktobjekts mit den restlichen Beschriftungen zurück. Die zufällig ausgewählten Beschriftungspositionen werden in einem Array gespeichert. Wenn eine Beschriftung auf eine neue Position umplatziert wird, dann wird die neue Position in das Array eingetragen. Die alte Beschriftungsposition wird vorher temporär in eine Variable gespeichert, um eine Repositionierung wieder rückgängig machen zu können. Dann wird mit einer Funktion überprüft, ob und wie viele Beschriftungen auf der neuen

Algorithmus 6.2 Simulated Annealing

```
temp ← 1.0
counter ← 1
no_replace ← 1
replace ← 1
while (counter ≤ 50) && (no_replace ≤ 20n) do
  l ← random label
  Oold ← ObjFunction(l)
  pos ← new random position of l
  Onew ← ObjFunction(l)
  if (replace % 5n = 0) then
    temp ← temp * 0.9
    counter ← counter + 1
    replace ← 1
    no_replace ← 1
  end if
  replace ← replace + 1
  if (Onew > Oold) then
    if rand(0,1) > 1.0 - expp, where p = (Oold - Onew)/temp then
      do nothing
    else
      reverse changes to label l
      no_replace ← no_replace + 1
      replace ← replace - 1
    end if
  end if
end while
```

Position überlappen. Listing 6.5 zeigt, wie in Processing überprüft werden kann, ob zwei Beschriftungsboxen kollidieren. Die ersten vier Parameter definieren die x-Koordinate, die y-Koordinate, die Breite und die Höhe der ersten Beschriftungsbox. Die restlichen Parameter definieren analog die zweite Beschriftungsbox. Wenn zwei Beschriftungsboxen sich überlappen, wird der boolesche Wert *true* ausgegeben. Da die Beschriftungsboxen iso-orientiert sind, also da alle Rechteckseiten der Beschriftungsboxen parallel zu den Koordinatenachsen verlaufen, kann eine Überschneidung mit Hilfe aller Kanten der Beschriftungsboxen getestet werden. Sobald Überlappungen festgestellt werden können, wird die lokale Anzahl an Überlappungen einer Beschriftung summiert.

Konfigurationsänderung

Bei der Konfigurationsänderung wird festgelegt, wann eine neue Position der Beschriftung übernommen wird. Sobald eine Verbesserung der Objective Function eintritt, wird die neue Position sofort übernommen. Das ist genau dann der Fall, wenn die Objective Function einen

Listing 6.5 Überprüfung auf Überlappungen in Processing

```

boolean LabelIntersect(float ax0, float ay0, float ax1, float ay1, float bx0, float by0,
    float bx1, float by1) {
    float topA = min(ay0, ay1);
    float botA = max(ay0, ay1);
    float leftA = min(ax0, ax1);
    float rightA = max(ax0, ax1);
    float topB = min(by0, by1);
    float botB = max(by0, by1);
    float leftB = min(bx0, bx1);
    float rightB = max(bx0, bx1);

    if (botA < topB || botB < topA || rightA < leftB || rightB < leftA) {
        return (false);
    }
    else {
        return (true);
    }
}

```

kleineren Wert zurückliefert. Wenn eine Verschlechterung eintritt, wird die neue Position mit der Wahrscheinlichkeit $p = 1.0 - e^{(\Delta E / \text{Temperatur})}$ nicht akzeptiert. Das ΔE steht dabei für die Differenz zwischen der alten und der neuen Objective Function, wie in Algorithmus 6.2 dargestellt ist. Die Temperatur liegt zu Beginn bei 1.0 und nimmt pro Iteration der while-Schleife ab, bis sie einen Wert von 0.0 hat. Bei einer hohen Temperatur ist dabei die Wahrscheinlichkeit höher, eine Verschlechterung zuzulassen als bei einer niedrigeren Temperatur. Das liegt daran, da mit einer sinkenden Temperatur die Wahrscheinlichkeit p gegen Eins geht. Am Ende hat die Wahrscheinlichkeit p nur noch einen Wert von 1.0. Mit Hilfe einer reellen Zufallszahl aus dem Intervall $[0,1)$ wird letztendlich entschieden, ob eine Verschlechterung zugelassen wird oder nicht. Somit werden zu Beginn eher Verschlechterungen zugelassen und gegen Ende nur noch Verbesserungen akzeptiert.

Annealing Schedule

Das Annealing Schedule zeigt, wie die Temperatur linear gesenkt wird. Der initiale Wert der Temperatur liegt bei 1.0. Wenn 5n Beschriftungen neu positioniert wurden, dann wird die Temperatur um 10% gesenkt und ein Temperaturzähler um Eins erhöht. n steht für die Anzahl der zu beschrifteten Punktobjekte. In Algorithmus 6.2 ist dieser Ablauf dargestellt: Die Anzahl an Neuplatzierungen wird in die Variable *replace* und die Anzahl an Neuplatzierungen, die rückgängig gemacht wurde, wird in die Variable *no_replace* gespeichert. Die Temperatur *temp* wird mit dem Faktor 0.9 multipliziert, also um 10 Prozent gesenkt und *replace* und *no_replace* werden wieder auf Eins gesetzt und der Temperaturzähler *counter* um Eins erhöht. *Replace*, *no_replace* und *counter* werden jeweils mit dem Wert 1 initialisiert. Für jede Temperatur *temp* werden maximal 20n Beschriftungen neu positioniert. Dieser Prozess wird für maximal 50 Temperaturstufen *counter* wiederholt. Wenn für dieselbe Temperatur

6 Implementierung

temp nach 20n Schritten keine Neuplatzierung stattfindet, dann stoppt der Algorithmus mit den aktuellsten Beschriftungspositionen.

Abbildung 6.5 zeigt ein beispielhaftes Ergebnis für eine statische Beschriftung der Punktobjekte, das mit Hilfe des Simulated Annealing Algorithmus erzeugt wurde. Die Anzahl an Überlappungen wurde minimiert. Nur für niedrige x- und y-Variablenwerte sind innerhalb der positiven Korrelation Überlappungen vorhanden.

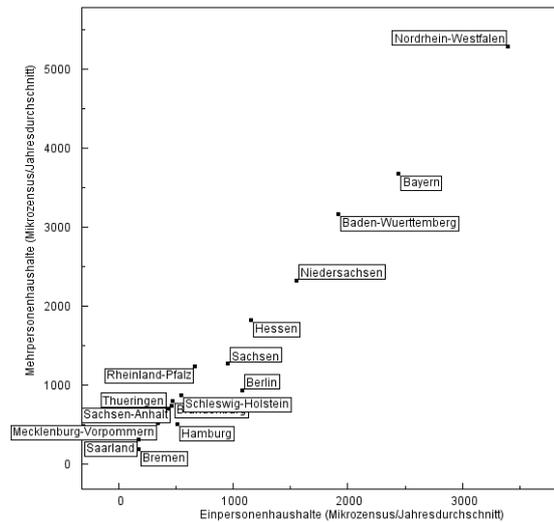


Abbildung 6.5: Scatterplot mit statischer Beschriftung

7 Eyetracking-Studie

In diesem Kapitel wird auf die Durchführung der Eyetracking-Studie eingegangen. Das Ziel dieser Studie besteht darin, die neue dynamische Eyetracking-Technik mit der Maus-Technik und der statischen Technik, deren Implementierungen im letzten Kapitel vorgestellt wurden, zu vergleichen, um anschließend folgende Hypothesen zu beweisen oder zu widerlegen:

- **Vergleich der Zeit**

Bei der Suche nach Annotationen von Punktobjekten in Scatterplots liefert die neue dynamische Eyetracking-Technik schnellere Ergebnisse als die Maus und die statische Technik.

- **Vergleich der Genauigkeit**

Bei der Suche nach Annotationen von Punktobjekten in Scatterplots liefert die neue dynamische Eyetracking-Technik eine höhere Genauigkeit als die Maus und die statische Technik.

Der Ablauf der Studie lässt sich in vier Abschnitte untergliedern: Im ersten Abschnitt werden geeignete Datensätze gesammelt, die in der Pilotstudie und in der finalen Studie als Stimuli eingesetzt werden. Anschließend werden für die Datensätze passende Aufgaben vorbereitet, die ebenfalls in der Pilotstudie und der finalen Studie den Probanden gestellt werden. Die Pilotstudie dient zur Bestimmung von Parametern und zur Erprobung des Studienablaufs, um Fehler beim Entwurf aufzudecken. In der finalen Studie wird die neue dynamische Eyetracking-Technik mit der Maus-Technik und der statischen Technik verglichen. Nach der Studie werden die aufgenommenen Daten ausgewertet.

7.1 Studienkonzept

In diesem Abschnitt wird der Entwurf der Eyetracking-Studie beschrieben, der die grobe Struktur der Studie festlegt.

Zu Beginn wurde festgelegt, welche Technik einem Probanden zugeteilt wird. Es gab die Überlegung jedem Probanden nur eine Technik zuzuordnen. Letztendlich fiel die Entscheidung aber darauf, jedem Probanden jede Technik zuzuteilen.

Da der Einsatz von Testdaten notwendig war, wurden einige Datensätze als Stimuli für die Studie ausgewählt. Zu den Testdaten wurden zudem Aufgabenstellungen nach dem Lokalisierungsproblem vorbereitet, die den Probanden während der Studie gestellt werden sollten. Die Reihenfolge der Techniken und Stimuli sollte vor der Studie nach dem *Latin*

Square generiert werden [DK91], um einen direkten Vergleich zu ermöglichen und Lern-, Ermüdungs- sowie Reihenfolgeeffekte zu vermeiden. Lerneffekte liegen dann vor, wenn mehrere Aufgaben hintereinander gelöst werden. Die endgültige Anzahl an Stimuli und Aufgabenstellungen wurde erst nach der Pilotstudie festgelegt.

Die Studie sollte zwei Ergebnisse liefern: Die Zeit, die vergeht, bis der Proband die Lösung findet und die Genauigkeit der Lösung. Die Techniken sollten also nach der Schnelligkeit der Antwortzeiten und nach der Exaktheit der Ergebnisse miteinander verglichen werden. Die Antwortzeiten der Probanden wurden in Textdateien aufgenommen, die mit Hilfe eines integrierten Timers in allen drei Techniken gespeichert werden können. Der Timer sollte durch eine beliebige Taste ein- und ausgeschaltet werden. Diese Taste wurde kurz vor der Pilotstudie festgelegt. Die Aufnahme der Exaktheit der Ergebnisse sollte während der Studie protokolliert werden. Die Genauigkeit der Antworten sollten prozentual angegeben werden. Für die Teilnahme an der Studie wurden zwischen 15 und 25 Teilnehmer geplant. Die Probanden mussten keine Vorkenntnisse oder Qualifikationen aufweisen. Keine Vortests wurden geplant und es wurden keine besonders gute bzw. schlechte Leistungen der Probanden erwartet.

Um Sehbeschwerden zu identifizieren, wurde ein Sehtest geplant. Durch diesen Test sollten notwendige Fähigkeiten der Versuchspersonen festgestellt werden. Wichtig bei der Durchführung eines Sehtests ist, die Buchstaben mit der richtigen Distanz auf Augenhöhe der Teilnehmer darzustellen. Falls mögliche Hinweise auf eine Sehschwäche erkannt wurden, sollte der Proband nicht darüber informiert werden und die Studie zu Ende führen. Erst nach der Studie sollte entschieden werden, ob das Ergebnis verworfen wird oder nicht. Für das Testen des Sehvermögens wurde das *Snellen Eye Chart* vom niederländischen Augenarzt Herman Snellen verwendet [Kau07].

7.2 Testdaten und Aufgabenstellungen

In der Studie ist der Einsatz von Testdaten notwendig, die als Stimuli eingesetzt werden. Es wurden insgesamt 19 Datensätze gesammelt. Die Verteilung der Punktobjekte erfüllen dabei mindestens eine der folgenden Eigenschaften, die bereits in Kapitel 3.4 diskutiert wurden:

- **Cluster**

Der Datensatz enthält einen Bereich, in denen die Punktobjekte nahe beisammen liegen, sodass sie in Gruppen zusammengefasst werden können.

- **Korrelation**

Eine Korrelation stellt einen linearen Zusammenhang zwischen zwei metrischen Variablen dar. Die Verteilung der Punktobjekte entspricht entweder einer positiven oder einer negativen Korrelation.

- **Ausreißer**

Der Datensatz enthält mindestens ein Punktobjekt, das auffällig weit von der Masse entfernt ist.

Für diese Datensätze wurden Aufgaben auf Basis des Lokalisierungsproblems formuliert, sodass Punktbeschriftungen für bestimmte Variablenwerte im Scatterplot zurückgegeben werden sollen. Das bedeutet, eine bestimmte Menge an Punkten im Scatterplot soll in Abhängigkeit der Achsenwerte gefiltert und anschließend deren Beschriftung identifiziert werden. Beispielsweise sollen Bezeichnungen von Punktobjekten innerhalb eines Clusters bestimmt werden. Für jeden Datensatz wurden drei Fragen vorbereitet. Die erste Aufgabe dient dabei als Testfrage, die nicht bewertet wird. Nur die restlichen zwei Fragen werden ausgewertet. Eine dieser Fragen konzentriert sich dabei auf die Lokalisierung einer Beschriftung und die andere Frage auf die Lokalisierung mehrerer Beschriftungen. Das bedeutet, dass sich eine Frage auf einen Cluster bezogen hat und die andere Frage auf einen Ausreißer oder auf ein Minimum bzw. Maximum einer Korrelation. Die Begriffe Cluster, Korrelation und Ausreißer traten in den Aufgabenstellungen nicht auf, sondern wurden mit Hilfe der Achsenwerte umschrieben, um eindeutige Antworten zu definieren.

7.3 Pilotstudie

Die Pilotstudie dient dem Erkennen von Mängeln des Studien-Entwurfs und zur Behebung von Fehlern. Aufgrund dessen wird die Pilotstudie vor der finalen Studie durchgeführt.

An der Pilotstudie nahmen fünf Probanden teil. Davon waren drei Probanden wissenschaftliche Mitarbeiter des Visualisierungsinstituts der Universität Stuttgart. Die restlichen zwei Probanden waren Studenten der Universität Hohenheim. Für jeden Proband wurden sechs Stimuli mit jeweils drei bis fünf Fragen vorbereitet.

Die Pilotstudie wurde allerdings hauptsächlich zur Bestimmung von Parametern für die umgesetzten Techniken durchgeführt. Folgende Parameter wurden getestet:

Größe der Fokussierung

Ein geeigneter Radius für die Fokussierung in der dynamischen Eyetracking-Technik musste bestimmt werden. Dafür wurde das Mausrad eingesetzt, um den Kreis, der die Fokussierung darstellt, zu verkleinern oder zu vergrößern. Da für die Eyetracking-Technik die Fokussierung nicht angezeigt wird, konnte zusätzlich mit einer Maustaste der Kreis ein- und ausgeblendet werden. Durch diese Maus-Events konnten somit während der Pilotstudie verschiedene Radien des Kreises eingestellt werden. Die Aufgabe der Probanden lautete, alle Punktobjekte für zwei verschiedene Datensätze mit verschiedenen Radien zu treffen. Anfangs wurde ein Radius von 20 Pixel gewählt. Falls nicht alle Punktobjekte unmittelbar selektiert werden konnten, wurde der Radius um 5 Pixel erhöht bis alle getroffen wurden. Das Ergebnis ergab, dass ein Radius von 35 Pixel geeignet ist. Alle Probanden konnten mit diesem Radius alle Punktobjekte sofort selektieren.

Automatische Verkleinerung der Fokussierung

Um nicht zu viele Beschriftungen zugleich einzublenden, sollte eine automatische Verkleinerung der Fokussierung in der dynamischen Technik mit dem Eyetracker getestet werden. Dabei sollten die Probanden ebenfalls alle Punktobjekte von zwei Datensätzen unmittelbar selektieren. Der Kreis verkleinerte sich genau dann, wenn über fünf Punktobjekte im Fokus lagen. Der Default-Wert lag bei 35 Pixel. Bei einer Verkleinerung betrug der Radius 20 Pixel. Durch die automatische Verkleinerung ergab sich jedoch das Problem, dass nicht mehr alle Punktobjekte unmittelbar selektiert werden konnten, da der Eyetracker nur in groben Bereichen und sprunghaft gut arbeitet. Erst wenn die Augen einen anderen Bereich fixiert und anschließend dasselbe Objekt wieder selektiert hatten, wurde der Punkt getroffen. In der Zwischenzeit hat sich jedoch die Größe der Fokussierung geändert, sodass eine Verkleinerung nicht mehr notwendig war. Nur bei einem Probanden funktionierte die Verkleinerung mit einem bestimmten Datensatz. Die Entscheidung ist trotzdem darauf gefallen, eine automatische Verkleinerung in der Studie nicht einzusetzen.

Beschriftungsboxen

Da das Anzeigen der Beschriftungsboxen für die statische Technik eine subjektive Entscheidung ist, wurden die Probanden während der Pilotstudie gefragt, welche Version (mit oder ohne Beschriftungsboxen) sie besser finden. Mit einer Maustaste konnten die Beschriftungsboxen ein- und ausgeblendet werden, sodass die Probanden einen direkten Vergleich hatten. Für jeden Probanden wurden dieselben zwei Stimuli eingesetzt. In acht Fällen haben sich die Probanden für und in zwei Fällen gegen die Beschriftungsboxen entschieden. Die Probanden, die sich für die Beschriftungsboxen entschieden haben, empfanden die Zuordnung von Objekt und Beschriftung durch die Boxen eindeutiger. Da sich die Mehrzahl für die Boxen entschieden hat, wurden auch in der finalen Studie die Boxen eingesetzt.

7.4 Finale Studie

Nach der Pilotstudie fand kurz darauf die finale Studie statt. Die festgelegten Parameter aus der Pilotstudie wurden für die finale Studie übernommen. Im Folgenden wird näher beschrieben, wie die Studie verlief.

Probanden

An der Studie nahmen 7 weibliche und 12 männliche Probanden teil. Der jüngste Proband war 17 Jahre und der älteste Proband war 47 Jahre alt. Das Alter der restlichen Probanden lag zwischen 20 und 30 Jahren. Zum Zeitpunkt der Studie hatten 4 Probanden ihr Studium erfolgreich abgeschlossen. Einer davon hat promoviert und drei davon waren Informatiker oder Softwaretechniker. Von den restlichen 15 Probanden waren 11 Probanden Studenten,

davon haben 5 Informatik an der Universität Stuttgart studiert. Insgesamt trugen 7 Probanden eine Sehhilfe, 4 davon trugen Brillen und 3 Kontaktlinsen. Erfahrungen mit Eyetracking hatten 4 Probanden und 2 mit Scatterplots.

Umgebung

Die Studie fand im Projektlabor 00.023 des Visualisierungsinstitutes der Universität Stuttgart statt. Während des gesamten Studienablaufs waren die Fenster verschlossen und alle Rollläden herunter gelassen, um Störungen zu vermeiden. Der Raum war künstlich beleuchtet und unnötige Gegenstände wurden entfernt. An der Tür des Labors hing ein Hinweiszettel mit der Aufschrift: „Eyetracking-Studie – Bitte nicht stören!“. Diese Maßnahmen sollten eine ruhige und störungsfreie Umgebung schaffen. Vor Beginn des Studienablaufs wurden die Probanden gebeten, ihre Mobiltelefone auszuschalten.

Die Studie wurde am Eyetracker-System Tobii T60 XL mit dem Betriebssystem Windows 7 Professional durchgeführt. Die Auflösung des Bildschirms betrug 1920×1200 Pixel. Da die Auflösung der Scatterplots 700×700 Pixel betrug, wurde vor Beginn der Studie der Hintergrund des Bildschirms auf einen einfarbigen blauen Farbton gesetzt, um die Blickrichtung nicht außerhalb des Fensters zu lenken.

Stimuli

Nach der Pilotstudie wurde die endgültige Anzahl an Datensätzen festgelegt. In der finalen Studie wurden sechs verschiedene Datensätze untersucht, die nicht in der Pilotstudie eingesetzt wurden. Die ausgewählten Datensätze entsprachen den geforderten Eigenschaften (Cluster, Korrelation, Ausreißer) und wurden auf die drei Techniken gleichmäßig verteilt. Abbildung 7.1 zeigt zwei verschiedene Scatterplots. Die Verteilung der Datenpunkte des linken Scatterplots entsprechen einer positiven Korrelation. Das Scatterplot auf der rechten Seite erfüllt sogar zwei Eigenschaften, da sowohl ein Cluster als auch zwei Ausreißer erkannt werden können. Der Cluster befindet sich innerhalb des Kreises und die zwei Ausreißer sind außerhalb davon.

Für jede Technik wurden genau zwei Datensätze eingesetzt. Zu jedem Datensatz wurden genau drei Aufgaben den Probanden gestellt, wobei nur zwei davon ausgewertet wurden. Die erste Aufgabe diente zur Übung. Jeder Proband erhielt Aufgaben zu jeder Technik, sodass jeder Proband insgesamt 18 Aufgaben beantworten musste, wovon 12 ausgewertet wurden. Die Reihenfolge der Techniken und der Datensätze wurden nach dem *Latin Square* rotiert, um Lern- und Reihenfolgeeffekte zu verhindern und um für den Vergleich der Techniken gleiche Bedingungen zu schaffen. Die drei Techniken können nur miteinander verglichen werden, wenn sie auf demselben Datensatz basieren. Ein Latin Square ist eine $n \times n$ Matrix der Ordnung n mit n verschiedenen Symbolen, sodass jedes Symbol genau einmal in jeder Zeile und Spalte auftritt [DK91]. Sudoku entspricht einem Spezialfall davon, das aus neun 3×3 Feldern besteht. Da drei Techniken zu vergleichen sind, wird in dieser Arbeit ein 3×3 Feld eingesetzt. Abbildung 7.2 zeigt die Reihenfolge der Techniken für

7 Eyetracking-Studie

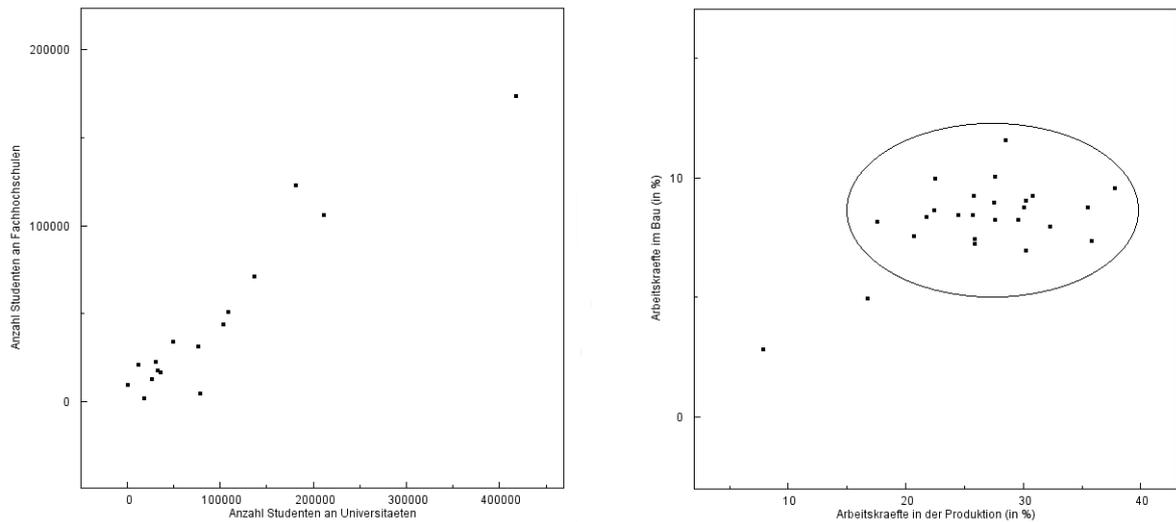


Abbildung 7.1: Korrelation, Ausreißer und Cluster in Scatterplots

die ersten drei Probanden. Eine Zeile entspricht einem Proband und eine Spalte entspricht zwei Datensätzen. Das bedeutet, dass der erste Proband mit der Maus-Technik begonnen hat, anschließend wurde die Eyetracking-Technik eingesetzt und zum Schluss wurde ihm die statische Technik zugewiesen. Alle drei Probanden hatten dieselbe Reihenfolge der Datensätze. Die Reihenfolge der Datensätze wurde erst nach dem Latin Square rotiert, wenn alle möglichen Reihenfolgen der Techniken durchgeführt wurden. Da die Permutation der drei Techniken $3! = 6$ entspricht, wurden die Datensätze immer nach sechs Probanden rotiert.

	D1/D2	D3/D4	D5/D6
P1	Maus	Eyetracker	Statisch
P2	Eyetracker	Statisch	Maus
P3	Statisch	Maus	Eyetracker

Abbildung 7.2: Reihenfolge der Techniken nach dem Latin Square

Studienablauf

Zu Beginn der Eyetracking-Studie haben die Probanden die Aufklärung zum Ablauf der Studie durchgelesen und unterschrieben. Sie wurden darauf hingewiesen, dass sie jederzeit das Recht haben, die Studie abubrechen und dass ihre Daten anonym ausgewertet werden. Anschließend haben die Probanden einen Fragebogen mit persönlichen Daten ausgefüllt. Danach wurde ein Sehtest durchgeführt, um zuverlässige Ergebnisse zu erzielen. Ein Farbttest war nicht notwendig, da für die drei Techniken keine Farben eingesetzt wurden. Bevor den Probanden Aufgaben gestellt wurden, erfolgte eine kurze Einführung, die in Textform ausgedruckt war. Dadurch konnte sichergestellt werden, dass alle Probanden dieselben Informationen während der Einführung erhielten. Zu Beginn der Einführung wurden Scatterplots sowie die drei zu vergleichenden Techniken erklärt und jeweils ein Beispieldatensatz mit einer Beispielaufgabe aufgeführt. Die Beispielaufgabe sollten die Probanden beantworten, um zu testen, ob sie die verschiedenen Techniken verstanden haben. Da die Fragen sich ausschließlich auf das Lokalisierungsproblem mit Hilfe der Scatterplot-Achsen eingeschränkt hatten, war eine Aufklärung von Cluster, Korrelation und Ausreißer nicht notwendig, da diese in den Fragestellungen umschrieben wurden. Nach der Einführung begann der eigentliche Ablauf der Studie. Jede Aufgabenstellung wurde auf Papier gedruckt, sodass der Proband die Aufgabe in Ruhe durchlesen konnte. Um die Zeit zu erfassen, wie lange ein Proband zum Beantworten einer Frage benötigt, mussten die Probanden den Timer, der in allen drei Techniken integriert ist, mit der Taste '1' starten und beenden. Damit die Stimuli nicht vor oder während dem Lesen der Aufgabenstellung präsentiert wurden, wurde ein weißes Bild angezeigt, solange der Timer nicht gestartet wurde bzw. nach Beenden des Timers. Eine Kalibrierung des Eyetrackers wurde vor den Aufgabenstellungen zur Eyetracking-Technik durchgeführt. Anschließend mussten die Probanden die Aufgaben mündlich beantworten. Während der Durchführung wurde die Exaktheit der Antworten auf einem separaten Bogen notiert. Am Ende der Studie mussten die Probanden eine Liste für die Geldausgabe ausfüllen und unterschreiben. Jeder Proband, mit Ausnahme von Hiwis oder Mitarbeitern der Universität Stuttgart, erhielt daraufhin 10 Euro als Aufwandsentschädigung.

7.5 Statistische Auswertung

Dieser Abschnitt präsentiert die Auswertung und die Ergebnisse der finalen Studie, ohne die Ergebnisse zu interpretieren. Die Auswertung wurde in zwei Analyse-Gruppen unterteilt: Antwortzeiten und Genauigkeitswerte. Nach der Studie wurden alle Zeiterfassungen und Genauigkeitswerte mit den entsprechenden Techniken, Datensätzen und Fragen in zwei getrennte Excel-Tabellen manuell übertragen. Die statistische Auswertung erfolgte unter Verwendung von R [Teao8]. Um auf Varianzen bezüglich der Antwortzeiten und der Genauigkeitswerte zwischen den drei Techniken zu prüfen, wurde die Analyse mit der *One-way ANOVA* für jeweils beide Analyse-Gruppen durchgeführt. Die Antwortzeiten beziehungsweise die Genauigkeitswerte wurden als abhängige Variable und die Techniken als unabhängige Einflussvariable festgelegt. Für die Anwendung der ANOVA sollten die zu untersuchenden

Daten eine Varianzhomogenität und eine nicht stark abweichende Normalverteilung aufweisen. Die Varianzhomogenität ist dann erfüllt, wenn sich die Varianzen der Stichproben in ihrer Größe nicht signifikant unterscheiden. Per *Shapiro-Wilk-Test* kann die Normalverteilung der Daten und mit Hilfe des *Bartlett-Tests* die Varianzhomogenität untersucht werden. Im Folgenden werden die Ergebnisse der beiden Analyse-Gruppen präsentiert.

Zeit

Für die Analyse bezüglich der Zeit wurden die Antwortzeiten in Millisekunden untersucht. Die Antwortzeiten wichen zu Beginn etwas von der Normalverteilung ab. Deshalb wurden sie mit Hilfe des natürlichen Logarithmus transformiert. Abbildung 7.3 zeigt einen *Normal Quantil-Quantil-Plot* mit den kumulativen Häufigkeiten auf der x-Achse und den Antwortzeiten auf der y-Achse. Dieser zeigt, dass eine Normalverteilung vorliegt, da die Punkte sich der blauen Gerade annähern. Per Shapiro-Wilk-Test wurde die Normalverteilung der Daten und mit Hilfe des Bartlett-Tests die Varianzhomogenität untersucht. Beide Tests haben die Eigenschaften für die Antwortzeiten nachgewiesen.

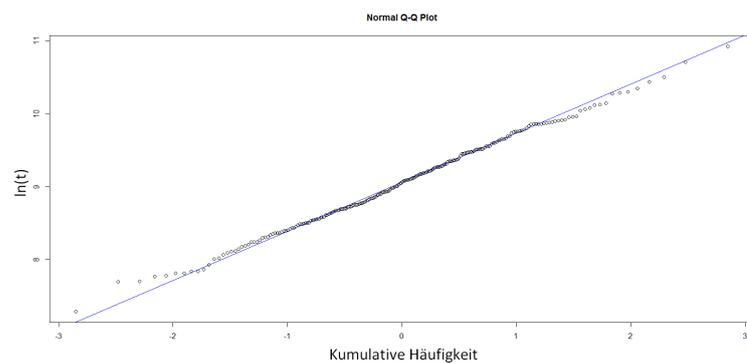


Abbildung 7.3: Q-Q-Plot für Antwortzeiten

Die Antwortzeiten der Probanden wurden in zwei Fragetypen unterteilt: Der erste Fragetyp stellt die Antwortzeiten für die Lokalisierung *eines* Labels dar, der andere für die Lokalisierung *mehrerer* Labels eines Datensatzes. In der Excel-Tabelle wurde dafür eine neue Spalte angelegt, um die Fragetypen unterscheiden zu können. So konnte eine Analyse in Abhängigkeit der Fragen durchgeführt werden.

Die Analyse mit ANOVA wurde für jeweils beide Fragetypen durchgeführt. Da die Analyse gezeigt hat, dass keine signifikanten Unterschiede zwischen den drei Techniken für beide Fragetypen vorhanden sind, wurden die Datensätze innerhalb der Fragetypen separat analysiert. Die Analyse mit ANOVA hat gezeigt, dass signifikante Unterschiede in einem Datensatz vorhanden sind. Da die ANOVA keine Aussage darüber macht, zwischen welchen konkreten Techniken der Unterschied vorliegt, wurde zudem der *Tukey-HSD-Test* durchgeführt. Dieser Test hat gezeigt, dass zwischen der statischen Technik und der Maus-Technik ein

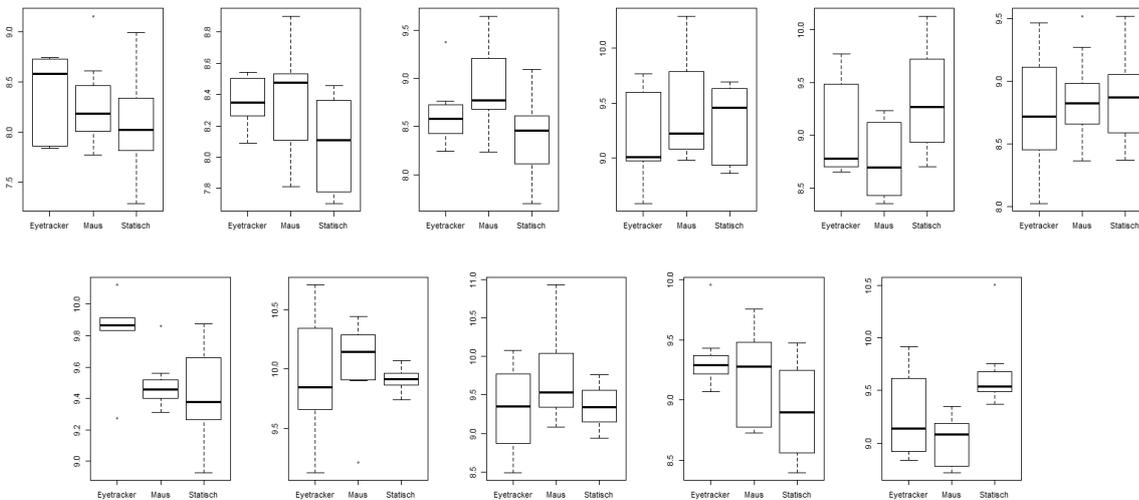


Abbildung 7.4: Boxplots für Antwortzeiten pro Datensatz

signifikanter Unterschied ($p = 13,8094 \times 10^{-3}$) für einen Datensatz (Boxplot in Abbildung 7.4 unten rechts) mit dem Fragetyp „mehrere Labels“ existiert. Für denselben Datensatz (Fünftes Boxplot der oberen Reihe in Abbildung 7.4), aber mit dem Fragetyp „ein Label“ hat sich ein nahezu signifikanter Unterschied ($p = 91,9152 \times 10^{-3}$) zwischen statischer Technik und Maus-Technik ergeben. Des Weiteren hat die ANOVA gezeigt, dass zwischen der statischen Technik und der Eyetracking-Technik ein nahezu signifikanter Unterschied ($p = 51,3159 \times 10^{-3}$) für einen anderen Datensatz (Boxplot in Abbildung 7.4 unten links) mit dem Fragetyp „mehrere Labels“ existiert. In Abbildung 7.4 sind die Boxplots für jeden Datensatz dargestellt, die zwischen dem Fragetyp „ein Label“ und dem Fragetyp „mehrere Labels“ getrennt wurden. Sie zeigen für jeden Datensatz und jeder Technik innerhalb eines Fragetyps die transformierten Antwortzeiten. Die Punkte stellen Ausreißer dar.

Genauigkeit

Für die Analyse bezüglich der Genauigkeit wurden die Genauigkeitswerte prozentual angegeben. Wenn beispielsweise ein Proband zwei Labels lokalisieren sollte, aber nur einen gefunden hat, dann beträgt der Genauigkeitswert 50 Prozent. Die Genauigkeitswerte weichen allerdings ein wenig von einer Normalverteilung ab, auch wenn sie mit dem Logarithmus transformiert werden. Das lässt sich mit den fehlerfreien Antworten begründen: Viele Fragen wurden korrekt beantwortet, sodass viele Genauigkeitswerte mit 0 Prozent vorliegen. Abbildung 7.5 zeigt den Verlauf der Genauigkeitswerte. Auf der x-Achse sind die kumulativen Häufigkeiten abgebildet und auf der y-Achse die Prozentzahl an Fehlern. Nur auf der Höhe von 0 und 25 Prozent nähern sich die Werte nicht der blauen Gerade an. Die ANOVA ist jedoch relativ robust gegenüber Verletzungen der Normalverteilung. Nach Hays

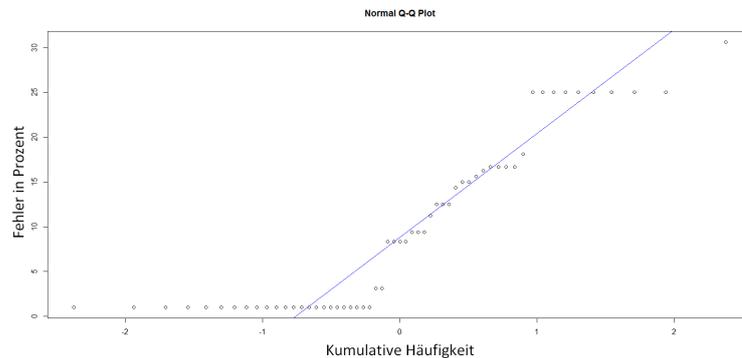


Abbildung 7.5: Q-Q-Plot für Genauigkeitswerte

[Hay73] lässt sich mit dem Zentralen Grenzwertsatz für eine Stichprobenmenge von über 30 eine angenäherte Normalverteilung annehmen. Die Varianzhomogenität wurde mit dem Bartlett-Test gezeigt.

Für die Auswertung mit der ANOVA wurden zu Beginn die Genauigkeitswerte der Probanden über die drei Techniken gemittelt. Das bedeutet, dass ein Durchschnittswert für die Genauigkeit pro Technik für jeden Probanden berechnet wurde. Abbildung 7.6 zeigt einen Boxplot pro Technik mit den Durchschnittswerten pro Technik für die Genauigkeit aller Probanden. Die Ergebnisse der ANOVA haben gezeigt, dass es keine signifikanten Unterschiede gibt. Anschließend wurden die Genauigkeitswerte der Probanden ebenfalls in zwei Fragetypen, „ein Label“ und „mehrere Labels“, unterteilt und für diese jeweils eine ANOVA durchgeführt. Die Ergebnisse ergaben wieder keine signifikanten Unterschiede. Zum Schluss wurden die Genauigkeitswerte der einzelnen Datensätze untersucht, die ebenfalls in „ein Label“ und „mehrere Labels“ separat betrachtet wurden. Das Ergebnis hat endgültig gezeigt, dass für die Genauigkeitswerte keine signifikanten Unterschiede existieren.

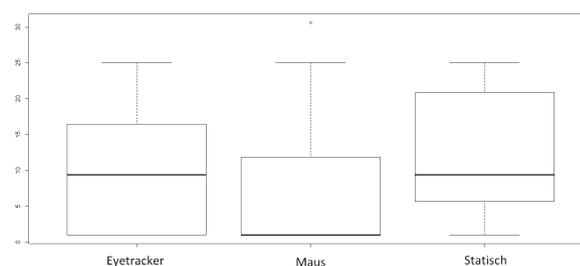


Abbildung 7.6: Boxplots für Genauigkeit pro Technik

8 Schlussfolgerung

In diesem Kapitel werden die Ergebnisse der Eyetracking-Studie kritisch betrachtet und interpretiert. Auf Basis der Ergebnisse der Auswertung werden die Hypothesen bewiesen oder widerlegt und die Ergebnisse diskutiert.

8.1 Überprüfung der Hypothesen

In dieser Diplomarbeit werden zwei Hypothesen aufgestellt, die durch eine Eyetracking-Studie bewiesen oder widerlegt werden. Für die erste Hypothese wird angenommen, dass bei der Suche nach Punkten in Scatterplots anhand des Lokalisierungsproblems, deren Labels dynamisch in einem Scatterplot ein- und ausgeblendet werden, der Eyetracker schnellere Ergebnisse liefert als die Maus und als Techniken, welche die Labels statisch anzeigen. Diese Hypothese kann nicht bewiesen werden.

Die Auswertung hat ergeben, dass der Eyetracker sich nicht signifikant von der Maus und der statischen Technik in Bezug zu den Antwortzeiten bei der Lokalisierung von Labels unterscheidet. Nur für einen Datensatz ergab sich ein nahezu signifikanter Unterschied zwischen dem Eyetracker und der statischen Technik. Allerdings waren für diesen Datensatz die Antwortzeiten mit der statischen Technik sogar etwas schneller als mit dem Eyetracker. In diesem Fall hat sich jedoch die Fragestellung auf die falschen Punktobjekte bezogen, sodass die Vorteile der Eyetracking-Technik gegenüber der statischen Technik nicht ausgenutzt werden konnten. Abbildung 8.1 zeigt diesen Datensatz. Auf der linken Seite wird der Datensatz in einem Scatterplot mit der statischen Technik und auf der rechten Seite mit der Eyetracking-Technik dargestellt. Die Verteilung der Punktobjekte zeigt eine positive Korrelation, wobei die Punkte im unteren Bereich der Korrelation dichter beieinander liegen. Die Fragestellung hat sich auf den Bereich von 100.000 bis 300.000 Studenten an Universitäten (x-Achse der Scatterplots aus Abbildung 8.1) bezogen. In diesem Bereich werden mit der statischen Technik kaum Überlappungen erzeugt, sodass die Labels gut und schnell lesbar sind. Wenn die Fragestellung sich auf den unteren Bereich (unter 100.000) bezogen hätte, dann wäre der Eyetracker sicherlich im Vorteil. In diesem Fall wäre eine Abgrenzung der Punkte mit der Eyetracking-Technik eventuell schneller, da die statisch eingeblendeten Labels das Bild überladen und teilweise Punktobjekte verdecken, sodass eine Abgrenzung der Punktobjekte im dichten Punktebereich nicht mehr möglich ist.

Für die zweite Hypothese wird angenommen, dass bei der Suche nach Punkten in Scatterplots anhand des Lokalisierungsproblems, die neue Eyetracking-Technik eine höhere Genauigkeit

8 Schlussfolgerung

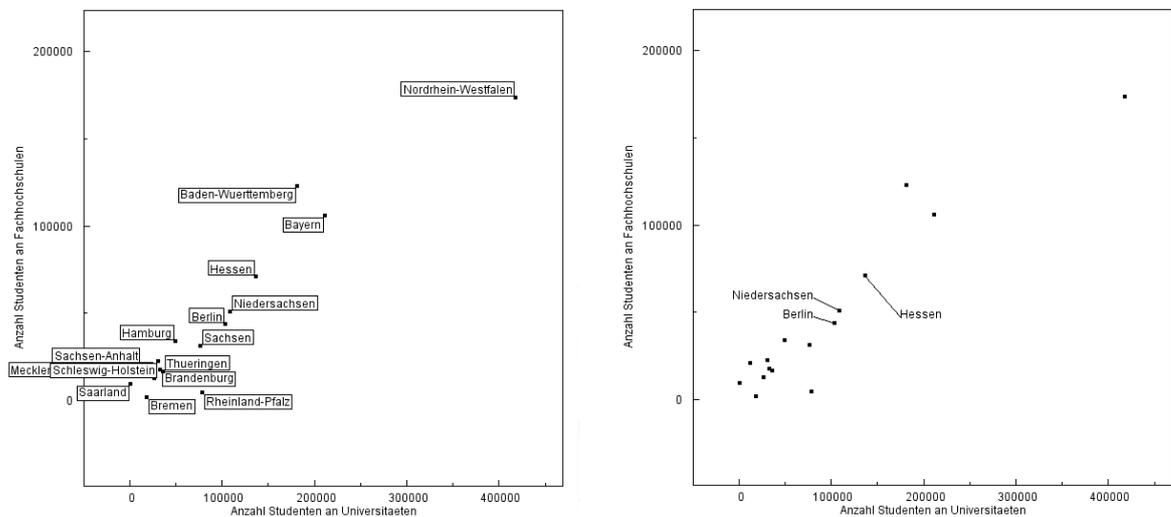


Abbildung 8.1: Scatterplot mit gleichem Datensatz für statische Technik und Eyetracking-Technik

der Ergebnisse liefert als die Maus und die statische Technik. Diese Hypothese kann auch nicht bewiesen werden.

Die Auswertung hat ergeben, dass zwischen den drei Techniken keine signifikanten Unterschiede existieren. Es wurde erwartet, dass vor allem die statische Technik eine niedrigere Genauigkeit liefert, da beispielsweise Datensätze mit Cluster dichte Punktebereiche haben, sodass die statisch angezeigten Labels sich an diesen Stellen teilweise überlappen und unlesbar sind. Wenn die Fragestellung sich auf solche Bereiche bezieht, dann wurde erwartet, dass nicht alle Labels identifiziert werden können und sich somit eine niedrigere Genauigkeit des Ergebnisses ergibt. Leider wurden in der Studie keine passenden Daten geschaffen, sodass der Vorteil für die Eyetracking-Technik nicht sichtbar wird. Allerdings wurde auch erwartet, dass die statische Technik eine hohe Genauigkeit bei der Identifikation von Ausreißern liefert, da in solchen Fällen die Labels lesbar sind. Dadurch wurde die Genauigkeit der statischen Technik erhöht.

8.2 Diskussion über die Ergebnisse der Studie

Die Ergebnisse der Eyetracking-Studie haben sowohl für die Antwortzeiten als auch für die Genauigkeitswerte keine signifikanten Unterschiede zwischen den drei Techniken gezeigt. Viele Faktoren beeinflussen das Ergebnis. Im Folgenden wird erläutert, welche Gründe vermutet werden, wieso die Eyetracking-Technik keine Vorteile erwiesen hat.

Zeitverzögerungen und Einschränkungen der Kopfbewegung beim Eyetracker waren die häufigsten Faktoren, welche die Antwortzeiten der Probanden stark beeinflusst haben. Die Zeitverzögerung des Eyetrackers ist dabei das Hauptproblem, weshalb die Labels nicht sofort

ausgelöst wurden. Da die Maus sehr präzise und ohne Verzögerung arbeitet, konnte die Maus den Eyetracker schnell einholen. Außerdem arbeitet der Eyetracker nur dann sehr schnell, wenn Bereiche sprunghaft fokussiert werden. Wenn allerdings Punktobjekte nacheinander fixiert werden, tritt eine Verzögerung oder teilweise keine Reaktion auf. Das hat zur Folge, dass andere Bereiche betrachtet werden müssen, um sprunghaft Objekte selektieren zu können. Dadurch geht wieder Zeit verloren, sodass der Eyetracker nicht mehr wesentlich schneller arbeitet als die anderen Techniken. Ein weiterer Grund, wieso die Performanz zwischen Eyetracker und Maus gleich ist, ist die Auflösung des Fensters, in denen die Scatterplots für die Techniken dargestellt wurden. Diese beträgt 700×700 Pixel und ist somit relativ klein. Dadurch hat die Maus einen kurzen Weg und benötigt weniger Zeit, um Punktobjekte auszuwählen. Für einen besseren Vergleich hätte eventuell für dieses Fenster eine höhere Auflösung genutzt werden sollen. Des Weiteren können in Zukunft für den Vergleich von Eyetracking mit herkömmlichen Techniken mehrere Aspekte betrachtet werden. Zum einen ist nicht nur eine Betrachtung von Scatterplots mit verschiedenen Eigenschaften der Verteilung der Punktobjekte zu berücksichtigen, sondern auch „äußere“ Faktoren, wie zum Beispiel das Einbinden von verschiedenen Geschwindigkeitsstufen der Maus und die Berücksichtigung der Erfahrungen mit dem Eyetracker. Die Ergebnisse der Studie haben gezeigt, dass Probanden, die bereits Erfahrungen mit Eyetracking haben, für etwa 50 Prozent schnellere Antwortzeiten mit der Eyetracking-Technik geliefert haben. Abbildung 8.2 zeigt die Boxplots mit den Antwortzeiten für die drei zu vergleichenden Techniken. Wie zu sehen ist, liegt das Minimum und das untere Quartil des Boxplots für die Eyetracking-Technik wesentlich tiefer als die anderen Minima und unteren Quartile. Ausgehend vom Minimum bis zum Median (durchgehender Strich in der Box) sind 50 Prozent der Antwortzeiten enthalten. Der Median für den Eyetracker ist etwas tiefer als die Minima der anderen Techniken. Das bedeutet, dass 50 Prozent der Antwortzeiten von Probanden mit Eyetracking-Erfahrung schnellere Ergebnisse geliefert haben als die zu vergleichenden Techniken. Daraus kann geschlossen werden, dass eine Vertrautheit mit Eyetrackern Voraussetzung ist, um die Vorteile der Geschwindigkeit ausnutzen zu können und dass in Zukunft mehr Zeit in einer Studie für die Übung mit dem Eyetracker eingeplant werden muss.

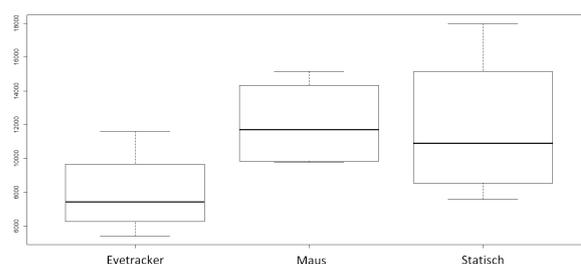


Abbildung 8.2: Boxplot für Probanden mit Eyetracking-Erfahrung

Das Einblenden der Labels außerhalb der Fokussierung (direkt außerhalb des Kreises, der die Fokussierung darstellt) hat die Performanz des Eyetrackers nicht beeinflusst. Auch das Betätigen der Leertaste, um Labels lesen zu können, hat das Ergebnis ebenfalls nicht

beeinflusst, da die Ergebnisse in zwei Fragetypen unterteilt wurden: Fragestellungen, die sich auf ein Label und Fragestellungen, die sich auf mehrere Labels konzentrieren. Wenn in der Fokussierung ein Punktobjekt liegt, dann erscheint das Label direkt oberhalb des Punktes und somit nicht außerhalb der Fokussierung. In diesem Fall ist das Label in der Nähe vom Punktobjekt und kann gelesen werden, ohne die Leertaste betätigen zu müssen. Da sich für beide Fragetypen keine Unterschiede hinsichtlich der Performanz ergeben haben, wurde gezeigt, dass das Einblenden der Labels direkt außerhalb des Kreises (der Fokussierung) und die Betätigung der Leertaste die Performanz nicht beeinflusst hat.

Um in Zukunft die Vorteile von Eyetracking bezüglich der Genauigkeit zu zeigen, ist eine Konzentration der Fragestellungen auf Cluster und weniger auf Ausreißer vorteilhafter. Die Fragestellungen sollten leicht verständlich sein und sich nicht zu stark auf die Scatterplot-Achsen richten.

Aus den Ergebnissen der Studie kann insgesamt geschlossen werden, dass der Einsatz von Eyetracking die Eingabezeit verringern und die Genauigkeit erhöhen kann, wenn sich die Technologie weiterentwickelt, sodass Probleme mit der Zeitverzögerung behoben werden und wenn der Umgang mit Eyetracking vertrauter wird.

9 Zusammenfassung und Ausblick

Im Bereich der Mensch-Computer-Interaktion gewinnt Eyetracking immer mehr an Bedeutung. Dabei lag der Fokus zu Beginn auf motorisch eingeschränkten Nutzern, um ihnen den Zugang zu Desktop-Anwendungen zu erleichtern. Mittlerweile richten sich Forschungsarbeiten auf blickbasierte Schnittstellen im Desktop-Bereich, die sich nicht ausschließlich auf körperlich beeinträchtigte Menschen orientieren. Dabei wird die Blickrichtung als Mausersatz zu Interaktion eingesetzt. Der Einsatz der Blickbedienung ist allerdings gewissen Einschränkungen ausgesetzt. Eine Bestimmung des Blickpunktes durch den Eyetracker erfolgt ungenau. Ein weiteres Problem ist der sogenannte Midas Touch Effekt, der besagt, dass die Blickrichtung als Mausersatz ungewollte Aktionen bei der Betrachtung von Objekten auslösen kann.

Diese Arbeit umgeht die genannten Probleme, indem die Blickrichtung nicht als Mausersatz eingesetzt wird, sondern indem der Eyetracker zur Selektion diskreter Punktobjekte in Scatterplots genutzt wird, um ausschließlich Labels von Punktobjekten einzublenden, die sich innerhalb der Fokussierung befinden. Dabei wird nicht der genaue Blickpunkt betrachtet, sondern nur der grobe Fokusbereich des Nutzers. Das Ziel dieser Arbeit ist, eine neue dynamische Eyetracking-Technik zu entwickeln und diese mit zwei bereits bestehenden Techniken zu vergleichen. Die Frage besteht darin, ob mit der neuen Technik gewisse Vorteile für das Lokalisierungsproblem in Scatterplots in Bezug zur Performanz und zur Genauigkeit erbracht werden.

Damit diese Frage untersucht werden kann, stellt diese Arbeit die Implementierung der neuen und der zwei bestehenden Techniken vor, um anschließend eine Vergleichsstudie zwischen diesen Techniken durchzuführen. Die neue Technik stellt eine dynamische Technik dar, die den Eyetracker für die Auslösung der Labels von Punktobjekten in Scatterplots einsetzt. Die zweite Technik löst die Labels dynamisch mit der Maus aus und die dritte Technik zeigt die Labels statisch an. Für die Implementierung der Techniken müssen geeignete Algorithmen eingesetzt werden, um die Labels automatisch zu platzieren. Diese Arbeit stellt verschiedene dynamische und statische Algorithmen vor, um eine geeignete Auswahl für die Umsetzung der Techniken zu treffen. Zudem werden die Ergebnisse für den Vergleich der Performanz zwischen Maus und Eyetracker aus anderen Forschungsarbeiten vorgestellt, die als Motivation für den Einsatz von Eyetracking in dieser Arbeit dienen.

Die Ergebnisse der Studie haben gezeigt, dass die neue Eyetracking-Technik keine signifikanten Unterschiede sowohl in Bezug zur Performanz als auch zur Genauigkeit gegenüber den zu vergleichenden Techniken aufweist. Allerdings wurde festgestellt, dass 50 Prozent der Antwortzeiten trotz Zeitverzögerungen geringer sind als die zu vergleichenden Techniken, sobald Probanden Erfahrungen mit Eyetracking haben.

Ausblick

Wenn die Technologie des Eyetrackers sich weiterentwickelt, sodass Probleme mit der Zeitverzögerung und Einschränkungen der Kopfbewegung behoben werden, kann in Zukunft die Eingabezeit durchaus verringert werden. Die Genauigkeit der neuen Eyetracking-Technik kann bei der Suche von Labels erhöht werden, wenn sich die Suche auf Cluster in Scatterplots konzentriert.

Zu dieser Arbeit lassen sich Anknüpfungspunkte zum Thema Labeling in Kombination mit Eyetracking identifizieren. Das dynamische Labeling mit Eyetracking lässt sich für verschiedene Bildanalysen einsetzen. Im Bereich der Geo-Informationssysteme können beispielsweise verschiedene Daten, wie die Beschriftung einer Länderkarte, dynamisch mit einem Eyetracker visualisiert werden. Aber auch biomedizinische Abbildungen, wie Strukturen mit Beschriftungen des Kopfes oder Gehirns, könnten interaktiv durch Blicke gesteuert werden. Für die Umsetzung solcher Labeling-Verfahren müssen genaue Abgrenzungen der zu beschrifteten Flächen erkannt und die Blickpunkte durch den Eyetracker genauer aufgenommen werden. Solche Bildanalysen können eventuell auch mit 3D-Visualisierungen kombiniert werden.

Literaturverzeichnis

- [Bar10] E. Bartmann. *Processing*. O'Reilly Verlag GmbH & Co. KG, 2010.
- [BBS⁺10] R. Biedert, G. Buscher, S. Schwarz, J. Hees, A. Dengel. Text 2.0. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, S. 4003–4008. 2010.
- [BEPW00] Backhaus, Erichson, Plinke, Weiber. *Multivariate Analyseverfahren: Eine anwendungsorientierte Einführung*. Springer Berlin, 9 Auflage, 2000.
- [BHH⁺02] K. Bach, K. Hanig, T. Hoffmann, W. Kresse, J. Löcherbach, P. Rosenthal, S. Rudnick, P. Schreiber, M. Thon, A. Wolff. Beschriftungsalgorithmen in Theorie & Praxis. Technical report 13/2002, Institut für Mathematik und Informatik, Universität Greifswald, 2002.
- [CIP96] M. Chalmers, R. Ingram, C. Pfranger. Adding imageability features to information displays. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, S. 33–39. ACM, 1996.
- [CKB09] A. Cockburn, A. Karlson, B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, 2009.
- [CMS95] J. Christensen, J. Marks, S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graph.*, 14:203–232, 1995.
- [DC01] R. Dodge, T. S. Cline. The angle velocity of eye movements. In *Psychological Review*, Band 8(2), S. 145–157. 1901.
- [DK91] J. Denes, A. Keedwell. *Latin Squares: New Developments in the Theory and Applications*. Elsevier Science, 1991.
- [Duc07] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2. Auflage, 2007.
- [ED06] G. Ellis, A. Dix. The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '06, S. 266–269. 2006.
- [EKW04] D. Ebner, G. W. Klau, R. Weiskircher. Label number maximization in the slider model. In *Proceedings of the 12th international conference on Graph Drawing*, S. 144–154. 2004.
- [FJL50] P. M. Fitts, R. Jones, J. L. Milton. Eye movements of aircraft pilots during instrument-landing approaches. In *Aeronautical Engineering Review*, Band 9(2), S. 24–29. 1950.

- [FKCP99] J.-D. Fekete, R. A. Kastler, L. Chantrerie, C. Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In *CHI'99*, S. 512–519. Pittsburgh, PA, USA, 1999.
- [FM02] M. M. Farid, F. Murtagh. Eye-movements and Voice as Interface Modalities to Computer Systems. In *SPIE - Opto*. 2002.
- [FS04] G. Fuchs, H. Schumann. Visualizing Abstract Data on Maps. In *Proceedings of the Information Visualisation, Eighth International Conference, IV '04*, S. 139–144. 2004.
- [FTSS09] G. Fuchs, C. Thiede, M. Sips, H. Schumann. Device-based Adaptation of Visualizations in Smart Environments. In *Workshop Collaborative Visualization on Interactive Surfaces (CoVIS), IEEE VisWeek 2009*. 2009.
- [FV05] D. Fono, R. Vertegaal. EyeWindows: Evaluation of Eye-Controlled Zooming Windows for Focus Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '05*, S. 151–160. ACM, 2005.
- [GE06] E. D. Guestrin, M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE transactions on bio-medical engineering*, 53(6):1124–1133, 2006.
- [Gus96] R. Guski. *Wahrnehmen - ein Lehrbuch*. Kohlhammer, 1996.
- [Hay73] W. L. Hays. *Statistics for the social sciences*. Teil 5. Holt, Rinehart and Winston, 2. auflage Auflage, 1973.
- [HC05] A. Hornof, A. Cavender. EyeDraw: Enabling Children with Severe Motor Impairments to Draw with Their Eyes. In *CHI 2005*, S. 161–170. ACM Press, 2005.
- [Hir82] S. A. Hirsch. An Algorithm for Automatic Name Placement Around Point Data. *The American Cartographer*, 9(1):5–17, 1982.
- [HKMS11] H. Hagendorf, J. Krummenacher, H.-J. Müller, T. Schubert. *Allgemeine Psychologie für Bachelor: Wahrnehmung und Aufmerksamkeit*. Springer Berlin Heidelberg, 2011.
- [HRD03] J. Hyona, R. Radach, H. Deubel. *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, Kapitel 4, S. 573–605. Elsevier Science BV, Oxford, 2003.
- [HV03] S. Heinsen, P. Vogt. *Usability praktisch einsetzen: Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte*, Kapitel 11, S. 154–169. Carl Hanser Verlag, München, 2003.
- [IL99] C. Iturriaga, A. Lubiwi. Elastic Labels Around the Perimeter of a Map. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures*, S. 306–317. Springer-Verlag, 1999.
- [Imh62] E. Imhof. Positioning Names on Maps. *International Yearbook of Cartography*, 2:93–129, 1962.
- [Jac90] R. J. K. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *CHI'90*, S. 11–18. 1990.

- [Jac95] R. J. K. Jacob. *Virtual environments and advanced interface design*, Kapitel Eye tracking in advanced interface design, S. 258–288. Oxford University Press, Inc., New York, NY, USA, 1995.
- [Jav78] E. Javal. Essai sur la physiologie de la lecture. In *Annales d’Oculistique*. 1878.
- [JLo9] J. Janssen, W. Laatz. *Statistische Datenanalyse mit SPSS für Windows: Eine anwendungsorientierte Einführung in das Basissystem und das Modul Exakte Tests*. Springer, 6 Auflage, 2009.
- [JMS05] C. H. Judd, C. N. McAllister, W. M. Steel. General introduction to a series of studies of eye movements by means of kinoscopic photographs. In *Psychological Review, Monograph Supplements*. 1905.
- [Kab12] R. I. Kabacoff. Quick-R: accessing the power of R, 2012. URL <http://www.statmethods.net/graphs/scatterplot.html>.
- [Kau07] T. Kauffman. *Geriatric Rehabilitation Manual*. Elsevier Science Health Science, 2007.
- [KM03] G. W. Klau, P. Mutzel. Optimal Labelling of Point Features in Rectangular Labelling Models. *Mathematical Programming*, 94:435–458, 2003.
- [KSW99] M. V. Kreveld, T. Strijk, A. Wolff. Point Labeling with Sliding Labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.
- [Kü04] D. Kühlke. *Optik*. Harri Deutsch Verlag, 2. Auflage, 2004.
- [LAE⁺10] D. J. Lehmann, G. Albuquerque, M. Eisemann, A. Tatu, D. A. Keim, H. Schumann, M. Magnor, H. Theisel. Visualisierung und Analyse multidimensionaler Datensätze. *Informatik-Spektrum*, 33:589–600, 2010.
- [Mar03] J. Martens. *Statistische Datenanalyse mit SPSS für Windows*. Oldenbourg Wissenschaftsverlag, 2. Auflage, 2003.
- [MBGo8] A. McNamara, R. Bailey, C. Grimm. Improving Search Task Performance Using Subtle Gaze Direction. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, S. 51–56. ACM, 2008.
- [MS93] J. Marks, S. Shieber. The Computational Complexity of Cartographic Label Placement. Technischer Bericht, Center for Research in Computing Technology, Harvard University, 1993.
- [Nau12] D. Nauth. *Durch die Augen meines Kunden: Praxishandbuch für Usability Tests mit einem Eyetracking System*. Diplomica Verlag GmbH, Hamburg, 2012.
- [NBOH96] M. Nagel, A. Brenner, R. Ostermann, K. Heschke. *Grafische Datenanalyse*. Gustav Fischer Verlag Stuttgart Jena New York, 1996.
- [PB05] A. Poole, L. J. Ball. Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future. In *Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction*. Pennsylvania: Idea Group, Inc. 2005.

- [PBM09] M. Pritzel, M. Brand, H. J. Markowitch. *Gehirn und Verhalten: Ein Grundkurs der physiologischen Psychologie*. Spektrum Akademischer Verlag, 2009.
- [Poi11] T. M. Poitschke. *Blickbasierte Mensch-Maschine Interaktion im Automobil*. Dissertation, Technische Universität München, 2011.
- [Rey09] G. D. Rey. *E-Learning: Theorien, Gestaltungsempfehlungen und Forschung*. Psychologie Lehrbuch. Huber Verlag, 2009.
- [SGSo8] L. Schmidt, J. Grosche, C. M. Schlick. *Ergonomie und Mensch-Maschine-Systeme*, Kapitel 21, S. 371–391. Springer Verlag, Berlin, Heidelberg, 2008.
- [Shio8] D. Shiffman. *Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction*. Morgan Kaufmann Publishers Inc., 2008.
- [Shn96] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, S. 336–343. IEEE Computer Society, Washington, DC, USA, 1996.
- [SJT00] L. E. Sibert, R. J. K. Jacob, J. N. Templeman. *Evaluation And Analysis Of Eye Gaze Interaction*, 2000.
- [SM00] H. Schumann, W. Müller. *Visualisierung: Grundlagen und allgemeine Methoden*. Springer Verlag Berlin Heidelberg New York, 2000.
- [SMo4] H. Schumann, W. Müller. Informationsvisualisierung: Methoden und Perspektiven. *it - Information Technology*, 46:135–141, 2004.
- [Teao8] R. D. C. Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.
- [Vero8] R. Vertegaal. A Fitts Law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces*. 2008.
- [WM87] C. Ware, H. H. Mikaelian. An Evaluation of an Eye Tracker as a Device for Computer Input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, S. 183–188. ACM, New York, NY, USA, 1987.
- [Yar67] A. L. Yarbus. *Eye movements and vision*. Plenum Press, New York, 1967.
- [YCL05] M. Yamamoto, G. Camara, L. A. N. Lorena. Fast Point-Feature Label Placement Algorithm for Real Time Screen Maps. In *GeoInfo*, S. 122–138. INPE, 2005.
- [Yoe72] P. Yoeli. The Logic of Automated Map Lettering. *Cartographic Journal*, 9:99–108, 1972.
- [Zor86] S. Zoraster. Integer Programming applied to the Map Label Placement Problem. *Cartographica*, 23(3):16–27, 1986.

Alle URLs wurden zuletzt am 27. 11. 2012 geprüft.

Erklärung

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

(Giuseppina Maringolo)