

Projektbericht

Consultancy Project 2

«CLUSTERBASIERTER VERGLEICH VON EYE-TRACKING-DATEN IN RÄUMLICHER UND ZEITLICHER DIMENSION»

Name / Vorname: Reiser Sharon

Matrikelnr.: 17-655-366

Adresse: Alte Römerstrasse 28, 8404 Winterthur

E-Mail: sharon.reiser@stud.fhgr.ch.

Name / Vorname: Pellegatta Serge

Matrikelnr.: 18-165-068

Adresse: Usterstrasse 19, 8620 Wetzikon ZHE

Mail: serge.pellegatta@stud.fhgr.ch

Fachhochschule Graubünden

Studiengang: MSc Data Visualization

Modul: Consultancy Project 2

Betreuer: Dr. rer. nat. Michael Burch

Abgabedatum: 21.01.2025

Abstract

Aufbauend auf den Ergebnissen des Consultancy Project 1, das grundlegende Visualisierungen und erste Analysen zur Taskduration farbiger und graustufenbasierter Karten bereitstellte, wurde ein webbasiertes Dashboard weiterentwickelt. Dieses ermöglicht detaillierte Visualisierungen, statistische Analysen und eine differenzierte Untersuchung von Sakkaden- und Fixationsmustern. Ziel war es, die Analysefähigkeit durch die Implementierung neuer Werkzeuge wie Cluster-Analysen, Scarfplots und Opacity Maps zu verbessern und die Identifikation von Areas of Interest (AOIs) mithilfe von k-means Clustering zu unterstützen. Diese Erweiterungen erlauben es, Blickbewegungen in räumlicher und zeitlicher Dimension präzise zu analysieren und Unterschiede zwischen Nutzergruppen sowie deren Denkmustern sichtbar zu machen.

Die Forschungsfragen des Projekts zielen darauf ab, zu untersuchen, wie Farbcodierung und Designaspekte die Nutzerführung beeinflussen, welche Blickstrategien Nutzende verfolgen und wie diese Erkenntnisse zur Verbesserung von Kartendesigns genutzt werden können. Die Ergebnisse zeigen, dass farbige Karten die Nutzerführung erheblich verbessern, indem sie die Taskduration im Vergleich zu graustufenbasierten Karten deutlich reduzieren. Nutzende konzentrieren ihre Aufmerksamkeit auf kritische AOIs, wie Stationsübergänge, während weniger relevante Bereiche kaum beachtet werden. Die Visualisierungen machen zudem individuelle Verhaltensmuster sichtbar, beispielsweise das wiederholte Durchlaufen von Strecken oder das schnelle Springen zwischen Start und Ziel, was gezielte Ansätze zur Optimierung von Kartenlayouts ermöglicht.

Trotz der Fortschritte weist das Dashboard einige Limitierungen auf. Die Vielzahl an Parametern kann unerfahrene Nutzende überfordern, und die fehlende Möglichkeit, eigene Datensätze hochzuladen, schränkt die Flexibilität ein. Zudem liegt der Fokus auf Metrokarten, was die allgemeine Anwendbarkeit begrenzt. Zukünftige Erweiterungen könnten die Integration statistischer Tests, erweiterte Filtermöglichkeiten und interaktive Funktionen wie die Erstellung eigener AOIs umfassen.

Das Projekt liefert wertvolle Beiträge zur Erforschung der visuellen Wahrnehmung und bietet ein leistungsstarkes Werkzeug zur Analyse von Eye-Tracking-Daten. Es unterstützt Forschende dabei, Informationsvisualisierung zu verbessern und effektive Designlösungen abzuleiten.

Schlüsselwörter:

Eye Tracking, Datenvisualisierung, interaktives Dashboard, Metrokarten, Python, Plotly

Inhaltsverzeichnis

Abstract	II
Inhaltsverzeichnis	III
Formelverzeichnis.....	V
1. Einleitung	1
1.1. Projektziel.....	1
1.2. Abgrenzung	2
1.3. Zielgruppe	2
2. Theoretischer Ansatz	3
2.1. Untersuchung der visuellen Aufmerksamkeit	3
2.1.1. Definition visual Tasks und Areas of Interest	3
2.1.2. Definition Scan Pfad, Fixation und Sakkade.....	4
2.1.3. Motivation für die Erfassung von Eye Tracking Daten	5
2.2. Visualisierungstechniken	5
2.2.1. Point-based vs. AOI-based Methods	5
2.2.2. Point-based Methods	6
2.2.3. AOI-based Methods	7
2.3. Dashboard Design	11
2.3.1. Definition Dashboard.....	11
2.3.2. Nutzen von Dashboards.....	11
2.3.3. Designregeln für Datenvisualisierungen	12
3. Methodik	13
3.1. Analyse der Rohdaten und Preprocessing.....	13
3.1.1. Exploration der Daten	15
3.2. Analyse der Anspruchsgruppen	16
3.2.1. Identifikation der Anspruchsgruppen	16
3.2.2. Bedürfnisse und Anforderungen der Anspruchsgruppen	16
3.2.3. Personalisierung und Anpassung.....	16
3.3. Vom Wireframe zum Mockup.....	16
3.3.1. Wireframe-Erstellung	17

3.3.2.	Mockup-Erstellung	17
3.3.3.	Bedeutung des Übergangs vom Wireframe zum Mockup.....	17
3.4.	Dashboard Umsetzung mit Python Plotly.....	17
3.4.1.	Datensatz einlesen und transformieren	18
3.4.2.	Dash	18
3.4.3.	Dash Core Components.....	18
3.4.4.	Dash HTML Components.....	18
3.4.5.	Cascading Style Sheets (CSS)	18
3.4.6.	Plotly Express	18
3.4.7.	Callbacks	18
4.	Ergebnisse.....	19
4.1.	Use Case.....	19
4.2.	Entwicklung mit Python.....	20
4.2.1.	Datenimport	20
4.2.2.	Layout.....	20
4.2.3.	Interaktionen	22
4.2.4.	Visualisierungen.....	22
4.2.5.	Ausführung der Applikation 'app.py'	25
4.3.	Erkenntnisse aus dem Dashboard	34
5.	Implikationen für die Praxis	38
5.1.	Praxisbezug.....	38
5.2.	.Limitation	38
5.3.	Potenzial zur Weiterentwicklung	39
6.	Literaturverzeichnis.....	41
7.	Hilfsmittelverzeichnis	44
8.	Anhang	45
	Selbstständigkeitserklärung	61

*Bildquelle Titelblatt: <https://www.weareconflux.com/en/blog/eye-tracking-in-user-research/>,
aufgerufen am 01.04.2024*

Abbildungsverzeichnis

Abbildung 1: Beispiel von vier verschiedenen gruppierten Interessensbereichen (AOIs)	4
Abbildung 2: Schematische Darstellung eines Scan Pfades	5
Abbildung 3: Visualisierung einer Heat Map (links) und eines Gaze Plots (rechts).....	6
Abbildung 4: Opacity Plot im Vergleich mit Heat Map.....	7
Abbildung 5: Beispiel eines Scarf-Plots.....	8
Abbildung 6: Clustering mit K-Means.....	10
Abbildung 7 Anordnung der sieben Layout-Container	21
Abbildung 8: Finales Dashboard	33
Abbildung 9 Barcelona S1 Grey Map mit Cluster	34
Abbildung 10 User P15 Scarf Plot in Grey Map.....	35
Abbildung 11 User P11 Scarf Plot in Grey Map	36
Abbildung 12: Visualisierungen der globalen Analyse	58
Abbildung 13: Visualisierungen der Detail-Analyse (Beispiel Tokyo)	59
Abbildung 14 Neue Visualisierungen der Detail-Analyse (Beispiel Tokio)	60

Formelverzeichnis

Formel 1: Formen Intra Cluster Distanz	9
Formel 2: Formen Inter Cluster Distanz	9
Formel 3: Berechnung der Task Duration.....	14
Formel 4: Berechnung der Task Duration Kategorie.....	14
Formel 5: Berechnung der Sakkaden Länge.....	15
Formel 4: Berechnung der Average Fixation Duration.....	15

1. Einleitung

Der Augenkontakt spielt eine zentrale Rolle in der nonverbalen Kommunikation, indem er Emotionen, Interesse oder Desinteresse signalisiert und eine tiefere soziale Interaktion ermöglicht (Burch, 2021). Unsere Augen richten sich auf das, was unsere Aufmerksamkeit fesselt. Da Menschen nur begrenzt in der Lage sind, mehrere Sinneseindrücke gleichzeitig zu verarbeiten, konzentriert sich die visuelle Aufmerksamkeit stets auf ein spezifisches sichtbares Objekt (Duchowski, 2007). Dieses Konzept bildet die Grundlage für die Messung von Eye Tracking-Daten, die uns zeigen, wie und warum der Blick auf bestimmte Dinge gelenkt wird und was dies über Wahrnehmung und Interessen aussagt (Burch, 2021).

Das vorliegende Projekt ist eine Fortführung des Dashboards, das im ersten Teil des Consultancy-Projekts entwickelt wurde. Während im ersten Projekt ein webbasiertes Dashboard erstellt wurde, das grundlegende Visualisierungen zur Analyse von Eye Tracking-Daten beinhaltet, liegt der Fokus des aktuellen Projekts auf der Erweiterung und Optimierung dieses Dashboards. Ziel ist es, neue statistische Metriken und Visualisierungen zu implementieren, um die Analyse von Sakkaden- und Fixationsdaten zu verbessern. Zusätzlich wird ein Algorithmus entwickelt, der die Sortierung oder Kategorisierung der Eye-Tracking-Daten ermöglicht. Zur Erweiterung der Analyse werden clusterbasierte Ansätze integriert, die es erlauben, Muster in den Eye-Tracking-Daten zu identifizieren und verschiedene Nutzergruppen oder Aufgabenbereiche zu vergleichen. Mithilfe dieser Methodik können zum Beispiel Ähnlichkeiten und Unterschiede im Blickverhalten zwischen Gruppen mit unterschiedlichen kognitiven Strategien oder Präferenzen sichtbar gemacht werden. Diese erweiterte Analyse bietet nicht nur tiefere Einblicke in die Daten, sondern unterstützt auch die Entwicklung spezifischer Handlungsempfehlungen basierend auf den identifizierten Clustern.

Die theoretischen Grundlagen zur Messung von visueller Aufmerksamkeit, der verwendeten Visualisierungstechniken und der Integration von Dashboards werden in Kapitel 2 beschrieben. Kapitel 3 erläutert die Methodik und den zugrundeliegenden Datensatz. Die Ergebnisse der Implementierungen und Erweiterungen werden in Kapitel 4 dargestellt, während Kapitel 5 die Implikationen für die Praxis und potenzielle Anwendungsmöglichkeiten diskutiert.

1.1. Projektziel

Das Ziel des Consultancy Project 2 besteht in der umfassenden Weiterentwicklung eines webbasierten Dashboards zur Analyse und Visualisierung von Eye-Tracking-Daten. Dabei sollen nicht nur bestehende Visualisierungen optimiert, sondern auch neue analytische

Funktionen integriert werden, um eine tiefere Einsicht in Sakkaden- und Fixationsmuster zu ermöglichen.

Ein zentrales Element des Projekts ist die Implementierung eines Algorithmus, der die strukturierte Sortierung und Kategorisierung der Eye-Tracking-Daten nach relevanten Kriterien erlaubt. Dies schafft die Grundlage für eine effizientere Datenanalyse und ermöglicht es den Nutzenden, spezifische Muster und Zusammenhänge einfacher zu identifizieren.

Die Datenbasis besteht aus 24 Metro-Karten unterschiedlicher Städte, die im Rahmen einer Studie erhoben wurden. Diese Studie untersucht, wie farbige und graustufenbasierte Karten wahrgenommen werden und welche Unterschiede in der Nutzerführung erkennbar sind. Das Zentrum für Data Analytics, Visualization and Simulation (DAViS) der Fachhochschule Graubünden unterstützt die iterative Weiterentwicklung des Projekts bis Januar 2025.

1.2. Abgrenzung

Die Arbeit fokussiert sich ausschliesslich auf die technische Entwicklung und Optimierung des Dashboards. Die Interpretation oder Bewertung der Eye-Tracking-Daten liegt ausserhalb des Projektumfangs und wird den Nutzenden des Dashboards überlassen. Das Dashboard dient als unterstützendes Werkzeug für Forschende, um eigenständig Analysen effizient durchführen zu können.

1.3. Zielgruppe

Das Dashboard richtet sich primär an Forschende und Analyst:innen im Bereich Eye-Tracking. Durch die erweiterten Metriken, neuen Visualisierungen und den Algorithmus zur Datenkategorisierung bietet es eine verbesserte Plattform zur Evaluation und zum Vergleich der Effektivität farbiger und graustufenbasierter Kartendarstellungen. Ziel ist es, empirische Untersuchungen zu erleichtern und eine fundierte Entscheidungsgrundlage für weitere Studien und praxisorientierte Anwendungen zu schaffen.

2. Theoretischer Ansatz

In diesem Kapitel wird die relevante Literatur aufgearbeitet, um den theoretischen Rahmen der Studie zu untermauern. Ziel ist es, bestehende Forschungsergebnisse sowie deren Bezug zur vorliegenden Untersuchung darzustellen.

Ein Grossteil der theoretischen Grundlagen wurde bereits im Rahmen des ersten Consultancy Projects behandelt und bildet die Basis für die aktuelle Arbeit. Daher wird an dieser Stelle lediglich das grundlegende Konzept zusammengefasst, mit Verweis auf die im Consultancy Project (Nyffeler , Reiser , & Pellegatta, 2024) ausführlich dokumentierten Ansätze.

Neue theoretische Erkenntnisse, die für die Erweiterung des Dashboards und die Implementierung zusätzlicher Analysemethoden relevant sind, werden hingegen ausführlich erläutert. Dies umfasst insbesondere aktuelle Entwicklungen im Bereich der Eye-Tracking-Analyse, clusterbasierter Methoden zur Datenkategorisierung und innovativer Visualisierungsansätze. Der Fokus liegt dabei auf der Integration dieser Theorien in den praktischen Kontext des Projekts, um die Weiterentwicklung des Dashboards fundiert zu untermauern.

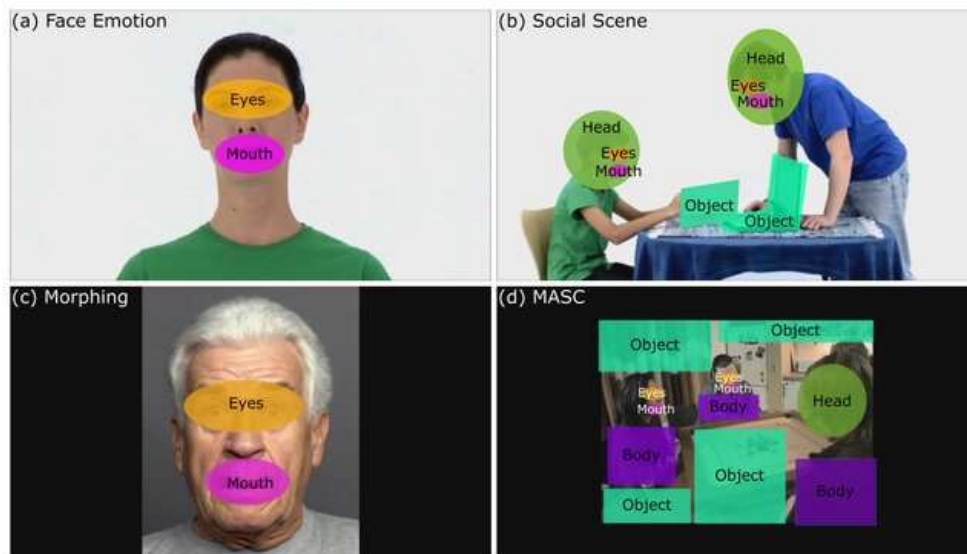
2.1. Untersuchung der visuellen Aufmerksamkeit

Die Eye-Tracking-Technologie ist eine zentrale Methode zur Untersuchung visueller Aufmerksamkeit und wird in verschiedenen Forschungsfeldern eingesetzt. Sie ermöglicht tiefere Einblicke in die kognitiven Prozesse, die das menschliche Sehverhalten steuern. Die Messung visueller Aufmerksamkeit bei Menschen reicht bereits seit über einem Jahrhundert, wobei sowohl qualitative als auch quantitative Ansätze entwickelt wurden (Duchowski, 2007). Nähergehende Informationen zu den theoretischen Grundlagen und bisherigen Forschungsergebnissen zur visuellen Aufmerksamkeit können im vorhergehenden Projekt (Nyffeler , Reiser , & Pellegatta, 2024) eingesehen werden.

2.1.1. Definition visual Tasks und Areas of Interest

Die Eye-Tracking-Technologie konzentriert sich auf die Messung von Bearbeitungszeiten und Genauigkeitsraten bei der Ausführung definierter visueller Aufgaben, sogenannter visual Tasks. Visual Tasks sind spezifische Anforderungen oder Stimuli, die den Teilnehmern in experimentellen Studien präsentiert werden (Burch, 2021). Zur detaillierteren Analyse werden Areas of Interest (AOIs) festgelegt – dies sind spezifische Regionen, die von besonderem Interesse für die Quantifizierung der visuellen Aufgabe sind (Blascheck, et al., 2017). Abbildung 1 zeigt anhand vier Beispielen, wie AOI's für verschiedene Aufgaben dynamisch oder manuell gruppiert werden können (Prillinger et al., 2023).

Abbildung 1: Beispiel von vier verschiedenen gruppierten Interessensbereichen (AOIs)



Anmerkung: (Prillinger et al., 2023)

2.1.2. Definition Scan Pfad, Fixation und Sakkade

Der Eye-Tracking-Datensatz enthält einen Scanpfad, der in Fixationen und Sakkaden unterteilt ist. Fixationen sind Zeitintervalle, in denen die Augen stabil auf einem Blickpunkt verweilen, wobei Informationen aus diesem Bereich detailliert aufgenommen werden. Sakkaden beschreiben das Springen von einem Fixationspunkt zum nächsten, während die visuelle Wahrnehmung in dieser Phase vorübergehend unterdrückt wird (Blascheck, et al., 2017).

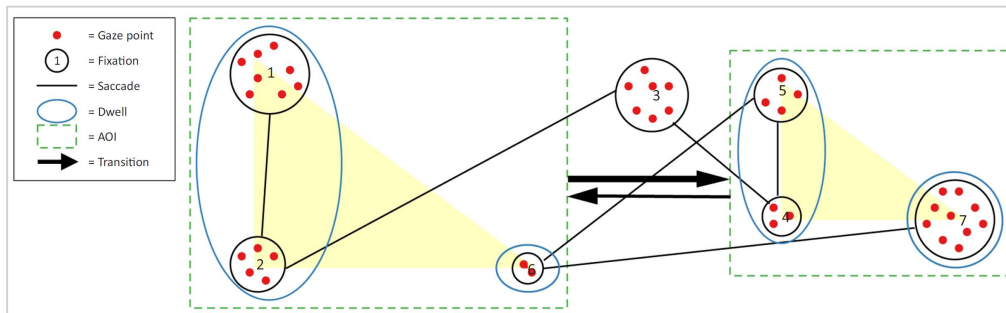
Fixationen: Zeitintervalle, während derer die Augen stabil auf einem bestimmten Blickpunkt verweilen, werden als Fixationen bezeichnet. Während dieser Zeit nimmt das Auge detaillierte Informationen aus dem fixierten Bereich auf. Fixationen sind wichtige Indikatoren dafür, welche Teile eines visuellen Stimulus die Aufmerksamkeit einer Person anziehen (Blascheck et al., 2017).

Sakkaden: Das Springen von einem Fixationspunkt zum nächsten wird als Sakkade bezeichnet. Während dieser Zeit wird die visuelle Wahrnehmung vorübergehend unterdrückt. Das Gehirn verarbeitet während einer Sakkade kaum visuellen Informationen (Blascheck et al., 2017).

Scan Pfad: Die gesamte Abfolge von Fixationen und Sakkaden bilden einen Scan Pfad (Blascheck et al., 2017). Abbildung 2 zeigt eine schematische Darstellung eines Scanpfades, unterteilt in Fixationen und Sakkaden, sowie die Eingrenzung von Interessensbereichen. Die roten Blickpunkte werden zu einer Fixation zusammengefasst, da sie relativ konstant sind. Der Wechsel zum nächsten Fixationspunkt wird als Sakkade bezeichnet und ist als schwarze Linie eingezeichnet. Die Reihenfolge von einer Fixation zur nächsten definiert

den Ablauf des Scan Pfades. Der definierte Interessensbereich innerhalb eines Stimulus ist in grün eingezeichnet.

Abbildung 2: Schematische Darstellung eines Scan Pfades



Anmerkung: Blascheck, et al., 2017).

2.1.3. Motivation für die Erfassung von Eye Tracking Daten

Das Ziel der Eye-Tracking-Daten liegt darin, Einblicke in die kognitiven Prozesse der visuellen Aufmerksamkeit zu gewinnen. Dies geschieht durch die Analyse des Blickverlaufs von Probanden während experimenteller Aufgaben. Die gewonnenen Erkenntnisse werden für verschiedene Anwendungsbereiche genutzt, darunter Usability- und UX-Forschung, Werbung und Marketing, sowie die Visualisierungswissenschaften (Duchowski, 2007; Wedel & Pieters, 2008; Burch & Schmid, 2023). Weiterführende Informationen zu den Anwendungsfeldern finden sich im vorhergehenden Projekt (Nyffeler, Reiser, & Pellegatta, 2024).

2.2. Visualisierungstechniken

Der Typ des Stimulus beeinflusst die Auswahl und Entwicklung von Eye Tracking Visualisierungstechniken massgeblich (Burch, 2022). Stimuli können dabei sowohl statische als auch dynamische Eigenschaften aufweisen. Als statische Stimuli werden beispielsweise Bilder oder Textobjekte bezeichnet, deren Inhalte sich nicht verändern. Dynamische Stimuli hingegen zeichnen sich durch Interaktivität aus, wie beispielsweise Videos oder Szenen aus der realen Welt (Blascheck et al., 2017). Da diese Projektarbeit den Fokus ausschliesslich auf statische Bilder legt, werden im Folgenden Methoden zur Visualisierung vorgestellt, die speziell für nichtbewegte Stimuli geeignet sind.

2.2.1. Point-based vs. AOI-based Methods

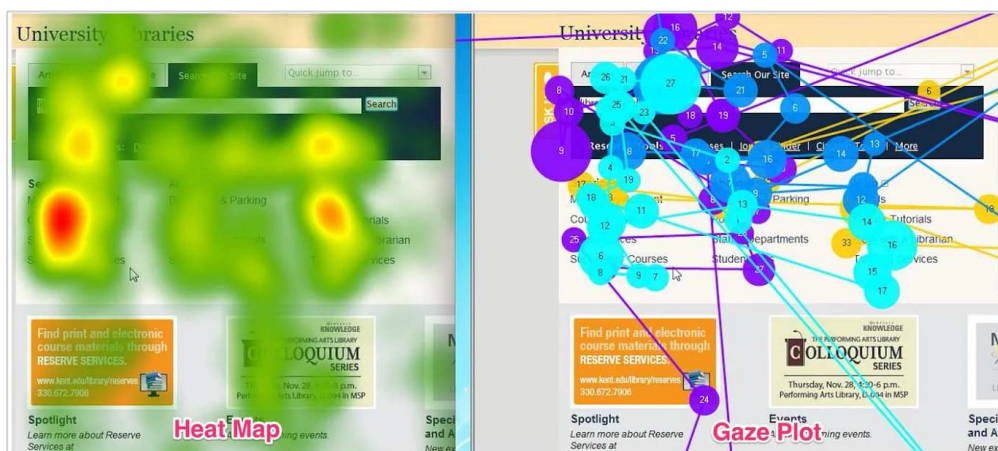
Die Analyse der in Eye-Tracking-Experimenten erhobenen Blickdaten erlaubt gemäss Burch (2017) die Unterscheidung zwischen punktbasierten (point-based) und Interessensbereich-basierten (AOI-based) Visualisierungstechniken. Die punktbasierte Auswertung befasst sich mit der Gesamtheit der Augenbewegungen sowie deren räumlicher und/oder zeitlicher Verteilung. Das heisst, die X- und Y-Koordinaten von Fixierungspunkten stehen im Fokus. Die zeitliche Abfolge der Sakkaden kann ebenfalls in die Visualisierung inkludiert

werden. Die geläufigsten punktbasierten Visualisierungstypen sind Heat Maps und Scatter-plots, die die Verteilung und Dichte der Fixationspunkte aufzeigen und zusätzlich auch Scanpfade visualisieren können (Blascheck et al., 2017).

2.2.2. Point-based Methods

Der aktuelle Forschungsstand legt nahe, dass Heat Maps und Gaze Plots zu den am häufigsten verwendeten Visualisierungstechniken für Eye Tracking-Daten zählen (Blascheck, et al., 2017). Beide Techniken veranschaulichen, welche Bereiche eines Stimulus von den Probanden am intensivsten betrachtet werden. Heat Maps nutzen eine farbliche Kodierung, um die Verteilung der visuellen Aufmerksamkeit darzustellen. Dabei werden intensiv betrachtete Bereiche in Rot gekennzeichnet, während weniger beachtete Regionen in Blau oder Grün erscheinen. Gaze Plots hingegen repräsentieren die Fixationen durch Kreise, deren Grösse proportional zur Fixationsdauer ist. In diesem Sinne deuten grössere Kreise auf eine längere Verweildauer hin. Darüber hinaus visualisieren Gaze Plots die Reihenfolge der Blicksequenzen durch nummerierte Fixationspunkte und die Augenbewegungen (Sakkaden) durch verbundene Linien (Tan, 2016). Abbildung 4 zeigt ein Beispiel einer Heat Map und eines Gaze Plots.

Abbildung 3: Visualisierung einer Heat Map (links) und eines Gaze Plots (rechts)



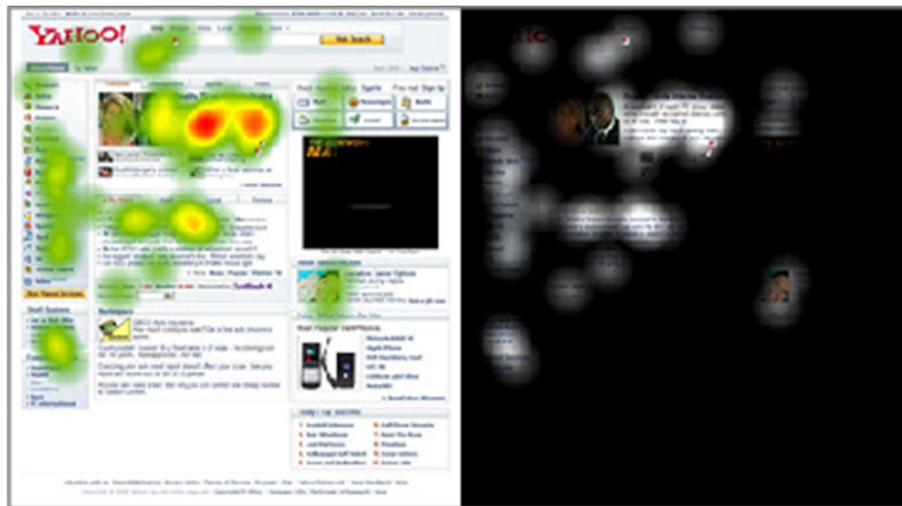
Anmerkung: (Tang, 2016)

Neben Heat Maps und Gaze Plots hat sich der Opacity Plot als ergänzende Visualisierungstechnik etabliert, um einige der Limitierungen der bestehenden Methoden zu adressieren. Der Opacity Plot nutzt eine transparente Überlagerung von Fixationspunkten, um sowohl die Verteilung der visuellen Aufmerksamkeit als auch individuelle Blickmuster darzustellen. Je höher die Fixationsdichte an einem Punkt, desto weniger transparent (also opaker) wird die Darstellung. Dies ermöglicht eine differenzierte Visualisierung, bei der sowohl die Intensität der Aufmerksamkeit als auch deren räumliche Verteilung sichtbar wird, ohne dabei die Ergebnisse einzelner Probanden zu überdecken.

Ein wesentlicher Vorteil des Opacity Plots besteht darin, dass er die visuelle Überlastung reduziert, die bei Heat Maps und Gaze Plots häufig auftritt, wenn Daten von mehreren Probanden überlagert werden. Statt die Fixationsdaten durch Farben oder Kreise darzustellen, nutzt der Opacity Plot eine Graustufen-Kodierung, bei der dunklere Bereiche auf eine höhere Fixationsdichte hinweisen. Dadurch bleibt die Klarheit der Darstellung auch bei grösseren Datenmengen erhalten.

Laut Smith et al. (2023) bietet der Opacity Plot zusätzlich eine bessere Integration in multivariate Analysen, da die Transparenzwerte mit anderen Variablen, wie Fixationsdauer oder Sakkadenlänge, kombiniert werden können. In Abbildung 5 wird ein Beispiel eines Opacity Plots gezeigt, der die Fixationsdichte von 20 Probanden auf einer graustufenbasierten Metroroute darstellt.

Abbildung 4: Opacity Plot im Vergleich mit Heat Map



Anmerkung: (Veloso & Almeida, 2011)

Obwohl der Opacity Plot viele Vorteile bietet, ist er nicht in allen Anwendungskontexten optimal. Eine potenzielle Limitation liegt in der Schwierigkeit, komplexe Blicksequenzen darzustellen, da die Reihenfolge der Fixationen in der Darstellung verloren gehen kann. Aus diesem Grund eignet sich der Opacity Plot besonders gut für die Analyse von Fixationsdichten und weniger für die Visualisierung von Blickverläufen.

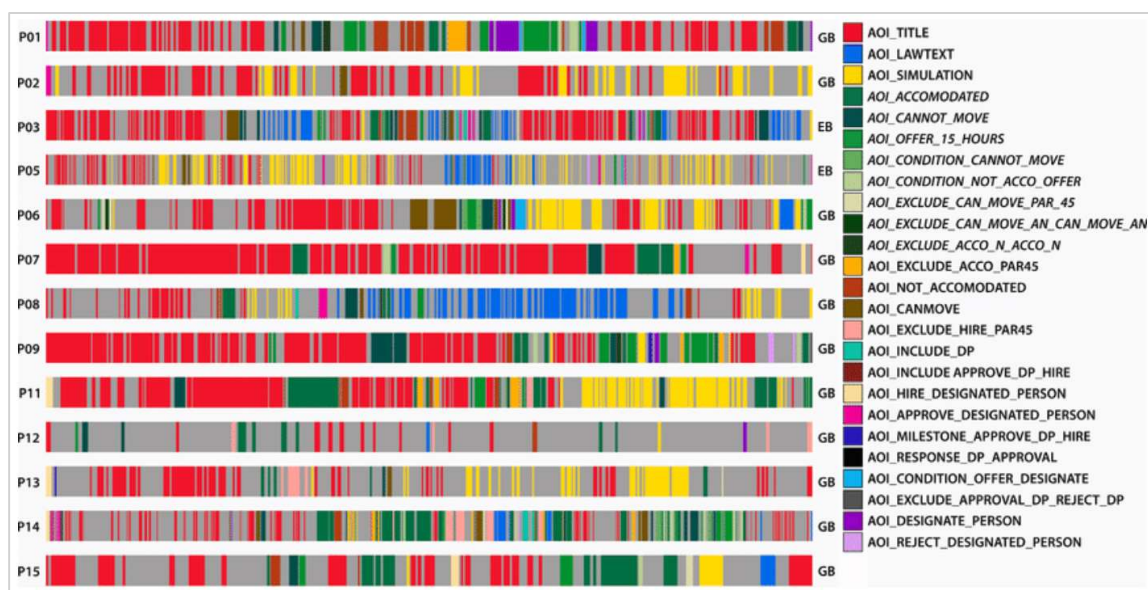
In Kombination mit Heat Maps und Gaze Plots kann der Opacity Plot jedoch eine wertvolle Ergänzung sein, um sowohl die aggregierten Blickdaten als auch individuelle Unterschiede zwischen Probanden sichtbar zu machen. Diese hybride Nutzung wird in der aktuellen Forschung zunehmend diskutiert, da sie eine umfassendere Analyse der Eye Tracking-Daten ermöglicht (Blascheck, et al., 2017).

2.2.3. AOI-based Methods

Im Gegensatz zu den point-based Methoden arbeiten, gemäss Blascheck (2017), die AOI-based Methode mit definierten Interessensbereichen. Das heisst, es werden nur Daten in

Bezug auf spezifische Regionen innerhalb eines Stimulus betrachtet. Dabei werden Metriken ausgewertet, die Auskünfte über die Verweildauer oder Anzahl der Fixationen im festgelegten AOI geben (Blascheck et al., 2017). Eine geläufige AOI-basierte Visualisierungstechnik ist gemäss Burch (2017) der Scarf Plot. Diese Darstellung wird verwendet, um die Sequenz und Dauer der Fixationen über verschiedene AOIs – Zeit dargestellt auf der X-Achse – von verschiedenen Probanden – untereinander dargestellt auf der Y-Achse – zu vergleichen. Abbildung 5 zeigt das Visualisierungsbeispiel eines Scarf Plots, der die Fixationssequenzen (X-Achse) verschiedener AOIs und Probanden (Y-Achse) zeigt, die eine definierte Aufgabe gelöst haben. Die AOIs sind farblich codiert.

Abbildung 5: Beispiel eines Scarf-Plots



Anmerkung: (ResearchGate, 2021)

Clustering

Clustering ist eine Methode des unüberwachten Lernens, die darauf abzielt, Datensätze in Gruppen (Cluster) aufzuteilen, wobei Datenpunkte innerhalb eines Clusters ähnliche Eigenschaften aufweisen. In der Eye-Tracking-Analyse wird Clustering verwendet, um Fixationen mit ähnlichen räumlichen oder zeitlichen Merkmalen zu gruppieren. Zum Beispiel können Cluster basierend auf den Fixationsdichten, den Dauerwerten oder den Sakkadenlängen gebildet werden (Tan, 2016)

Das Clustering ermöglicht es Forschenden, Muster in den Blickbewegungen zu erkennen, die sonst schwer zu interpretieren wären (Chen & Ling, 2003). Für die Visualisierung von Metroplänen könnte das Clustering beispielsweise verwendet werden, um Regionen zu identifizieren, die die Aufmerksamkeit der Betrachtenden besonders stark auf sich ziehen. Ein weiterer Vorteil des Clustering-Ansatzes ist seine Flexibilität, da er sowohl für kleinere Datenmengen als auch für grosse, komplexe Datensätze geeignet ist.

Zu den Herausforderungen des Clustering gehört die Wahl der richtigen Anzahl von Clustern. Die Anzahl der Cluster beeinflusst direkt die Qualität und Aussagekraft der Ergebnisse, da zu wenige Cluster wichtige Details übersehen und zu viele Cluster das Bild unnötig verkomplizieren können. Daher ist eine sorgfältige Auswahl und Validierung der Clusteranzahl essenziell, um die zugrunde liegenden Muster korrekt zu erfassen (Chen & Ling, 2003)

Ein Clustering-Problem kann formal wie folgt definiert werden:

Gegeben eine Menge von Datenpunkten $X=\{x_1, x_2, \dots, x_n\}$ IM \mathbb{R}^d Raum, wird das Ziel darin bestehen, k Cluster C_1, C_2, \dots, C_k zu finden, sodass:

1. Innerhalb-Cluster-Distanz minimiert wird:

Die Punkte innerhalb eines Clusters C_i sollten so nah wie möglich beieinander liegen. Dies wird häufig als Summe der quadrierten Abstände zwischen den Punkten und dem Clusterzentrum μ_i beschrieben:

Formel 1: Formen Intra Cluster Distanz

$$\text{Intra Cluster Distanz} = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

wobei $\mu_i = 1/|C_i| \sum_{x \in C_i} x$ das Zentrum des Clusters ist.

Anmerkung: (Tan, 2016)

2. Zwischen-Cluster-Distanz maximiert wird:

Die Clusterzentren μ_i und μ_j sollten so weit wie möglich voneinander entfernt sein, um eine klare Trennung zwischen den Clustern zu gewährleisten:

Formel 2: Formen Inter Cluster Distanz

$$\text{Inter Cluster Distanz} = \min \| \mu_i - \mu_j \|$$

Anmerkung: (Tan, 2016)

Diese Ziele können mit verschiedenen Algorithmen umgesetzt werden, die sich hinsichtlich ihrer Optimierungsmethode und Rechenkomplexität unterscheiden.

K-Means

Der K-Means-Algorithmus ist eine weit verbreitete Clustering-Methode, die durch eine iterative Optimierung die Datenpunkte in eine vordefinierte Anzahl von Clustern (k) gruppiert. Der Algorithmus basiert auf der Minimierung der Summe der quadrierten Abstände zwischen jedem Punkt und dem Zentrum seines zugehörigen Clusters (Tan, 2016)

- **Initialisierung der Clusterzentren**

Der erste Schritt besteht darin, k Clusterzentren zufällig oder durch eine Methode wie den K-means++-Ansatz zu initialisieren. Beim K-means++-Ansatz werden die initialen Clusterzentren so gewählt, dass sie gleichmässig über den Datensatz verteilt sind.

- **Zuweisung der Punkte zu Clustern**

In diesem Schritt wird jedem Punkt x_i aus der Datenmenge X das Cluster C_j zugewiesen, dessen Zentrum μ_j am nächsten liegt. Dies geschieht basierend auf der Intra-Cluster-Distanz. Das bedeutet, dass jeder Punkt zu dem Cluster gehört, dessen Zentrum am nächsten ist.

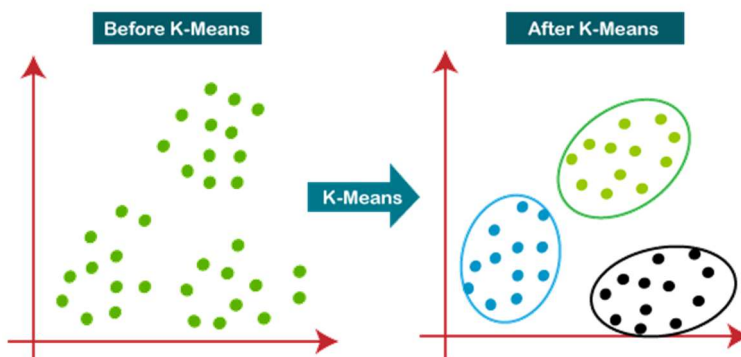
- **Berechnung der neuen Clusterzentren**

Nach der Zuweisung der Punkte zu den Clustern wird das Zentrum jedes Clusters aktualisiert. Das neue Zentrum μ_j wird als Mittelwert der Punkte im Cluster C_j berechnet.

- **Wiederholung**

Die Zuweisung der Punkte zu Clustern und das Berechnen neuer Clusterzentren wird iterativ wiederholt, bis sich keine signifikanten Änderungen mehr in den Clusterzuweisungen oder den Clusterzentren zeigen. Das bedeutet, dass die Zuweisungen der Punkte stabil bleiben oder die Clusterzentren nicht mehr weiter wandern.

Abbildung 6: Clustering mit K-Means



Anmerkung: (KeyToDataScience, 2024)

Im Kontext der Eye-Tracking-Daten kann K-Means verwendet werden, um Fixationen zu gruppieren, die geografisch oder zeitlich nahe beieinander liegen (Chen & Ling, 2003). Zum Beispiel können Cluster innerhalb eines Metroplans helfen, Bereiche zu identifizieren, die als Orientierungspunkte oder besonders auffällig wahrgenommen werden.

Ein Vorteil von K-Means ist seine Effizienz und einfache Implementierung. Der Algorithmus ist jedoch empfindlich gegenüber der Wahl der Anfangswerte und der Anzahl der Cluster. Um dieses Problem zu bewältigen, ist es wichtig, geeignete Methoden zur Validierung der Clusteranzahl anzuwenden, um sicherzustellen, dass die Gruppierungen die zugrunde liegenden Muster bestmöglich widerspiegeln.

2.3. Dashboard Design

Dashboards spielen eine zentrale Rolle bei der Verbindung von Eye-Tracking-Daten mit Visualisierungstechniken, um Daten zugänglich und nutzbar zu präsentieren (Burch & Schmid, 2023). Der theoretische Nutzen sowie die wesentlichen Designaspekte von Dashboards im Eye-Tracking-Kontext wurden bereits im Consultancy Project 1 (Nyffeler, Reiser, & Pellegatta, 2024) detailliert analysiert. Die folgenden Punkte greifen diese Ergebnisse auf und erweitern sie um weitere spezifische Aspekte.

2.3.1. Definition Dashboard

Dashboards werden oft mit einem Cockpit verglichen und dienen als visuelle Instrumententafel zur konsolidierten Darstellung von Informationen (Capone, 2015). Historisch wurde der Begriff im Kontext von Pferdekutschen verwendet, bevor er sich über die Automobilindustrie hinaus in den Geschäftskontext entwickelte (Park & Jo, 2015).

Im modernen Gebrauch versteht man unter einem Dashboard ein Werkzeug zur Visualisierung und Analyse von Daten, das Entscheidungsprozesse unterstützt. Die adressatengerechte Darstellung von Informationen ist jedoch oft eine Herausforderung, insbesondere angesichts wachsender Datenmengen (Malik, 2005). Interaktive Dashboards können diese Problematik überwinden, indem sie präzise und nutzerfreundliche Visualisierungen ermöglichen (Few, 2013).

Die Definition und historische Entwicklung von Dashboards, einschliesslich ihrer Anwendung in verschiedenen Domänen, wurde umfassend im Consultancy Project 1 (Nyffeler, Reiser, & Pellegatta, 2024) beschrieben und kann dort als Referenz herangezogen werden.

2.3.2. Nutzen von Dashboards

Dashboards sind sowohl in der Praxis als auch in der Wissenschaft essenziell: Sie dienen als Entscheidungs- und Analysewerkzeuge, wobei Visualisierungen eine entscheidende Rolle spielen (, 2015). Taschner Sie ermöglichen eine effizientere Informationsverarbeitung, indem sie visuelle Darstellungen von Daten bieten, die schneller erfasst und länger erinnert werden als reine Text- oder Zahlenformate (Burch, 2021).

Stephen Few unterscheidet drei Haupttypen von Dashboards:

1. **Strategische Dashboards:** Fokussieren auf hochrangige Kennzahlen und Prognosen.
2. **Analytische Dashboards:** Unterstützen die Datenanalyse mit detaillierten Darstellungen und Vergleichen.

- 3. Operative Dashboards:** Bieten Echtzeit-Daten zur Überwachung operativer Ziele (Few, 2013).

Die Vorteile und Einsatzmöglichkeiten dieser Dashboard-Typen wurden im Consultancy Project 1 (Nyffeler , Reiser , & Pellegatta, 2024) umfassend dargestellt und mit Beispielen untermauert.

2.3.3. Designregeln für Datenvisualisierungen

Eine gute Visualisierung hängt von der Wahrnehmung und den Bedürfnissen der Nutzer ab. Funktionalität und Ästhetik müssen harmonisiert werden, um sowohl die Lesbarkeit als auch die Akzeptanz zu maximieren (Burch & Schmid, 2023).

Jonathan Schwabish (2021) definiert fünf wesentliche Designregeln:

4. **Klarheit und Präzision:** Daten sollen deutlich präsentiert werden, um Entscheidungen zu unterstützen.
5. **Minimierung visueller Unordnung:** Überflüssige Elemente vermeiden, um die kognitive Belastung zu reduzieren.
6. **Integration von Grafiken und Text:** Beschreibungen und Titel sollen die Daten verständlich ergänzen.
7. **Vermeidung von "Spaghetti-Charts":** Komplexe und schwer lesbare Diagramme reduzieren.
8. **Gezielte Farbverwendung:** Neutrale Farben als Grundlage, gezielte Akzente durch zusätzliche Farben.

Diese Ansätze bauen auf den Prinzipien von Edward Tufte auf und wurden im Consultancy Project 1 (Nyffeler , Reiser , & Pellegatta, 2024) mit konkreten Beispielen für Eye-Tracking-Daten illustriert.

3. Methodik

3.1. Analyse der Rohdaten und Preprocessing

Der erste Schritt, um ein Dashboard erstellen zu können, ist die Analyse und Verarbeitung der Rohdaten. Eine gründliche Datenanalyse ist notwendig, um aussagekräftige Dashboards zu erstellen. Insbesondere die Explorative Datenanalyse bietet eine fundierte Basis für die Datenverarbeitung und Visualisierung (Turkey, 1977).

Der vorliegende Datensatz wurde von Dr. rer. nat Michael Burch zur Verfügung gestellt und stammt aus seinen Forschungsarbeiten im Bereich Eye Tracking. Der Datensatz umfasst Eye Tracking Daten verschiedener Probanden einer Studie und beinhaltet die wesentlichen Variablen, die Informationen zur Analyse von Blickbewegungen und Fixationspunkten liefern. Im Folgenden werden die einzelnen vordefinierten Variablen (Burch, 2024) detailliert beschrieben, um ein umfassendes Verständnis der Datensatzstruktur und ihrer Bedeutung zu ermöglichen.

- **Timestamp:** Diese Variable gibt den genauen Zeitpunkt der Datenaufnahme an. Jeder Eintrag im Datensatz ist mit einem spezifischen Zeitstempel versehen, was eine präzise zeitliche Zuordnung der Blickbewegungen ermöglicht.
- **StimuliName:** Der Name des Stimulus, auf den der Proband blickt, wird hier angegeben. In diesem Fall bezieht sich der Stimulus Name auf eine Metrokarte, die während des Eye Tracking-Experiments präsentiert wurde.
- **FixationIndex:** Diese Variable nummeriert die Fixationen chronologisch. Jede Fixation erhält eine eindeutige Nummer, die die Abfolge der Blickbewegungen angibt.
- **FixationDuration:** Die Dauer, wie lange der Blick auf einem bestimmten Punkt verweilt, wird in Millisekunden gemessen und in dieser Variable erfasst. Dies ist ein wichtiger Indikator für die Aufmerksamkeit und Verarbeitung des betrachteten Stimulus.
- **MappedFixationPointX und MappedFixationPointY:** Diese Variablen geben die genaue Position der Fixation auf dem Stimulus in einem zweidimensionalen Koordinatensystem an. Sie sind essenziell für die räumliche Analyse der Blickbewegungen.
- **user:** Diese Variable identifiziert den Probanden, der die Daten generiert hat.
- **description:** Diese Variable bezieht sich auf den Stimulus und beschreibt dessen Eigenschaften, wie zum Beispiel ob die Metrokarte in Farbe (color) oder in Graustufen (grey) dargestellt wurde. Diese Information ist wichtig für die Analyse, da sie Unterschiede in der visuellen Verarbeitung je nach Darstellungsmodus aufzeigen kann.

- **CityMap:** Diese Variable ist eine der selbst generierten Variablen und enthält den Namen der Stadtkarte ohne die Dateiendung «.jpg». Dies erleichtert die Identifikation der spezifischen Karten, die in den Experimenten verwendet wurden.

City: Auch dies ist eine generierte Variable und enthält den Namen der Stadt, jedoch ohne die Präfixe «S1» und «S2», die möglicherweise auf spezifische Szenarien oder Bedingungen hinweisen. Diese Bereinigung erleichtert die Analyse und Vergleichbarkeit der Daten.

Um die Analyse der Rohdaten weiter zu vertiefen, wurden zusätzliche Variablen aus den bestehenden Daten berechnet. Diese neuen Variablen bieten weiterführende Einblicke in das Blickverhalten der Probanden und ermöglichen eine detailliertere Analyse (Hastie et al., 2009). Im Folgenden werden diese neuen Variablen und ihre Berechnungsmethoden beschrieben:

- **Task Duration:** Diese Variable repräsentiert die Gesamtdauer der Fixationen eines Benutzers pro Karte. Sie wird berechnet, indem die Fixationsdauern für jede Karte und jeden Benutzer summiert werden. Diese Metrik gibt Auskunft über die Gesamtdauer, die ein Benutzer benötigt hat, um eine bestimmte Karte zu betrachten und zu verarbeiten.

Formel 3: Berechnung der Task Duration

$$Task\ Duration = \sum Fixation\ Duration\ [User, Karte]$$

Anmerkung: (Hastie et al., 2009)

- **Task Duration Kategorie:** Diese kategoriale Variable klassifiziert die Task-Dauer in zwei Kategorien:

K1 für Task-Dauern unter 10 Sekunden

K2 für Task-Dauern von 10 Sekunden oder mehr

Diese Kategorisierung hilft dabei, unterschiedliche Betrachtungsstrategien zu identifizieren und zu vergleichen (Schwellwert von 10 Sekunden basiert auf eigenen Annahmen).

Formel 4: Berechnung der Task Duration Kategorie

$$Task\ Duration\ Kategorie = \begin{cases} K1, & \text{wenn Task Duration unter 10 Sekunden} \\ K2, & \text{wenn Task Duration über 10 Sekunden} \end{cases}$$

Anmerkung: Eigene Definition

Saccade Length: Diese Variable berechnet die Länge der Sakkaden, also die Wegstrecke zwischen zwei aufeinanderfolgenden Fixationspunkten. Die Berechnung erfolgt mittels der euklidischen Distanzformel zwischen den Koordinaten der Fixationspunkte.

Formel 5: Berechnung der Sakkaden Länge

$$Saccade\ Length = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2}$$

Anmerkung: (Hastie et al., 2009)

Number Fixation Points: Diese Variable gibt die Anzahl der Fixationspunkte pro Benutzer und pro Karte an. Sie zählt die Fixationen, die ein Benutzer auf einer bestimmten Karte vorgenommen hat (reine Zählung im Datensatz).

Avg. Fixation Duration: Diese Variable gibt die durchschnittliche Fixationsdauer pro Benutzer und pro Karte an. Sie wird berechnet, indem die Gesamtdauer der Fixationen durch die Anzahl der Fixationspunkte dividiert wird.

Formel 6: Berechnung der Average Fixation Duration

$$Avg. Fixation\ Duration = \frac{Anzahl\ der\ Fixation\ Points}{Summe\ der\ Fixation\ Duration}$$

Anmerkung: (Burch, 2024)

Diese zusätzlichen Variablen ermöglichen eine umfassendere Analyse der Eye Tracking-Daten und bieten wertvolle Einblicke in das visuelle Such- und Betrachtungsverhalten der Probanden. Sie bilden die Grundlage für weiterführende Untersuchungen und die Erstellung aussagekräftiger Dashboards (Burch, 2024).

3.1.1. Exploration der Daten

Gemäss Auftraggeber, Dr. rer. nat. Michael Burch, haben wir die folgende Ausgangslage bei den Daten:

Stimuli

24 Metrokarten à jeweils 2 Ausrichtungen (S1 und S2) pro Farbausprägung:

→ 48 Stimuli in Farbe vs. 48 Stimuli in Grau

Probanden

40 Probanden haben an der Studie teilgenommen

Jeder Proband hat eine Kartenausprägung S1 in Farbe und eine Ausprägung S2 in Grau gesehen (oder umgekehrt): 24 Farbkarten vs. 24 Graukarten

Jeder Stimulus zählt Eye Tracking Daten von 20 Probanden

Weite Infos inklusiver Explorativer Datenanalyse sind erstlich im vorhergehenden Projekt (Nyffeler, Reiser, & Pellegatta, 2024).

3.2. Analyse der Anspruchsgruppen

Die Entwicklung und Implementierung von Dashboards in Organisationen erfordert eine sorgfältige Analyse der Anspruchsgruppen (Stakeholder), um sicherzustellen, dass die Bedürfnisse und Erwartungen aller relevanten Parteien erfüllt werden. Diese Analyse ist entscheidend, um Dashboards zu gestalten, die sowohl informativ als auch benutzerfreundlich sind, und um sicherzustellen, dass die bereitgestellten Daten den Entscheidungsprozess effektiv unterstützen (Raposo et al., 2022).

3.2.1. Identifikation der Anspruchsgruppen

Die erste Phase der Analyse besteht darin, alle potenziellen Anspruchsgruppen zu identifizieren, die von dem Dashboard profitieren könnten oder auf irgendeine Weise mit ihm interagieren werden (Freeman & Mcvea, 2001). Diese Gruppen können intern, wie Führungskräfte, Abteilungsleiter und Mitarbeiter, oder extern, wie Kunden, Partner und Investoren, sein. Die systematische Identifikation und Einbeziehung von Stakeholdern ist ein wesentlicher Bestandteil des strategischen Managements (Freeman & Mcvea, 2001).

3.2.2. Bedürfnisse und Anforderungen der Anspruchsgruppen

Die spezifischen Bedürfnisse jeder Gruppe, wie bevorzugte Daten und Darstellungen, werden analysiert. Die Berücksichtigung dieser Anforderungen ist entscheidend für die Benutzerakzeptanz (Davis, 1989).

3.2.3. Personalisierung und Anpassung

Basierend auf der Analyse der Bedürfnisse und Anforderungen können Dashboards personalisiert und angepasst werden, um spezifischen Anspruchsgruppen gerecht zu werden. Diese Anpassung kann die Auswahl relevanter Metriken, die Gestaltung benutzerfreundlicher Schnittstellen und die Integration von interaktiven Elementen umfassen. Benutzerzentrierte Designs verbessern die Effizienz und Zufriedenheit der Nutzer erheblich (Burmeister, 2008).

3.3. Vom Wireframe zum Mockup

Die Entwicklung eines Dashboards ist ein mehrstufiger Prozess, der typischerweise mit der Erstellung eines Wireframes beginnt und schließlich in einem detaillierten Mockup mündet (Few, 2013). Dieser Prozess stellt sicher, dass die Benutzeranforderungen und funktionalen Spezifikationen effektiv umgesetzt werden. Im Folgenden wird der Übergang vom Wireframe zum Mockup im Kontext von Dashboards detailliert beschrieben.

3.3.1. Wireframe-Erstellung

Ein Wireframe ist das erste visuelle Konzept eines Dashboards. Es dient als skizzenhafte Darstellung der grundlegenden Struktur und Layouts, ohne sich auf Design-Details wie Farben, Schriftarten oder genaue Abstände zu konzentrieren (Raposo et al., 2022). Das Hauptziel eines Wireframes ist es, die Anordnung und Hierarchie der Inhalte festzulegen, um sicherzustellen, dass alle notwendigen Informationen und Funktionen logisch und benutzerfreundlich platziert sind (Raposo et al., 2022).

Ein Wireframe für ein Dashboard könnte folgende Elemente umfassen (Few, 2013):

- Platzierung von Diagrammen und Grafiken
- Navigationselemente und Filteroptionen
- Platz für Überschriften und Beschreibungen
- Grundlegende Anordnung der Datenvisualisierungen

3.3.2. Mockup-Erstellung

Ein Mockup ist eine detailliertere und realistischere Darstellung des endgültigen Dashboards. Es baut auf dem Wireframe auf und integriert detaillierte Designelemente wie Farben, Typografie, Bilder und spezifische Stilrichtlinien. Mockups vermitteln einen präziseren Eindruck davon, wie das fertige Produkt aussehen und funktionieren wird (Raposo et al., 2022). Sie sind entscheidend, um Stakeholdern eine klare Vorstellung des Endprodukts zu geben und frühzeitiges Feedback zu ermöglichen.

Ein Mockup für ein Dashboard könnte folgende verfeinerte Elemente beinhalten (Few, 2013):

- Detaillierte Farbgestaltung und Typografie
- Konkrete Datendarstellungen mit Beispielwerten
- Interaktive Elemente wie Buttons und Dropdown-Menüs
- Endgültige Platzierung und Skalierung aller visuellen Komponenten

3.3.3. Bedeutung des Übergangs vom Wireframe zum Mockup

Der Übergang vom Wireframe zum Mockup verbessert die Kommunikation zwischen Designteams und Stakeholdern, minimiert Missverständnisse und sorgt für ein benutzerfreundliches Endprodukt (Raposo et al., 2022) (Few, 2013).

3.4. Dashboard Umsetzung mit Python Plotly

Die Entwicklung eines Dashboards mit Python und Plotly umfasst mehrere Schritte, von der Datenverarbeitung bis zur Bereitstellung, und basiert auf bewährten Methoden (Burch, 2021). Für Details zu einer ähnlichen Anwendung siehe Consultancy Project 1.

3.4.1. Datensatz einlesen und transformieren

Der erste Schritt besteht darin, die Rohdaten zu laden und zu transformieren, um sie für die Visualisierung vorzubereiten. Dies kann die Bereinigung der Daten, das Berechnen zusätzlicher Variablen und das Formatieren der Daten umfassen. Die Datenvorbereitung ist ein kritischer Schritt, der die Genauigkeit und Aussagekraft der nachfolgenden Analysen bestimmt (McKinney, 2017).

3.4.2. Dash

Dash ist ein Framework für die Erstellung von Webanwendungen mit Python. Es basiert auf Flask und Plotly und ermöglicht die Erstellung interaktiver Dashboards. Nach den Entwicklern von Plotly (Plotly, 2024) bietet Dash eine benutzerfreundliche Schnittstelle für die Entwicklung und das Hosting von Web-Apps.

3.4.3. Dash Core Components

Dash Core Components (dcc) sind grundlegende Bausteine für interaktive Elemente im Dashboard, wie Dropdown-Menüs, Schieberegler und Diagramme. Diese Komponenten ermöglichen die Erstellung dynamischer und interaktiver Benutzeroberflächen (Plotly, 2024).

3.4.4. Dash HTML Components

Dash HTML Components (html) bieten die Möglichkeit, HTML-Elemente wie Divs, Tabellen und Überschriften zu verwenden, um die Struktur und das Layout des Dashboards zu definieren. (Plotly, 2024).

3.4.5. Cascading Style Sheets (CSS)

CSS wird verwendet, um das Erscheinungsbild des Dashboards zu gestalten. Es ermöglicht die Anpassung von Farben, Schriftarten und Layouts, um eine ansprechende Benutzeroberfläche zu schaffen. CSS ist ein wesentliches Werkzeug für die Gestaltung von Webanwendungen und bietet flexible Designmöglichkeiten (Meyer, & Weyl, 2023).

3.4.6. Plotly Express

Plotly Express ist eine einfache und schnelle Möglichkeit, interaktive Diagramme und Graphen zu erstellen. Es bietet eine High-Level-API, die die Erstellung von Visualisierungen mit wenigen Codezeilen ermöglicht. Plotly Express ist ideal für die schnelle Entwicklung von Visualisierungen, die in Dash-Apps integriert werden können (Plotly, 2024).

3.4.7. Callbacks

Callbacks ermöglichen dynamische Interaktionen und Datenaktualisierungen durch Benutzeraktionen (Plotly, 2024). Callbacks sind das Herzstück der Interaktivität in Dash-Anwendungen und ermöglichen eine dynamische Datenaktualisierung (Plotly, 2024).

4. Ergebnisse

In diesem Kapitel werden die Ergebnisse der Erweiterung des Dashboards erläutert. Die folgenden Abschnitte geben einen umfassenden Überblick über den Anwendungsfall des Dashboards. Darüber hinaus wird die Entwicklung des Dashboards beschrieben sowie die neuen Interaktionen und Leistungsmerkmale vorgestellt.

4.1. Use Case

In diesem Abschnitt wird erläutert, wie die Erweiterung verwendet werden kann. Das entwickelte Dashboard zur Analyse von Eye-Tracking-Daten soll als vielseitiges Werkzeug für Forschende dienen, die sich mit Visualisierungswissenschaften beschäftigen. Es ermöglicht eine detaillierte Untersuchung und vergleichende Analyse farbiger und graustufenbasierter Metrokarten verschiedener Städte. Die Erweiterung erlaubt es, Vergleiche zwischen farbigen Metrokarten durchzuführen, wobei eine Variable gezielt geändert werden kann. Dadurch können farbige Metrokarten unter verschiedenen Parametern analysiert und untersucht werden. Dasselbe gilt für die graustufenbasierten Metrokarten. Das Dashboard dient dazu, Daten zu explorieren und zu interpretieren. Aus diesem Grund können die einzelnen Metrokarten nach Nutzern gefiltert werden. Dies ermöglicht es, identische Metrokarten unter verschiedenen Parametern miteinander zu vergleichen und neue Erkenntnisse zu gewinnen. Um den explorativen Nutzen weiter zu erhöhen, wurde eine Funktion integriert, die interessante Bereiche innerhalb der Karten anzeigt.

Eine zentrale Komponente des Dashboards ist die Integration von **k-means Clustering**, einem bewährten Verfahren zur Gruppierung von Fixationsdaten. Das Clustering ermöglicht es, Fixationen in Gruppen (Cluster) zu unterteilen, die auf ähnlichen räumlichen oder zeitlichen Merkmalen basieren. Mithilfe dieses Verfahrens können Forschende visuell interessante Bereiche identifizieren, die besonders viel Aufmerksamkeit der Betrachtenden auf sich ziehen. Beispielsweise können Regionen mit hoher Fixationsdichte oder längerer Verweildauer als sogenannte Hotspots hervorgehoben werden. Diese Informationen sind essenziell, um die Benutzerfreundlichkeit und Lesbarkeit von Metrokarten zu verbessern.

Die Anzahl der Cluster ist flexibel einstellbar, wodurch Forschende das Niveau der Analyse an die Komplexität ihrer Daten anpassen können. Wenige Cluster bieten eine grobe Übersicht, während eine grössere Anzahl detailliertere Muster offenlegt. Das Dashboard unterstützt die Wahl der optimalen Clusteranzahl durch Verfahren wie den Elbow-Test, um die Aussagekraft der Ergebnisse zu maximieren.

Die visuelle Darstellung der Cluster auf den Metrokarten hilft, Muster intuitiv zu erkennen. So können beispielsweise farblich codierte Bereiche anzeigen, welche Teile der Karte komplexe Haltestellenknoten darstellen oder farblich hervorstechend sind und dadurch die

Aufmerksamkeit der Nutzenden stärker anziehen. Ergänzend dazu können die Daten nach Nutzenden gefiltert werden, um die Unterschiede in der Wahrnehmung zwischen verschiedenen Benutzergruppen zu analysieren. Dies ermöglicht eine differenzierte Betrachtung und erleichtert die Ableitung von Erkenntnissen.

Zusammenfassend erweitert das k-means Clustering die Funktionalitäten des Dashboards erheblich, indem es Forschenden ermöglicht, Muster in Blickbewegungen zu erkennen, die sonst schwer interpretierbar wären. Durch die Kombination von explorativen Analysewerkzeugen und einer flexiblen Clusterbildung wird das Dashboard zu einem leistungsstarken Instrument, um Eye-Tracking-Daten effizient zu analysieren und visuelle Designs gezielt zu optimieren.

4.2. Entwicklung mit Python

Aufbauend auf dem bisherigen Dashboard wurden neue Visualisierungen und Interaktionen hinzugefügt. Der Hauptcode für das Dashboard wurde erweitert und befindet sich weiterhin in der Datei `app.py`. Die Designkomponenten wurden in die CSS-Datei `custom.css` ausgelagert und ebenfalls neugestaltet, um ein verbessertes Benutzererlebnis zu bieten.

4.2.1. Datenimport

Im ersten Schritt wird der Datensatz mit der Python-Bibliothek Pandas als CSV-Datei importiert. Der Datensatz bleibt unverändert, da sich die zugrunde liegenden Daten für das Analysetool nicht geändert haben.

4.2.2. Layout

Am Layout-Grundgerüst wurden keine Änderungen vorgenommen. Die erste Spalte bleibt unverändert und ist in zwei Bereiche unterteilt: Der obere Bereich dient der Eingabe von Benutzerdaten, der untere Bereich der Anzeige kleinerer Visualisierungen. Änderungen wurden ausschliesslich im dritten Container vorgenommen, in dem die neuen Visualisierungen hinzugefügt wurden.

Die mittlere Spalte definiert den Output-Bereich für die farbigen Metrokarten, während die dritte Spalte die graustufenbasierten Metrokarten darstellt. Dabei wurde den zweiten und dritten Spalten mehr Platz zugewiesen, da diese Visualisierungen im Fokus stehen.

Innerhalb dieser drei Spalten wurden sieben Container definiert. Deren Anordnung ist in Abbildung 7 schematisch dargestellt.

1. Container – Header Area: Wird verwendet, um die Titelzeile zu gestalten. Die Elemente in der Zeile sind mit dem Parameter `'display = flex'` definiert, um die Inhalte gleichmässig zu verteilen.

2. Container – Input City: Ist für das Dropdown Menü der City Map vorgesehen. Grundsätzlich haben alle Bereiche einen Rahmen und abgerundete Ecken. Mittels den Parametern 'padding' und 'margin' werden Innen- und Aussenabstände festgelegt.

3. Container – Input Visualization Type: Ist für die Auswahl des Visualisierungstyps vorgesehen. Die Klasse 'button_viz_type' gestaltet die Anordnung der Click-Buttons und die Klasse 'viz_button' definiert deren Aussehen, einschliesslich Hover-Effekt und aktiven Zuständen.

4. Container – KPI Table: Wird verwendet, um das Layout der KPI-Tabelle zu definieren.

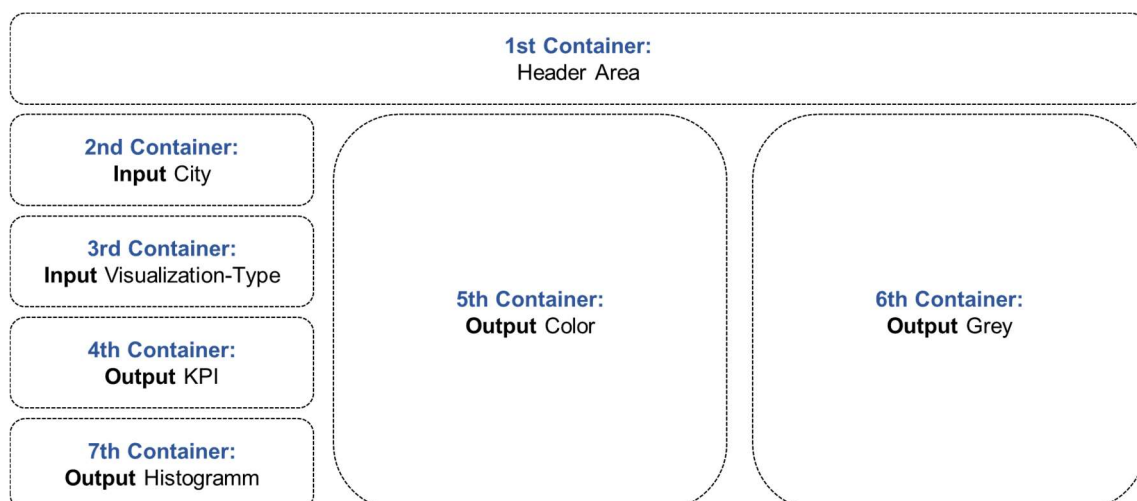
5. Container – Color Map: Wird verwendet, um die Visualisierungen der farbigen Karten darzustellen. Um den Visualisierungen so viel Platz wie möglich zu geben, wurden hier alle Innenabstände minimal gehalten.

6. Container – Grey Map: Ist analog dem fünften Container definiert und wird für die graustufen Visualisierungen verwendet.

7. Container – Histogramm: Wird verwendet, um ein Histogramm darzustellen. Diese Visualisierung ist weniger Platz eingeräumt, da sie mehr als Orientierung für den Output in den beiden Container 5 und 6 steht.

Für die neuen Visualisierungen zur Vergleichsanalyse wurden die Container 5 und 6 genutzt, um die farbigen und graustufigen Metrokarten doppelt anzuzeigen und so Vergleiche zu ermöglichen.

Abbildung 7 Anordnung der sieben Layout-Container



Anmerkung: Eigene Darstellung aus Consultancy Projekt 1

4.2.3. Interaktionen

In der dritten Sektion werden die notwendigen Dash-Anwendungen konfiguriert, damit die Benutzeroberfläche dynamisch auf Benutzereingaben reagiert und die Visualisierungen und Filteroptionen korrekt angezeigt werden. In **Fehler! Verweisquelle konnte nicht gefunden werden.** im Anhang ist der komplette Code dazu zu finden. Grundsätzlich sind die folgenden Komponenten enthalten:

- **Interaktive Buttons:** Aktivierung und Speicherung des Click-Buttons für die Wahl des Visualisierungstyps.
- **Anzeigen von Visualisierungen:** Abhängig vom aktiven Button wird die entsprechende Visualisierung im Container 5 und 6 dargestellt. Die Steuerung erfolgt über eine CSS-Klassenänderung.
- **Dropdown-Filter:** Die Dropdown-Filter für die Auswahl der Probanden werden dynamisch an die gewählte Städtekarte angepasst. Diese Funktion ist besonders relevant für die Heat Map und den Gaze Plot, da die angezeigten Probanden direkt von der ausgewählten Karte abhängen. Dadurch wird sichergestellt, dass nur die für die jeweilige Karte verfügbaren Daten zur Analyse herangezogen werden. Zusätzlich wurde für die Cluster-Analyse ein Filter hinzugefügt, der es ermöglicht, die Anzahl der Cluster flexibel anzupassen. Ausserdem erlaubt ein weiterer Filter die Anpassung der Darstellung der AOIs, sodass verschiedene Visualisierungsoptionen für die interessanten Bereiche bereitgestellt werden können.
- **Range-Slider:** Dynamische Anpassung der Range-Slider für die Filterung nach 'Task-duration', in Abhängigkeit der gewählten Städtekarte (relevant bei Heat Map und Gaze Plot).
- **Theme-Mode:** Wechsel zwischen einem hellen und dunklen Theme-Mode, wobei die entsprechenden CSS-Klassen angewendet werden, um das Aussehen der Anwendung zu ändern.

4.2.4. Visualisierungen

Die vorherigen Visualisierungen wurden beibehalten. In der vierten Sektion des Dashboards werden zehn Visualisierungen erstellt, die verschiedene Aspekte der Eye-Tracking-Daten analysieren. Dazu gehören eine **KPI-Tabelle**, die zentrale Kennzahlen darstellt, sowie **Gaze Plots** und **Heatmaps** für Fixationspunkte und -dichten, jeweils getrennt für Farbkarten und Graustufenkarten. **Box-Plots** zeigen die Verteilung der Bearbeitungsdauer und der durchschnittlichen Fixationsdauer, während ein **Histogramm** die Taskduration für einzelne oder alle Städte visualisiert. Schließlich analysieren **Scatter-Plots** die Korrelation zwischen

Fixationsdauer und Sakkadenlänge für beide Kartentypen. Alle Visualisierungen basieren auf einheitlichen Callbacks, die Benutzereingaben verarbeiten, Daten filtern und die Ergebnisse mit Plotly aktualisieren.

In der vierten Sektion werden die eigentlichen Visualisierungen erstellt. Neu wurden sechs Visualisierungen hinzugefügt. Die zusätzlichen Visualisierungs-Funktionen sind im Code 'app.py' enthalten und werden folgend kurz vorgestellt:

Cluster Analyse Color (4.1): Erstellt ein Cluster für Farbkarten, das die Fixationspunkte ausgewählter Benutzer innerhalb einer Stadt darstellt. Der Plot wird auf einem Hintergrundbild der jeweiligen Stadtkarte angezeigt. Zusätzlich wird ein AOI-Bereich (Area of Interest) darübergelegt, um besser zu verdeutlichen, welche Bereiche geclustert wurden.

Cluster Analyse Grey (4.2): Erstellt ein Cluster für graustufenbasierte Karten, das die Fixationspunkte ausgewählter Benutzer innerhalb einer Stadt darstellt. Der Plot wird auf einem Hintergrundbild der jeweiligen Stadtkarte angezeigt. Zusätzlich wird ein AOI-Bereich (Area of Interest) darübergelegt, um klarer darzustellen, welche Bereiche geclustert wurden.

Scarf Plot Color (4.3):

Erstellt eine Visualisierung der Fixationspunkte und deren Zuordnung zu verschiedenen AOIs, kategorisiert nach ausgewählten Benutzern innerhalb der Farbkarten. Der Plot basiert auf einem gestapelten Balkendiagramm, bei dem die x-Achse die Fixationsdauer (*FixationDuration*) über die Zeit darstellt und die y-Achse die Benutzer (*user*).

Die Balken werden farblich nach den entsprechenden AOI-Clustern unterschieden, wobei die Farben über die Mapping-Funktion `color_discrete_map` definiert werden. Die horizontale Darstellung (*orientation='h'*) sorgt für eine übersichtliche Visualisierung der Verteilung der Fixationsdauer über die verschiedenen Benutzer hinweg.

Scarf Plot Grey (4.4):

Erstellt eine Visualisierung der Fixationspunkte und deren Zuordnung zu verschiedenen AOIs, kategorisiert nach ausgewählten Benutzern innerhalb der graustufenbasierten Karten. Der Plot basiert auf einem gestapelten Balkendiagramm, bei dem die x-Achse die Fixationsdauer (*FixationDuration*) über die Zeit darstellt und die y-Achse die Benutzer (*user*).

Die Balken werden farblich nach den entsprechenden AOI-Clustern unterschieden, wobei die Farben über die Mapping-Funktion `color_discrete_map` definiert werden. Die horizontale Darstellung (*orientation='h'*) sorgt für eine übersichtliche Visualisierung der Verteilung der Fixationsdauer über die verschiedenen Benutzer hinweg.

Opacity Map Color (4.5):

Erstellt eine Visualisierung der Fixationspunkte innerhalb Farbkarten durch eine Dichte-Heatmap. Der Plot basiert auf den normalisierten Koordinaten der Fixationspunkte (NormalizedPointX und NormalizedPointY) und visualisiert die Dichte dieser Punkte. Die Heatmap verwendet 50 Bins entlang der X- und Y-Achsen (nbinsx=50, nbinsy=50), um die Daten in fein aufgelösten Dichtezonen darzustellen.

Die Farbskala der Heatmap ist so definiert, dass:

- Schwarz (rgba(0, 0, 0, 1)) Bereiche mit hoher Dichte darstellt,
- Halbtransparente Grautöne (rgba(255, 255, 255, 0) bei 0.5) mittlere Dichte markieren,
- Weisse, vollständig transparente Bereiche (rgba(255, 255, 255, 0) bei 1.0) Zonen ohne Fixationspunkte repräsentieren.

Diese Darstellung ermöglicht eine anschauliche Visualisierung der Konzentration der Fixationspunkte auf der Karte und hebt die Regionen mit hoher Benutzerinteraktion besonders hervor.

Opacity Map Grey (4.6):

Erstellt eine Visualisierung der Fixationspunkte innerhalb graustufigen Karten durch eine Dichte-Heatmap. Der Plot basiert auf den normalisierten Koordinaten der Fixationspunkte (NormalizedPointX und NormalizedPointY) und visualisiert die Dichte dieser Punkte. Die Heatmap verwendet 50 Bins entlang der X- und Y-Achsen (nbinsx=50, nbinsy=50), um die Daten in fein aufgelösten Dichtezonen darzustellen.

Die Farbskala der Heatmap ist so definiert, dass:

- Schwarz (rgba(0, 0, 0, 1)) Bereiche mit hoher Dichte darstellt,
- Halbtransparente Grautöne (rgba(255, 255, 255, 0) bei 0.5) mittlere Dichte markieren,
- Weisse, vollständig transparente Bereiche (rgba(255, 255, 255, 0) bei 1.0) Zonen ohne Fixationspunkte repräsentieren.

Diese Darstellung ermöglicht eine anschauliche Visualisierung der Konzentration der Fixationspunkte auf der Karte und hebt die Regionen mit hoher Benutzerinteraktion besonders hervor.

4.2.5. Ausführung der Applikation 'app.py'

Um die Applikation 'app.py' auszuführen, ist die Installation von Python Version 3.10 oder höher erforderlich. Zudem müssen folgende Packages installiert sein, wobei der Import der Bibliotheken bereits im Code 'app.py' enthalten ist:

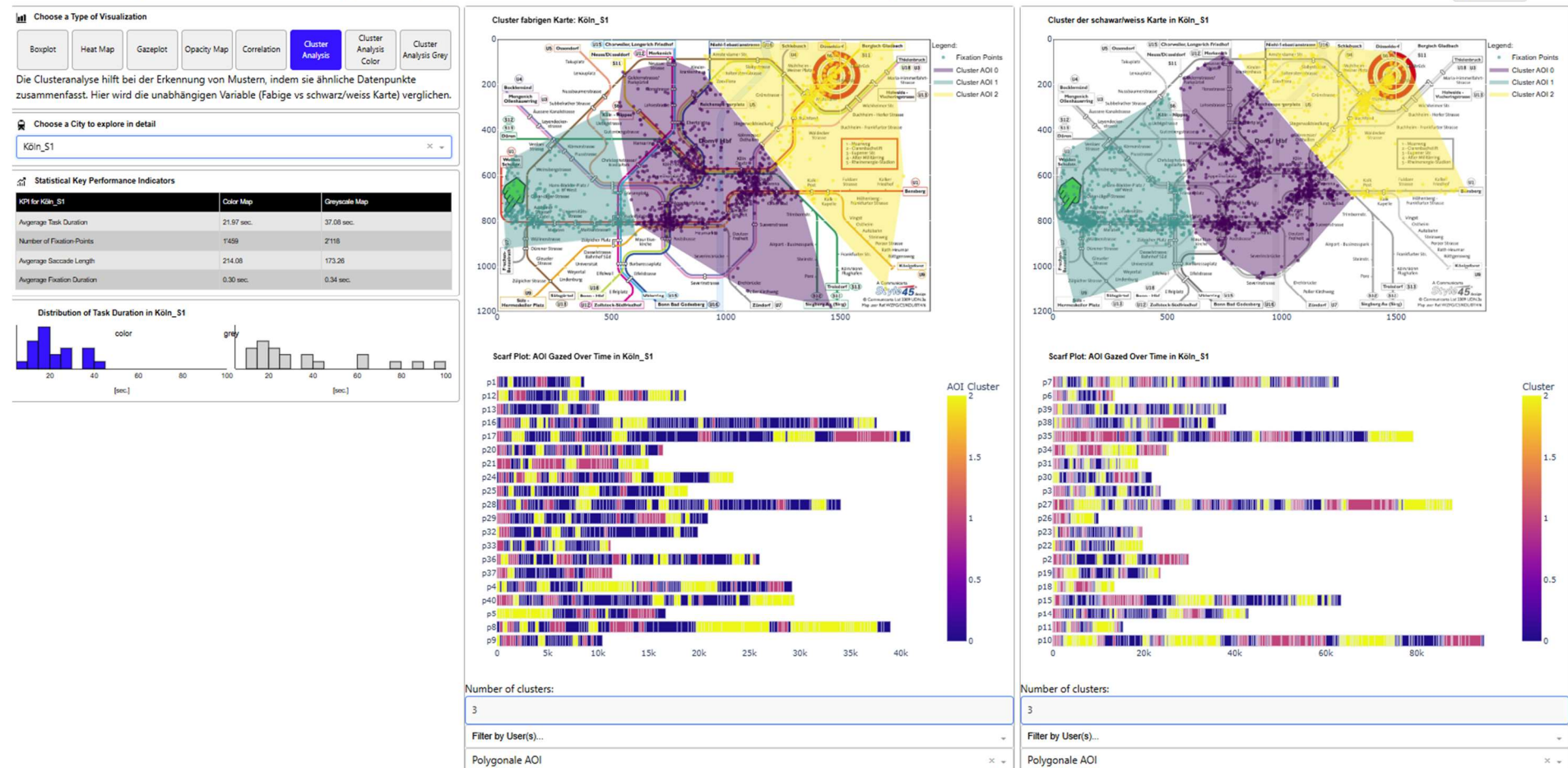
- from dash import Dash, dash_table, dcc, html, Input, Output, State, callback_context
- from dash_iconify import DashIconify
- from PIL import Image
- import dash_bootstrap_components as dbc
- import pandas as pd
- import plotly.express as px
- import glob
- import plotly.graph_objects as go

Die Applikation und für den Aufruf erforderlichen Mittel, wie Stylesheet 'custom.css', der Datensatz 'all_fixation_data_cleaned_up' sowie die Stadtkarten in jpg-Format, sind Teil des Github-Repositorys [≤ https://github.com/shaaark96/consultancy_2 >](https://github.com/shaaark96/consultancy_2). Das Repository ist öffentlich zugänglich.

Sobald die Applikation 'app.py' gestartet wird, öffnet sich die Dash-Anwendung automatisch im lokalen Webbrowser mit dem URL <http://127.0.0.1:8050/>. **Fehler! Verweisquelle konnte nicht gefunden werden.** Abbildung 8 zeigt das finale Dashboard, wie es beim Aufruf dargestellt wird.

Abbildung 8: Finales Dashboard

Clusterbasiervergleich räumlich zeitlichen Augenbewegungsdaten



Anmerkung: Eigene Darstellung

4.3. Erkenntnisse aus dem Dashboard

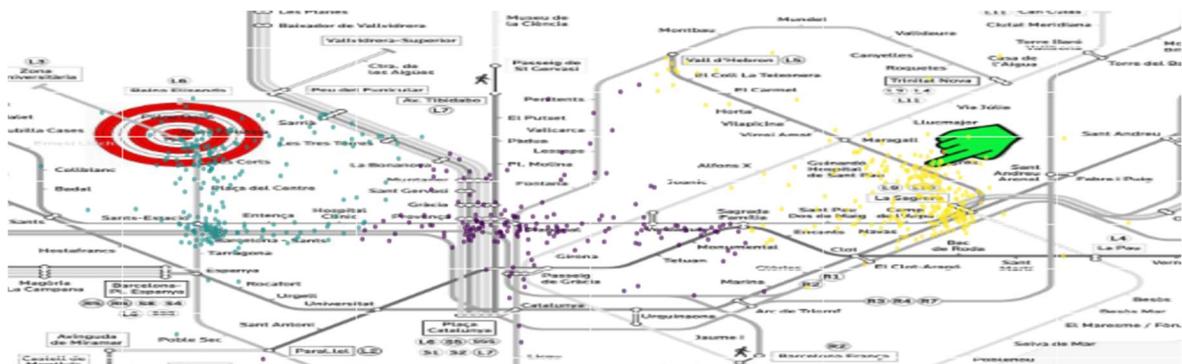
Bereits im vorherigen Projekt konnte der Boxplot einen ersten Überblick über Unterschiede in der Taskduration zwischen farbigen und Graustufenkarten liefern. Auffällig war, dass die Bearbeitungszeit bei der Graustufenkarte von Bologna fast doppelt so hoch war wie bei der farbigen Variante, während in Hamburg kaum Unterschiede erkennbar waren.

Auch der Boxplot auf der Landingpage des Dashboards bietet eine Übersicht über die Taskduration für verschiedene Stadtkarten. Er ermöglicht Forschenden, Rückschlüsse auf globale Unterschiede zu ziehen, wie etwa den deutlich höheren Median der Taskduration bei der Graustufenkarte von Bologna im Vergleich zu den farbigen Karten. Im Gegensatz dazu zeigen die Daten für Hamburg kaum Abweichungen zwischen den Kartentypen.

Ein Beispiel aus der Analyse der Metrokarte Barcelona S1 (Graustufenkarte) zeigt, dass die Karte in drei Bereiche unterteilt werden kann, die als AOIs den Stationsübergängen entsprechen. Der gelbe Bereich repräsentiert den ersten Stationsübergang, der violette Bereich den zweiten, und der grüne Bereich den dritten Stationsübergang. Die Analyse macht deutlich, dass der gelbe Bereich am meisten betrachtet wurde, gefolgt vom violetten Bereich, während der grüne Bereich vergleichsweise weniger Aufmerksamkeit erhielt.

Diese Erkenntnisse verdeutlichen, dass Nutzer ihre Aufmerksamkeit auf spezifische Bereiche der Karte konzentrieren, während andere weniger betrachtet werden. Solche Analysen sind entscheidend, um das Design der Karte gezielt an die Bedürfnisse der Nutzer anzupassen. Gleichzeitig lassen sich Muster im Nutzerverhalten erkennen, die Einblicke in die Nutzung der Karten und mögliche Optimierungsansätze liefern.

Abbildung 9 Barcelona S1 Grey Map mit Cluster



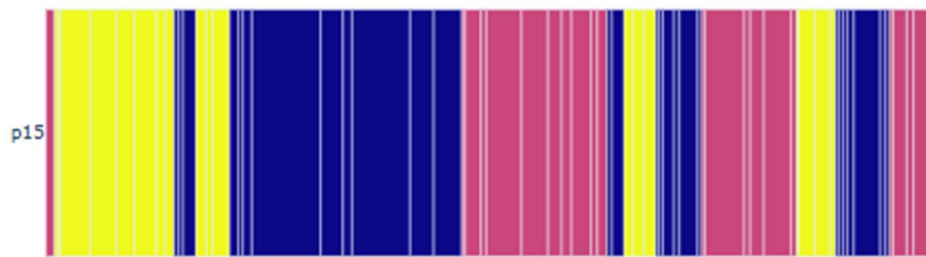
Anmerkung: Eigene Darstellung

Das Diagramm eignet sich besonders gut, um Unterschiede in den Fixationsdauern zwischen Usern und AOI-Clustern zu analysieren.

Es handelt sich um eine Cluster-Analyse der Grey Map von Barcelona S1 mit drei Clustern/AOIs. Auf der x-Achse sind die User dargestellt, während die Fixationsdauer

(FixationDuration) in einer horizontalen Skala angezeigt wird. Die Farbkodierung repräsentiert die Cluster, wobei jede Farbe einem bestimmten AOI-Cluster (Area of Interest) zuge-

Abbildung 10 User P15 Scarf Plot in Grey Map



ordnet ist.

Anmerkung: Eigene Darstellung

Einige User, wie P10, P1 und P35, weisen längere Gesamtdauern auf, während andere wie P6, P11, P18 und P26 vergleichsweise kurze Balken haben und die Aufgabe dadurch schneller gelöst haben.

User P15 benötigte im mittleren AOI deutlich mehr Zeit als alle anderen. Betrachtet man P15 genauer, wird ersichtlich, dass dieser sich häufiger im mittleren AOI (AOI 0) befand. Dies wird ebenfalls durch die Heatmap und den Gazeplot bestätigt. Es lässt sich vermuten, dass bei der Kreuzung Diagonal und Provenca Herausforderungen auftraten. Daher betrachtete P15 das Ziel erneut, bewegte sich in den AOI 1 und kehrte zurück in die Mitte (AOI 0) sowie schliesslich zum Ausgangspunkt. Diese Strecke wurde zweimal durchlaufen, vermutlich um sich abzusichern.

Am Ende wiederholt sich das Schema, was darauf schliessen lässt, dass User P15 die Strecke zweimal durchlief, bevor die Aufgabe beendet war.

User P11 löste die Aufgabe im Vergleich zu den anderen Probandinnen relativ schnell. User P11 durchlief die Strecke dreimal: Die ersten beiden Male führte der Weg vom Startpunkt zum Endpunkt, und beim dritten Mal ging die Strecke rückwärts und sprang anschliessend direkt vom Anfangspunkt zum Ziel. Diese Beobachtungen zeigen, dass es Probandinnen gibt, die die Strecke mehrfach durchlaufen und sie auch rückwärts gehen, um sicherzugehen.

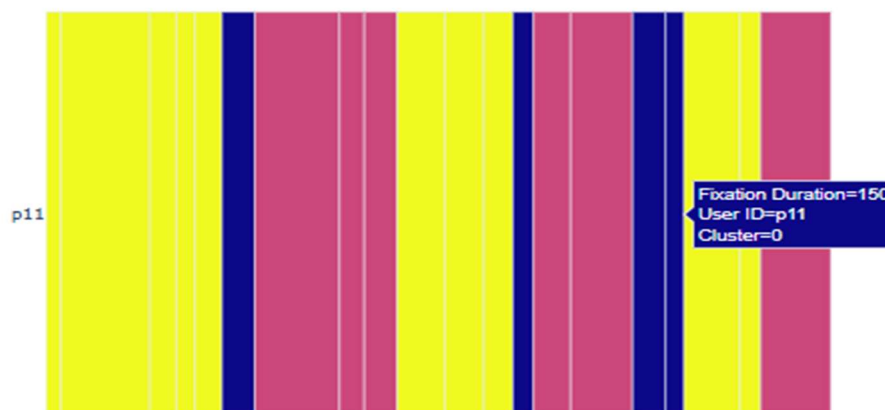
Die Analyse ermöglicht nicht nur den Vergleich zwischen schnellen und langsamen Usern, sondern auch zwischen den schnellsten Usern untereinander. Beispielsweise können User P6 und P11 verglichen werden, die beide die Aufgabe schnell gelöst haben. User P6 war jedoch schneller als P11. Im Gegensatz zu P11 durchlief P6 die Strecke nur zweimal, ohne sie rückwärts zu gehen und ohne den Start- und Endpunkt erneut zu betrachten.

Beim ersten Durchlauf betrachtete User P6 kurz den Anfang, blieb jedoch während des zweiten Durchgangs länger am Startpunkt. Ein möglicher Grund dafür könnte sein, dass die Station „La Sagrera“ überprüft wurde, um sicherzustellen, dass es sich um die richtige Strecke handelt, da das Bild graustufig ist. Anschliessend wurde die Strecke erneut durchlaufen, ohne zurückzugehen.

Mit Hilfe des Scarf-Plots und der Cluster-Analyse ist ersichtlich, wie lange sich ein User in einem Bereich aufhielt und wie oft der Bereich gewechselt wurde. Mit der Einbeziehung der Heatmap oder des Gazeplots können Bereiche identifiziert werden, die länger betrachtet wurden, insbesondere Kreuzungspunkte, an denen mehrere Strecken zusammenlaufen. Die Opacity Map ermöglicht es, betrachtete Bereiche sichtbar zu machen, anders als die Heatmap, die den Bereich überdeckt.

Cluster Analyse Grey Color von Barcelona S1 mit drei Cluster/AOIs.

Abbildung 11 User P11 Scarf Plot in Grey Map



Anmerkung: Eigene Darstellung

In dieser Visualisierung ist deutlich zu erkennen, dass AOI 1 und AOI 0 stark vertreten sind – dies entspricht dem Anfang und dem Ende der Strecke. Der mittlere Bereich, also AOI 2, ist hingegen weniger präsent. Dies könnte darauf hinweisen, dass durch die Farbkodierung die Strecke leichter zu verfolgen ist. Nur beim Wechsel der Farbe, beispielsweise an der Station „Sants Estació“, ist eine Häufung zu beobachten, weshalb AOI 0 stärker vertreten ist als AOI 1.

Es ist davon auszugehen, dass viele User von AOI 1 direkt zu AOI 0 wechseln und dabei den Streckenwechsel betrachten. Dies trifft auf einige User zu (z. B. P21, P29 und P36), die direkt vom Anfang zum Ziel springen und den mittleren Bereich überspringen. Allerdings gibt es auch User (z. B. P9, P37, P32, P33, P16 und P20), die der Linie/Strecke folgen. Diese bewegen sich zuerst durch AOI 1, dann AOI 2 und schliesslich zu AOI 0. Dabei verbringen sie anfangs ähnlich viel Zeit in den Bereichen. Daraus lässt sich schliessen, dass

der Startbereich (AOI 1) länger betrachtet wird, möglicherweise aufgrund von Unsicherheiten an der Station „La Sagrera“, wo die Strecke geändert werden könnte. Dies könnte daran liegen, dass die Linie L10 dieselbe Farbe aufweist und daher genauer betrachtet wird.

Im mittleren Bereich (AOI 2) verweilen die User weniger, da die Strecke hier geradlinig weiterführt und keine besonderen Entscheidungen getroffen werden müssen. Der Bereich AOI 0 ist wieder stärker ausgeprägt, da an der Station „Sants Estació“ ein Streckenwechsel erfolgt.

In dieser Visualisierung gibt es zwei User, die die Aufgabe am schnellsten gelöst haben: P1, P21 und P33. Diese waren im Vergleich zu den anderen Usern am schnellsten. Bei allen drei ist erkennbar, dass sie der Strecke entlang gegangen sind.

- User P21 betrachtete zuerst das Ende der Strecke, sprang dann zum Anfang und folgte anschliessend der Linie.
- User P33 folgte der Strecke und kehrte am Ende zum Anfang zurück.
- User P1 folgte ebenfalls der Strecke, untersuchte nach dem Ziel kurz den mittleren Bereich und betrachtete anschliessend erneut den Bereich der Station „Sants Estació“.

Ein weiterer interessanter Aspekt wäre, zu untersuchen, wie stark sich die Lernkurve der einzelnen User verändert. Dazu müsste die Visualisierung so gestaltet sein, dass nicht die Metrokarte selbst verglichen wird, sondern die Verhaltensmuster der User analysiert werden.

Zum Ende hin wäre es interessant, eine Möglichkeit zu haben, Verhaltensmuster jedes einzelnen Users zu ermitteln und diese zu gruppieren. Beispielsweise könnten User unterschieden werden, die der Linie/Strecke folgen, von denen, die direkt vom Start zum Ziel springen. Ebenso könnten jene identifiziert werden, die die Strecke mehrfach durchlaufen, um sie zu kontrollieren, oder die, die am Ende springen und sich rückwärts zum Start bewegen. Ausserdem könnten User gruppiert werden, die Schwierigkeiten beim Lesen der Karte haben, im Vergleich zu denen, die solche Probleme nicht zeigen.

5. Implikationen für die Praxis

Im vorhergehenden Kapitel wurden die Projektergebnisse und deren Entstehung erläutert. In diesem Kapitel geht es darum, den Praxisbezug näher zu beschreiben sowie Limitationen und Potenzial zur Weiterentwicklung aufzuzeigen.

5.1. Praxisbezug

Das erweiterte, webbasierte Dashboard bietet nun eine Vielzahl an Visualisierungsmöglichkeiten zur Darstellung und Analyse von Eye-Tracking-Daten. Mit der Integration neuer Funktionen wie der Cluster-Analyse, dem Scarfplot und der Opacity Map ermöglicht das Dashboard nicht nur den Vergleich zwischen farbigen und graustufenbasierten Karten, sondern auch detaillierte Analysen innerhalb dieser Kartentypen.

Besonders hervorzuheben ist die Möglichkeit, farbige Karten direkt gegenüberzustellen. Dabei können Benutzer und Cluster-Parameter dynamisch verändert werden, um spezifische Fragestellungen zu beantworten. Diese Funktion erweitert das Dashboard erheblich, da Forschende nun tiefergehende Vergleiche und Analysen durchführen können.

Die Cluster-Analyse visualisiert Fixationspunkte innerhalb bestimmter AOIs und hebt Regionen mit ähnlichen Blickbewegungen hervor. Der Scarfplot stellt die Fixationsdauer der Nutzer im Verhältnis zu den AOI-Clustern dar und erlaubt eine übersichtliche Darstellung, wie Benutzer verschiedene Bereiche der Karte betrachten. Mit der Opacity Map können zusätzlich Bereiche visualisiert werden, die besonders stark oder kaum beachtet wurden, was Rückschlüsse auf die Effektivität des Kartenlayouts zulässt.

Das Dashboard eröffnet Forschenden die Möglichkeit, nicht nur farbige und graustufenbasierte Karten zu vergleichen, sondern auch detaillierte Analysen innerhalb dieser Kartentypen durchzuführen. Forschende erhalten damit eine flexible Lösung, um Eye-Tracking-Daten umfassend und gezielt zu analysieren.

5.2. Limitation

Trotz der Erweiterungen weist das Dashboard einige Einschränkungen auf, die die Benutzerfreundlichkeit und die Aussagekraft der Ergebnisse beeinflussen können.

Erstens kann die Vielzahl an Visualisierungen und Parametern für unerfahrene Benutzer überwältigend wirken. Das Dashboard ist daher vor allem für Personen geeignet, die bereits mit Eye-Tracking-Daten vertraut sind.

Zweitens fehlen weitere Angaben zur statistischen Signifikanz von Unterschieden zwischen Datengruppen, wie beispielsweise Farbkarten und Graustufenkarten. Ohne diese

Information ist es schwierig, fundiertere Rückschlüsse aus den Analysen zu ziehen, insbesondere in wissenschaftlichen Kontexten.

Ein weiterer Punkt betrifft die Datenintegration. Der Datensatz ist fest im Dashboard eingebunden, was bedeutet, dass keine eigenen Daten hochgeladen werden können. Dies schränkt die Flexibilität des Tools ein, insbesondere da die Vorbereitung und Verarbeitung der Daten technisches Wissen erfordert.

Die Cluster-Analyse in Kombination mit dem Scarfplot bietet zwar wertvolle Einblicke in die Verteilung und Dauer der Fixationen innerhalb spezifischer AOIs, weist jedoch auch Limitationen auf. Die Datenaggregation der Cluster-Analyse führt dazu, dass detaillierte Informationen über individuelle Blickbewegungen und die zeitliche Abfolge der Fixationen verloren gehen. Der Scarfplot zeigt zwar die Fixationsdauer in den Clustern, bleibt jedoch bei der Darstellung dynamischer Aspekte, wie den Blickwechseln zwischen AOIs, eingeschränkt. Zudem wäre es hilfreich, mehrere Visualisierungen, wie etwa eine Cluster-Analyse, einen Gaze Plot und einen Scarfplot, gleichzeitig vergleichen zu können, um tiefere Einblicke in die Blickmuster einzelner Nutzer zu erhalten. Derzeit ist jedoch nur eine Visualisierung pro Karte möglich, was die Analyse komplexer Zusammenhänge erschwert.

Schliesslich liegt der Fokus des Dashboards derzeit ausschliesslich auf der Analyse von Metrokarten. Eine allgemeine Nutzbarkeit für andere Daten- oder Anwendungsbereiche ist in der aktuellen Version nicht gegeben.

5.3. Potenzial zur Weiterentwicklung

Im vorhergehenden Kapitel wurden die Grenzen des Dashboards aufgezeigt. Abschliessend werden die gewonnenen Erkenntnisse analysiert, um Entwicklungsmöglichkeiten darzulegen. Das Dashboard bietet eine solide Grundlage für zukünftige Erweiterungen, die seine Funktionalität und Benutzerfreundlichkeit weiter steigern können.

Ein wichtiger Ansatzpunkt ist die Integration statistischer Tests, um Unterschiede zwischen Datengruppen nicht nur visuell, sondern auch rechnerisch nachweisen zu können. Dies würde die Aussagekraft der Analysen erheblich verbessern und das Dashboard in wissenschaftlichen Kontexten noch nützlicher machen. Darüber hinaus könnten zusätzliche Metriken wie die Rückkehrzeit zu einem vorherigen AOI integriert werden, um noch detailliertere Einblicke in das Betrachtungsverhalten der Benutzer*innen zu ermöglichen.

Die Möglichkeit, eigene Datensätze hochzuladen, wäre ein weiterer entscheidender Schritt. Hierfür müssten klare Anleitungen bereitgestellt werden, die die Struktur der benötigten Daten und die Schritte zur Vorbereitung detailliert beschreiben. Eine solche Funktion würde das Dashboard vielseitiger und für eine grössere Zielgruppe zugänglich machen.

Auch erweiterte Filtermöglichkeiten könnten die Analyseoptionen vertiefen. Filter nach demografischen Merkmalen wie Alter oder Geschlecht würden es Forschenden erlauben, spezifischere Fragestellungen zu untersuchen, sofern diese Informationen im Datensatz vorhanden sind.

Ein weiteres wichtiges Feature wäre eine Berichtsfunktion, mit der Ergebnisse als PDF oder in anderen Formaten exportiert werden können. Diese Funktion würde es erleichtern, Analysen zu dokumentieren und zu präsentieren.

Das Dashboard kann durch zusätzliche Funktionen weiter optimiert und vielseitiger gestaltet werden. Ein vielversprechender Ansatz wäre die Integration weiterer Varianten der Cluster-Analyse, um verschiedene Perspektiven auf die Daten zu ermöglichen. Diese Vielfalt würde es Forschenden erlauben, je nach Fragestellung und Datenstruktur die geeignete Methode auszuwählen.

Ein weiterer wichtiger Entwicklungspunkt wäre die Möglichkeit, statistische Werte direkt im Dashboard auszugeben. Dies könnte beispielsweise Mittelwerte, Standardabweichungen oder Konfidenzintervalle für Fixationsdauer und Sakkadenlänge umfassen. Zusätzlich könnten statistische Signifikanztests wie t-Tests oder ANOVA integriert werden, um Unterschiede zwischen Gruppen rechnerisch nachzuweisen. Diese Ergänzungen würden die Aussagekraft der Analysen deutlich erhöhen und die Nutzbarkeit des Dashboards in wissenschaftlichen Kontexten stärken.

Auch die Interaktivität des Dashboards könnte weiter ausgebaut werden. Eine besonders nützliche Funktion wäre die Möglichkeit, eigene AOIs direkt im Dashboard zu erstellen und diese in Echtzeit zu analysieren. Forschende könnten so bestimmte Bereiche der Karte markieren und spezifische Analysen zu diesen Zonen durchführen, beispielsweise die Fixationsdauer oder die Häufigkeit von Blickbewegungen in den ausgewählten Bereichen. Dies würde die Flexibilität des Dashboards erheblich steigern und es den Benutzer*innen ermöglichen, ihre Analysen individuell anzupassen.

Zusammenfassend bietet das Dashboard grosses Potenzial für Weiterentwicklungen. Die Integration weiterer Cluster-Methoden, die Ausgabe statistischer Werte und die interaktive Erstellung eigener AOIs würden die Funktionalität und die Anwendungsbreite des Tools erheblich erweitern. Diese Ergänzungen könnten in zukünftigen Projekten realisiert werden und das Dashboard zu einem noch leistungstärkeren Werkzeug für die Analyse von Eye-Tracking-Daten machen. Die aktuellen Erweiterungen durch die Cluster-Analyse, den Scarfplot und die Opacity Map legen hierfür eine vielversprechende Grundlage, die in zukünftigen Projekten weiterentwickelt werden kann.

6. Literaturverzeichnis

- Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., & Ertl, T. (2017). Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum*, 36(8), 260–284. <https://doi.org/10.1111/cgf.13079>
- Brunyé, T. T., Drew, T., Weaver, D. L., & Elmore, J. G. (2019). A review of eye tracking for understanding and improving diagnostic interpretation. *Cognitive Research: Principles and Implications*, 4(1), 7. <https://doi.org/10.1186/s41235-019-0159-2>
- Burch, M. (2022). *Eye tracking and visual analytics* (1st Aufl.). River Publishers.
- Burch, M. (2024). *All_fixation_data_cleaned_up.csv* [CSV].
- Burch, M., & Schmid, M. (2023). *Dashboard Design*. River Publishers.
- Burmester, M. (2008). *Usability-Engineering*. https://link.springer.com/chapter/10.1007/978-3-540-69818-0_13
- Capone, R. (2015). *Nachhaltiges Vertriebscontrolling für Elektrotechniker: Integrationsmöglichkeiten von BSC und Management Cockpit in die Vertriebs- und Marketingsteuerung*. Springer Fachmedien Wiesbaden GmbH.
- Davis, F. (1989). *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*. <https://www.jstor.org/stable/249008>
- Duchowski, A. T. (2007). *Eye tracking methodology: Theory and practice* (2. ed). Springer.
- EyeSee Research. (2014, Mai 20). Eye Tracking Through History. *Medium*. <https://medium.com/@eyesee/eye-tracking-through-history-b2e5c7029443>
- Few, S. (2013). *Information dashboard design: Displaying data for at-a-glance monitoring* (Second edition). Analytics Press.
- Freeman, E., & Mcvea, J. (2001). *A Stakeholder Approach to Strategic Management*. University of Virginia. https://www.researchgate.net/publication/228320877_A_Stakeholder_Approach_to_Strategic_Management
- Grinberg, M. (2018). *Flask Web Development*. O'Reilly Media, Inc. <https://www.oreilly.com/library/view/flask-web-development/9781491991725/>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. <https://link.springer.com/book/10.1007/978-0-387-84858-7>
- Malik, S. (2005). *Enterprise dashboards: Design and best practices for IT*. Wiley.

- McKinney, W. (2017). *Python for Data Analysis*. O'Reilly Media, Inc. <https://www.oreilly.com/library/view/python-for-data/9781491957653/>
- Meyer, E., & Weyl, E. (2023). *CSS: The Definitive Guide*. O'Reilly Media, Inc. <https://www.oreilly.com/library/view/css-the-definitive/9781098117603/>
- Park, Y., & Jo, I.-H. (2015). Development of the Learning Analytics Dashboard to Support Students' Learning Performance. *Journal of Universal Computer Science*, Vol. 21(No. 1).
- Pfeifer, W., Braun, W., & Zentralinstitut für Sprachwissenschaft (Hrsg.). (1997). *Etymologisches Wörterbuch des Deutschen* (Ungekürzte, durchges. Ausg. der Taschenbuchausg., 2. Aufl.). Dt. Taschenbuch-Verl.
- Plotly. (2024). *Dash Documentation & User Guide | Plotly*. Plotly. <https://dash.plotly.com/>
- Podgorelec, V., & Kuhar, S. (2011). Taking Advantage of Education Data: Advanced Data Analysis and Reporting in Virtual Learning Environments. *Electronics And Electrical Engineering*, 114(8), 111–116. <https://doi.org/10.5755/j01.eee.114.8.708>
- Prillinger, K., Radev, S. T., Amador De Lara, G., Werneck-Rohrer, S., Plener, P. L., Poustka, L., & Konicar, L. (2023). Effects of Transcranial Direct Current Stimulation on Social Attention Patterns and Emotion Recognition Ability in Male Adolescents with Autism Spectrum Disorder. *Journal of Clinical Medicine*, 12(17), 5570. <https://doi.org/10.3390/jcm12175570>
- Rakoczi, G. (2012). Eye Tracking in Forschung und Lehre. Möglichkeiten und Grenzen eines vielversprechenden Erkenntnismittels. In *Digitale Medien—Werkzeuge für exzellente Forschung und Lehre: Bd. Medien in der Wissenschaft* (Nummer 61, S. 87–98). Waxmann : Münster u.a. <https://doi.org/10.25656/01:8301>
- Raposo, D., Silva, J., & Neves, João. (2022). *Perspectives on Design II*. <https://link.springer.com/book/10.1007/978-3-030-79879-6>
- ResearchGate. (2021). *ResearchGate—Scarf-plot*. ResearchGate. https://www.researchgate.net/figure/Scarf-plot-showing-the-sequences-of-fixations-for-participants-solving-a-scenario-task_fig4_342713292
- Schwabish, J. A. (2021). *Better data visualizations: A guide for scholars, researchers, and wonks*. Columbia University Press.
- Tang, L. (2016, Dezember 5). Subtleties of Eyetracking Heat Maps and Gaze Plots. *Medium*. <https://medium.com/@TheRealTang/subtleties-of-eyetracking-heat-maps-and-gaze-plots-a7ba4207f20f>

Taschner, A. (2015). *Management Reporting und Behavioral Accounting: Verhaltenswirkungen des Berichtswesens im Unternehmen*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-07699-3>

Turkey. (1977). *Exploratory Data Analysis* |. SpringerLink. https://link.springer.com/referenceworkentry/10.1007/978-0-387-32833-1_136

usability.de GmbH & Co. KG. (2024). *Eye Tracking—Usability-Test mit Eye Tracking*. usability.de. <https://www.usability.de/leistungen/ux-testing-nutzerforschung/eyetracking.html>

Wedel, Michel & Pieters, Rik. (2008). *A Review of Eye-Tracking Research in Marketing. Volume 4*, 123–147.

7. Hilfsmittelverzeichnis

Python Bibliotheken:

- dash
- dash_iconify
- PIL
- dash_bootstrap_components
- pandas
- plotly express
- glob
- plotly graph object

ChatGPT für die Fehlerfindung und Optimierung im Python Code

Fachliteratur gemäss Literaturverzeichnis in Kapitel 6

8. Anhang

Codeausschnitt 1: Komplette Layout Definition ('custom.css')

```
/* 1st Container - Header and Theme-Mode */
.first_container {
  width: 90%;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding-top: 6px;
  padding-bottom: 1px;
  padding-left: 10px;
  box-sizing: border-box;
}
/* Container für Header (H1 und H4) */
.header {
  display: flex;
  background-color: #aa503c;
  align-items: center;
  gap: 360px; /* Abstand zwischen H1 und H4 */
}
/* Dropdown für Theme-Mode */
.theme_dropdown {
  width: 220%;
  font-size: 14px;
}
/* ----- */
/* Split Layout in three columns */
.dash_container {
  display: flex;
  flex-direction: row;
  width: 90%;
  gap: 1px;
  box-sizing: border-box;
  padding-bottom: 10px;
}
/* Erste Spalte */
.first_column {
  flex: 0 0 25%;
  display: flex;
  flex-direction: column;
  box-sizing: border-box;
}
/* Zweite Spalte */
.second_column {
  flex: 0 0 37.5%;
  display: flex;
  flex-direction: column;
  box-sizing: border-box;
}
/* Dritte Spalte */
.third_column {
  flex: 0 0 37.5%;
  display: flex;
  flex-direction: column;
  box-sizing: border-box;
}
/* ----- */
/* 2nd Container - Input City */
.second_container {
  display: block;
  align-items: center;
  border: 1px solid darkgrey;
  border-radius: 5px;
  padding: 10px;
  gap: 2px;
  margin-left: 3px;
  margin-right: 3px;
}
```

```

        margin-top: 5px;
        box-sizing: border-box;
    }
    .dropdown {
        width: 100%;
        font-size: 14px;
        cursor: pointer;
    }
    .dash-dropdown .Select-placeholder {
        font-family: Arial, sans-serif;
        font-style: normal;
        font-size: 14px;
    }
    /* ----- */
    /* 3rd Container - Input Visualization Type */
    .third_container {
        display: block;
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        padding: 10px;
        gap: 2px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    .button_viz_type {
        display: flex;
        gap: 1px;
        justify-content: left;
        border-radius: 5px;
    }
    .viz_button {
        padding: 3px 3px;
        width: 80px;
        font-size: 12px;
        border: 1px solid darkgrey;
        border-radius: 5px;
        margin: 2px;
    }
    .viz_button:hover {
        background-color: #e0e0e0;
        cursor: pointer;
    }
    .viz_button.active {
        background-color: #0000FFFF;
        color: white;
        border: 1px solid blue;
    }
    /* ----- */
    /* 4th Container - KPI-Table */
    .fourth_container {
        display: block;
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        padding: 10px;
        gap: 2px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        padding-bottom: 10px;
        box-sizing: border-box;
    }
    /* ----- */
    /* 5th Container - Color Map */
    .fifth_container {
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
    }

```

```

        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    /* ----- */
    /* 6th Container - Grey Map */
    .sixth_container {
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    /* 7th Container - Histogram */
    .seventh_container {
        display: block;
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        padding: 10px;
        gap: 2px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    /* 8th Container - Histogram */
    .eighth_container {
        display: block;
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        padding: 10px;
        gap: 2px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    /* 9th Container - Histogram */
    .ninth_container {
        display: block;
        align-items: center;
        border: 1px solid darkgrey;
        border-radius: 5px;
        padding: 10px;
        gap: 2px;
        margin-left: 3px;
        margin-right: 3px;
        margin-top: 5px;
        box-sizing: border-box;
    }
    }

    /* ----- */
    /* Theme-Mode */
    .light_theme {
        background-color: white;
        color: black;
    }
    .dark_theme {
        background-color: #454a59;
        color: white;
    }
    }
    html, body {
        background-color: #e3f2fd;
        color: inherit;
    }
}

```

```

.header, .first_container, .second_container, .third_container, .fourth_container,
.fifth_container, .sixth_container, .seventh_container, .dash_container {
    background-color: inherit;
    color: inherit;
}
/* Dropdown für Theme-Mode */
/* Überschreiben der Standardklasse von Dash */
.light_theme .dash-dropdown,
.light_theme .dash-dropdown .Select-control,
.light_theme .dash-dropdown .Select-menu-outer {
    background-color: white !important;
    color: black !important;
}
.dark_theme .dash-dropdown,
.dark_theme .dash-dropdown .Select-control,
.dark_theme .dash-dropdown .Select-menu-outer {
    background-color: #485260 !important;
    color: white !important;
}
/* Schriftfarbe für den ausgewählten Wert im Light Theme */
.light_theme .dash-dropdown .Select-placeholder,
.light_theme .dash-dropdown .Select-value-label {
    color: black !important;
}
/* Schriftfarbe für den ausgewählten Wert im Dark Theme */
.dark_theme .dash-dropdown .Select-placeholder,
.dark_theme .dash-dropdown .Select-value-label {
    color: white !important;
}
/* ----- */
/* Font Style */
h1 {
    font-size: 24px;
    font-family: Arial, sans-serif;
    font-weight: bold;
    text-align: left;
    /* margin-bottom: 12px; */
}
h2 {
    font-size: 16px;
    font-family: Arial, sans-serif;
    font-weight: normal;
    text-align: left;
    /* margin-bottom: 12px; */
}
h3 {
    font-size: 12px;
    font-family: Arial, sans-serif;
    font-weight: bold;
    text-align: left;
    /* margin-bottom: 12px; */
}
h4 {
    font-size: 12px;
    font-weight: lighter;
    font-style: italic;
    margin-top: 10px;
}
p {
    font-size: 12px;
    font-family: Arial, sans-serif;
    font-weight: normal;
    text-align: left;
    margin-top: 6px;
    margin-bottom: 1px;
    margin-left: 12px;
}
#selected_n_clusters_color {
    border: 2px solid #007bff; /* Blue border */
    border-radius: 5px;      /* Rounded corners */
    padding: 10px;           /* Inner padding */
}

```

```

    font-size: 14px;          /* Font size */
    color: #333;              /* Text color */
    background-color: #f9f9f9; /* Light gray background */
    width: 100%;              /* Set width */
}

/* Add hover effect */
#selected_n_clusters_color:hover {
    border-color: #0056b3; /* Darker blue on hover */
    background-color: #e9f5ff; /* Light blue background on hover */
}

/* Add focus effect */
#selected_n_clusters_color:focus {
    outline: none;          /* Remove default outline */
    border-color: #0056b3; /* Darker blue on focus */
    box-shadow: 0 0 5px rgba(0, 123, 255, 0.5); /* Add shadow on focus */
}

#selected_n_clusters_grey {
    border: 2px solid #007bff; /* Blue border */
    border-radius: 5px;        /* Rounded corners */
    padding: 10px;             /* Inner padding */
    font-size: 14px;          /* Font size */
    color: #333;              /* Text color */
    background-color: #f9f9f9; /* Light gray background */
    width: 100%;              /* Set width */
}

/* Add hover effect */
#selected_n_clusters_grey:hover {
    border-color: #0056b3; /* Darker blue on hover */
    background-color: #e9f5ff; /* Light blue background on hover */
}

/* Add focus effect */
#selected_n_clusters_grey:focus {
    outline: none;          /* Remove default outline */
    border-color: #0056b3; /* Darker blue on focus */
    box-shadow: 0 0 5px rgba(0, 123, 255, 0.5); /* Add shadow on focus */
}

```

Codeausschnitt 2: Komplette HTML-Layout Definition ('app.py')

```

-----
Section 2:
Definition of Dash Layout
"""
app.layout = html.Div([
    # Header and Theme-Mode
    html.Div([
        html.Div([
            html.H1('Clusterbasiervergleich räumlich zeitlichen Augenbewegungsdaten'),
            html.H4('created on behalf of FHGR Chur, last updated: 17.01.2025']],
            className='header'),
        dcc.Dropdown(
            id='theme_dropdown',
            options=[
                {'label': 'Light Mode', 'value': 'light'},
                {'label': 'Dark Mode', 'value': 'dark'}
            ],
            value='light',
            clearable=False,
            className='theme_dropdown',
        ),
        dcc.Store(id='current_theme', data='light'),
    ], className='first_container'),

    html.Div([

```

```

# First column: Input, KPI, Histogram
html.Div([
    # Visualization Type Buttons
    html.Div([
        html.H3([
            DashIconify(icon="ion:bar-chart", width=16, height=16, style={"margin-
right": "12px"}),
            'Choose a Type of Visualization']],
        html.Div([
            html.Button('Boxplot', id='default_viz', n_clicks=0, className='viz_but-
ton'),
            html.Button('Heat Map', id='heat_map', n_clicks=0, className='viz_but-
ton'),
            html.Button('Gazeplot', id='gaze_plot', n_clicks=0, className='viz_but-
ton'),
            html.Button('Opacity Map', id='opacity_map', n_clicks=0, class-
Name='viz_button'),
            html.Button('Correlation', id='scatter_plot', n_clicks=0, class-
Name='viz_button'),
            html.Button('Cluster Analysis', id='cluster_analysis', n_clicks=0, class-
Name='viz_button'),
            html.Button('Cluster Analysis Color', id='cluster_analysis_map_color',
n_clicks=0,
                        className='viz_button'),
            html.Button('Cluster Analysis Grey', id='cluster_analysis_map_grey',
n_clicks=0,
                        className='viz_button'),
            ], id='button_viz_type', className='button_viz_type'),
        dcc.Store(id='active-button', data='default_viz'),
        html.Div(id='output-section'),
    ], className='third_container'),

    # City Selection Dropdown
    html.Div([
        html.H3([
            DashIconify(icon="vaadin:train", width=16, height=16, style={"margin-
right": "12px"}),
            'Choose a City to explore in detail']],
        dcc.Dropdown(
            id='city_dropdown',
            options=[{'label': city, 'value': city} for city in
sorted(df['CityMap'].unique())],
            placeholder='Select a City Map...',
            value=None,
            clearable=True,
            className='dropdown'
        ),
    ], className='second_container'),

    # KPI Table
    html.Div([
        html.H3([
            DashIconify(icon="fluent:arrow-trending-lines-24-filled", width=16,
height=16,
                        style={"margin-right": "12px"}),
            'Statistical Key Performance Indicators']],
        html.Div(id='table_container')
    ], className='fourth_container'),

    # Histogram
    html.Div([
        dcc.Graph(id='hist_taskduration'),
    ], className='seventh_container'),
], className='first_column'),

# Second column: Color Map
html.Div([
    html.Div([
        html.Img(id='city_image_color'),
        dcc.Graph(id='gaze_plot_color'),
        dcc.Graph(id='heat_map_color'),

```



```

        dcc.Dropdown(id='dropdown_user_color', multi=True),
        dcc.RangeSlider(id='range_slider_color', min=1, max=50, step=1, value=[1, 50],
            marks={i: f'{i}' for i in range(0, 51, 5)}),
        dcc.Graph(id='box_task_duration'),
        dcc.Graph(id='scatter_correlation_color'),
        dcc.Graph(id='cluster_analysis_color'),
        dcc.Graph(id='scarf_plot_color'),
        dcc.Graph(id='opacity_map_color'),
        dcc.Graph(id='cluster_analysis_map_color_1'),
        dcc.Graph(id='cluster_analysis_map_grey_1'),
        dcc.Input(id='selected_n_clusters_color'),
        dcc.Dropdown(id='selected_aoi_type_color'),
    ], id='color_plot_area', className='fifth_container'),
], className='second_column'),

# Third column: Grey Map
html.Div([
    html.Div([
        html.Img(id='city_image_grey'),
        dcc.Graph(id='gaze_plot_grey'),
        dcc.Graph(id='heat_map_grey'),
        dcc.Dropdown(id='dropdown_user_grey', multi=True),
        dcc.RangeSlider(id='range_slider_grey', min=1, max=50, step=1, value=[1, 50],
            marks={i: f'{i}' for i in range(0, 51, 5)}),
        dcc.Graph(id='box_avg_fix_duration'),
        dcc.Graph(id='scatter_correlation_grey'),
        dcc.Graph(id='cluster_analysis_grey'),
        dcc.Graph(id='scarf_plot_grey'),
        dcc.Graph(id='opacity_map_grey'),
        dcc.Graph(id='cluster_analysis_map_color_2'),
        dcc.Graph(id='cluster_analysis_map_grey_2'),
        dcc.Input(id='selected_n_clusters_grey'),
        dcc.Dropdown(id='selected_aoi_type_grey')
    ], id='grey_plot_area', className='sixth_container'),
], className='third_column'),
], className='dash_container'),
], id='page_content', className='light_theme')

```

Codeausschnitt 3: Cluster Analyse Color ('app.py')

```

@app.callback(
    [Output('cluster_analysis_color', 'figure'),
     Output('scarf_plot_color', 'figure')],
    [Input('city_dropdown', 'value'),
     Input('dropdown_user_color', 'value'),
     Input('selected_n_clusters_color', 'value'),
     Input('selected_aoi_type_color', 'value'),
     Input('current_theme', 'data')]
)
def cluster_analysis_color(selected_city, selected_users, selected_n_clusters_color, selected_aoi_type_color,
                           current_theme):
    title_color = 'black' if current_theme == 'light' else 'white'

    if not selected_city:
        fig_empty = go.Figure()
        fig_empty.update_layout(
            title={
                'text': "No City Map selected.<br><br>"
                "To display the <b>Cluster</b> on a specific map,<br>"
                "please select a city from the dropdown on the left.",
                'y': 0.6,
                'x': 0.5,
                'xanchor': 'center',
                'yanchor': 'middle',
                'font': {
                    'size': 14,
                    'color': title_color,
                    'family': 'Arial, sans-serif'
                }
            }
        )

```

```

    }
    },
    paper_bgcolor='rgba(0, 0, 0, 0)',
    plot_bgcolor='rgba(0, 0, 0, 0)',
    xaxis=dict(visible=False),
    yaxis=dict(visible=False),
    height=425
)
return fig_empty, fig_empty

# Filter data based on user and city selections
filtered_df = df[(df['CityMap'] == selected_city) & (df['description'] == 'color')]

if selected_users:
    if isinstance(selected_users, str):
        selected_users = [selected_users]
    filtered_df = filtered_df[filtered_df['user'].isin(selected_users)]

# Get image dimensions
image_path_color, width, height = get_image_path_color(selected_city)
if not (image_path_color and width and height):
    return go.Figure(), go.Figure()

# Normalize fixation points within image dimensions
filtered_df['NormalizedPointX'] = filtered_df['MappedFixationPointX']
filtered_df['NormalizedPointY'] = filtered_df['MappedFixationPointY']
filtered_df = filtered_df[
    (filtered_df['NormalizedPointX'] >= 0) & (filtered_df['NormalizedPointX'] <= width) &
    (filtered_df['NormalizedPointY'] >= 0) & (filtered_df['NormalizedPointY'] <= height)
]

if filtered_df.empty:
    return go.Figure(), go.Figure()

coords = filtered_df[['NormalizedPointX', 'NormalizedPointY']]
kmeans, labels = update_clusters_grey(coords, selected_n_clusters_color)
if kmeans is None:
    return go.Figure(), go.Figure()

# Cluster und Farben updaten
coords = filtered_df[['NormalizedPointX', 'NormalizedPointY']]
kmeans, labels = update_clusters_grey(coords, selected_n_clusters_color)

if kmeans is None:
    return go.Figure(), go.Figure()

filtered_df['AOI Cluster'] = labels
cluster_colors = generate_cluster_colors(selected_n_clusters_color)

# Create Cluster Analysis Figure
fig_cluster = go.Figure()
fig_cluster.add_trace(go.Scatter(
    x=coords['NormalizedPointX'],
    y=coords['NormalizedPointY'],
    mode='markers',
    marker=dict(
        color=[cluster_colors[label] for label in labels],
        size=5,
        opacity=0.6
    ),
    hoverinfo='skip',
    text=[f'Cluster: {label}' for label in labels],
    name='Fixation Points'
))
fig_cluster = add_aoi_visualization_color(
    fig_cluster, coords, labels, cluster_colors, selected_n_clusters_color, selected_aoi_type_color
)
fig_cluster.add_layout_image(
    dict(
        source=image_path_color,

```

```

        x=0,
        size=width,
        y=0,
        size=height,
        xref="x",
        yref="y",
        sizing="stretch",
        opacity=1,
        layer="below"
    )
)

fig_cluster.update_layout(
    title=dict(
        text=f'<b>Cluster fabrigen Karte: {selected_city}</b>',
        font=dict(size=12, family='Arial, sans-serif', color=title_color)
    ),
    margin=dict(l=50, r=30, t=50, b=50),
    height=525,
    xaxis=dict(range=[0, width]),
    yaxis=dict(range=[height, 0]),
    legend_title=dict(
        text='Legend:',
        font=dict(size=11, family='Arial, sans-serif', color=title_color)
    ),
    legend=dict(
        orientation='v',
        x=1,
        xanchor='left',
        y=1,
        yanchor='top',
        font=dict(size=11, family='Arial, sans-serif', color=title_color)
    ),
    plot_bgcolor='rgba(0, 0, 0, 0)', # Transparent plot area
)

# Create Scarf Plot
def create_scarf_plot(df, cluster_colors):
    if df.empty:
        return go.Figure().update_layout(
            title="No data available for the scarf plot.",
            xaxis=dict(visible=False),
            yaxis=dict(visible=False)
        )

    # Berechnung der kumulativen Fixationsdauer
    # Berechnung der kumulativen Fixationsdauer
    df['CumulativeTime'] = df.groupby(['user', 'AOI Cluster'])['Timestamp'].cumsum()

    # Explizite Sortierung der Nutzer und Cluster
    cluster_order = df['AOI Cluster'].unique() # Optional: Sortierung nach Namen/ID
    user_order = sorted(df['user'].unique()) # Optional: Sortierung nach Namen/ID

    # Scarf-Plot erstellen
    fig = px.bar(
        df,
        x='FixationDuration',
        y='user',
        color='AOI Cluster',
        color_discrete_map=cluster_colors,
        barmode='stack',
        orientation='h',
        category_orders={
            'user': user_order, # Explizite Sortierung der Nutzer
            'AOI Cluster': cluster_order # Explizite Sortierung der Cluster
        }
    )

    return fig

```

```

# Create the Scarf Plot
fig_scarf = create_scarf_plot(filtered_df, cluster_colors)

# Update the layout for additional customization
fig_scarf.update_layout(
    title={
        'text': f'<b>Scarf Plot: AOI Gazed Over Time in {selected_city}</b>',
        'font': {'size': 12, 'family': 'Arial, sans-serif', 'color': title_color}
    },
    xaxis=dict(visible=True, title=None),
    yaxis=dict(visible=True, title=None),
    legend_title=dict(
        text='AOI Cluster', # Title of the legend
        font=dict(size=11, family='Arial, sans-serif', color=title_color)
    ),
    margin=dict(l=50, r=30, t=50, b=50),
    height=525,
    legend=dict(
        orientation='h', # Horizontal legend
        font=dict(size=11, family='Arial, sans-serif', color=title_color)
    ),
    plot_bgcolor='rgba(0, 0, 0, 0)',
    # coloraxis_showscale=False
    # Transparent background
)

return fig_cluster, fig_scarf

```

Codeausschnitt 4: Opacity Map Grey ('app.py')

```

@app.callback(
    Output('opacity_map_grey', 'figure'),
    [
        Input('city_dropdown', 'value'),
        Input('dropdown_user_grey', 'value'),
        Input('range_slider_grey', 'value'),
        Input('current_theme', 'data')
    ]
)
def update_opacity_map(selected_city, selected_users, range_slider_value, current_theme):
    if selected_city:
        # Filter and sort data based on the selected filters (city and user):
        filtered_df = df[(df['CityMap'] == selected_city) & (df['description'] == 'grey')]

    if selected_users:
        if isinstance(selected_users, str):
            selected_users = [selected_users]
        filtered_df = filtered_df[filtered_df['user'].isin(selected_users)]

    min_duration, max_duration = range_slider_value
    filtered_df = filtered_df[
        (filtered_df['FixationDuration_aggregated'] >= min_duration) &
        (filtered_df['FixationDuration_aggregated'] <= max_duration)]

    # Extract Image Information and normalize data
    image_path_grey, width, height = get_image_path_grey(selected_city)
    if image_path_grey and width and height:
        # Normalize fixation points
        if selected_city == 'Antwerpen_S1':
            filtered_df['NormalizedPointX'] = (
                filtered_df['MappedFixationPointX'] / 1651.00 * width)
            filtered_df['NormalizedPointY'] = (
                filtered_df['MappedFixationPointY'] / 1200.00 * height)
        else:
            filtered_df['NormalizedPointX'] = filtered_df['MappedFixationPointX']
            filtered_df['NormalizedPointY'] = filtered_df['MappedFixationPointY']

    # Filter for fixation points within map only
    filtered_df = filtered_df[

```

```

width) &
    (filtered_df['NormalizedPointX'] >= 0) & (filtered_df['NormalizedPointX'] <=
height)]

    (filtered_df['NormalizedPointY'] >= 0) & (filtered_df['NormalizedPointY'] <=

# Set title color based on theme
title_color = 'black' if current_theme == 'light' else 'white'

# Initialize opacity values for the map
fig = px.density_heatmap(
    filtered_df,
    x='NormalizedPointX',
    y='NormalizedPointY',
    nbinsx=50,
    nbinsy=50,
    color_continuous_scale=[
        [0.0, "rgba(0, 0, 0, 1)"], # Black (opaque)
        [0.5, "rgba(255, 255, 255, 0)"], # Gray (semi-transparent)
        [1.0, "rgba(255, 255, 255, 0)"] # White (fully transparent)
    ],
)
fig.update_coloraxes(showscale=False)

# Update heatmap traces
fig.update_traces(
    showscale=False, # Hide the scale bar
    opacity=0.8
)

# Update axes
fig.update_xaxes(
    range=[0, width],
    autorange=True,
    showgrid=False,
    showticklabels=True,
    tickfont=dict(color=title_color, size=9, family='Arial, sans-serif'))

fig.update_yaxes(
    range=[height, 0],
    autorange=True,
    showgrid=False,
    showticklabels=True,
    tickfont=dict(color=title_color, size=9, family='Arial, sans-serif'))

# Add background image
fig.add_layout_image(
    dict(
        source=image_path_grey,
        x=0,
        sizex=width,
        y=0,
        sizey=height,
        xref="x",
        yref="y",
        sizing="stretch",
        opacity=0.8,
        layer="below"
    )
)

# Update layout
fig.update_layout(
    plot_bgcolor='rgba(255, 255, 255, 0.8)',
    paper_bgcolor='rgba(0, 0, 0, 0)',
    xaxis_title=None,
    yaxis_title=None,
    title={
        'text': f'<b>Opacity Map Observations in {selected_city}</b>',
        'font': {
            'size': 12,
            'family': 'Arial, sans-serif',

```

```

        'color': title_color}
    },
    margin=dict(l=0, r=5, t=40, b=5),
    showlegend=False,
    height=425)

    return fig

else:
    title_color = 'black' if current_theme == 'light' else 'white'
    cities = [
        {"name": "Antwerpen", "lat": 51.2194, "lon": 4.4025},
        {"name": "Berlin", "lat": 52.5200, "lon": 13.4050},
        {"name": "Bordeaux", "lat": 44.8378, "lon": -0.5792},
        {"name": "Köln", "lat": 50.9375, "lon": 6.9603},
        {"name": "Frankfurt", "lat": 50.1109, "lon": 8.6821},
        {"name": "Hamburg", "lat": 53.5511, "lon": 9.9937},
        {"name": "Moskau", "lat": 55.7558, "lon": 37.6173},
        {"name": "Riga", "lat": 56.9496, "lon": 24.1052},
        {"name": "Tokyo", "lat": 35.6895, "lon": 139.6917},
        {"name": "Barcelona", "lat": 41.3851, "lon": 2.1734},
        {"name": "Bologna", "lat": 44.4949, "lon": 11.3426},
        {"name": "Brüssel", "lat": 50.8503, "lon": 4.3517},
        {"name": "Budapest", "lat": 47.4979, "lon": 19.0402},
        {"name": "Düsseldorf", "lat": 51.2277, "lon": 6.7735},
        {"name": "Göteborg", "lat": 57.7089, "lon": 11.9746},
        {"name": "Hong-Kong", "lat": 22.3193, "lon": 114.1694},
        {"name": "Krakau", "lat": 50.0647, "lon": 19.9450},
        {"name": "Ljubljana", "lat": 46.0569, "lon": 14.5058},
        {"name": "New-York", "lat": 40.7128, "lon": -74.0060},
        {"name": "Paris", "lat": 48.8566, "lon": 2.3522},
        {"name": "Pisa", "lat": 43.7228, "lon": 10.4017},
        {"name": "Venedig", "lat": 45.4408, "lon": 12.3155},
        {"name": "Warschau", "lat": 52.2297, "lon": 21.0122},
        {"name": "Zürich", "lat": 47.3769, "lon": 8.5417}
    ]

    lats = [city["lat"] for city in cities]
    lons = [city["lon"] for city in cities]
    names = [city["name"] for city in cities]

    fig = go.Figure()

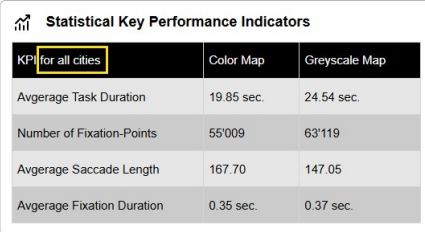
    fig.add_trace(go.Scattergeo(
        locationmode='ISO-3',
        lon=lons,
        lat=lats,
        text=names,
        mode='markers',
        marker=dict(
            size=6,
            symbol='circle',
            color='blue'
        ),
        textposition='top right',
        hoverinfo='text'
    ))

    fig.update_layout(
        title=dict(
            text='<br><br><b>Available City Maps</b><br>'
            '(zoom out to see cities outside Europe)',
            font=dict(size=12, family='Arial, sans-serif', color=title_color),
        ),
        plot_bgcolor='rgba(0, 0, 0, 0)',
        paper_bgcolor='rgba(0, 0, 0, 0)',
        geo=dict(
            projection_type='natural earth',
            showland=True,
            landcolor='lightgray',
            coastlinecolor='darkgray',

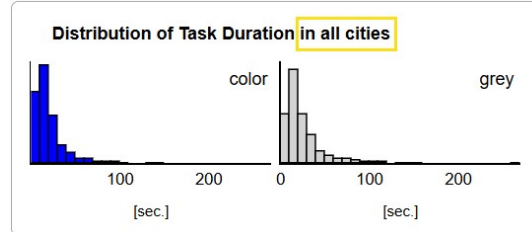
```

```
        showcoastlines=True,
        showcountries=True,
        countrycolor='darkgray',
        lonaxis=dict(range=[-10, 40]), # Longitude range for Europe
        lataxis=dict(range=[35, 65]) # Latitude range for Europe
    ),
    margin=dict(l=5, r=5, t=100, b=5),
    height=424
)
return fig
```

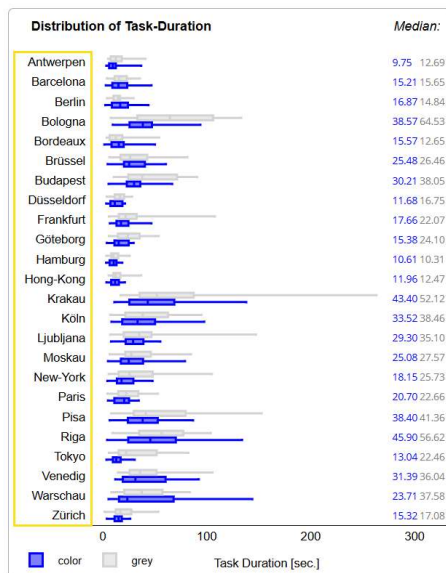
Abbildung 12: Visualisierungen der globalen Analyse



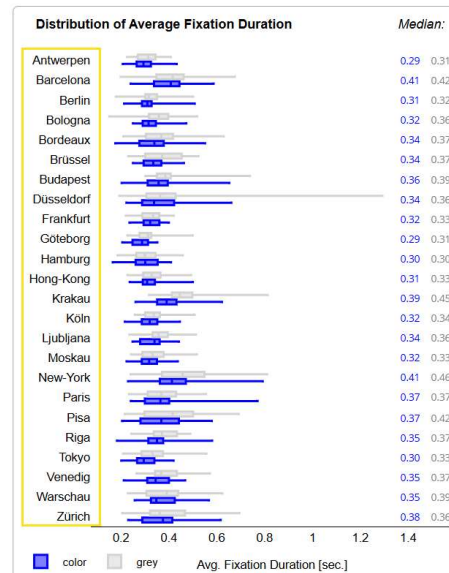
app.py – Sektion 4.1



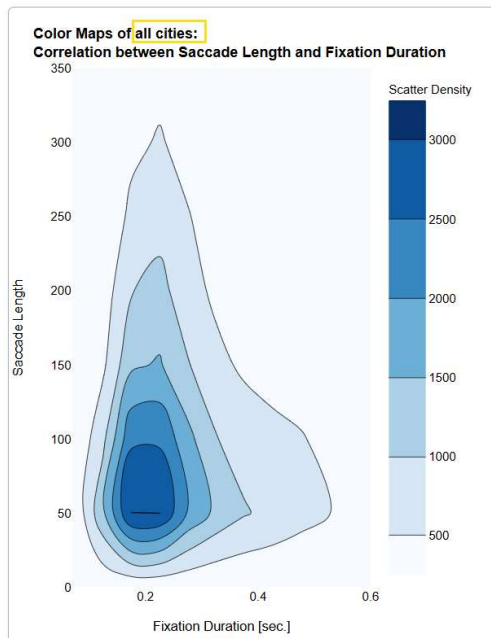
app.py – Sektion 4.8



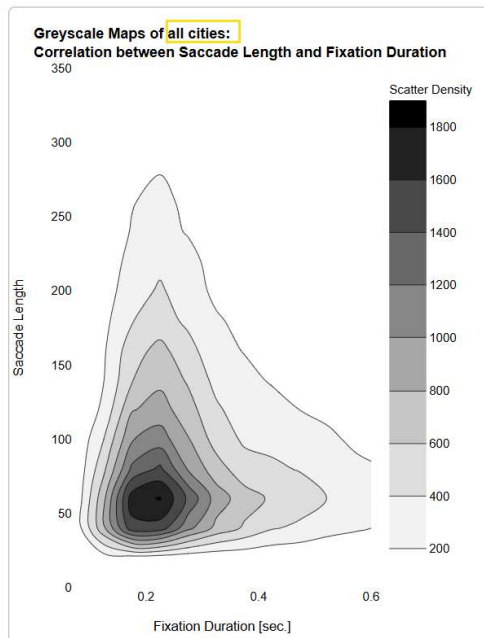
app.py – Sektion 4.6



app.py – Sektion 4.7



app.py – Sektion 4.9



app.py – Sektion 4.10

Abbildung 13: Visualisierungen der Detail-Analyse (Beispiel Tokyo)

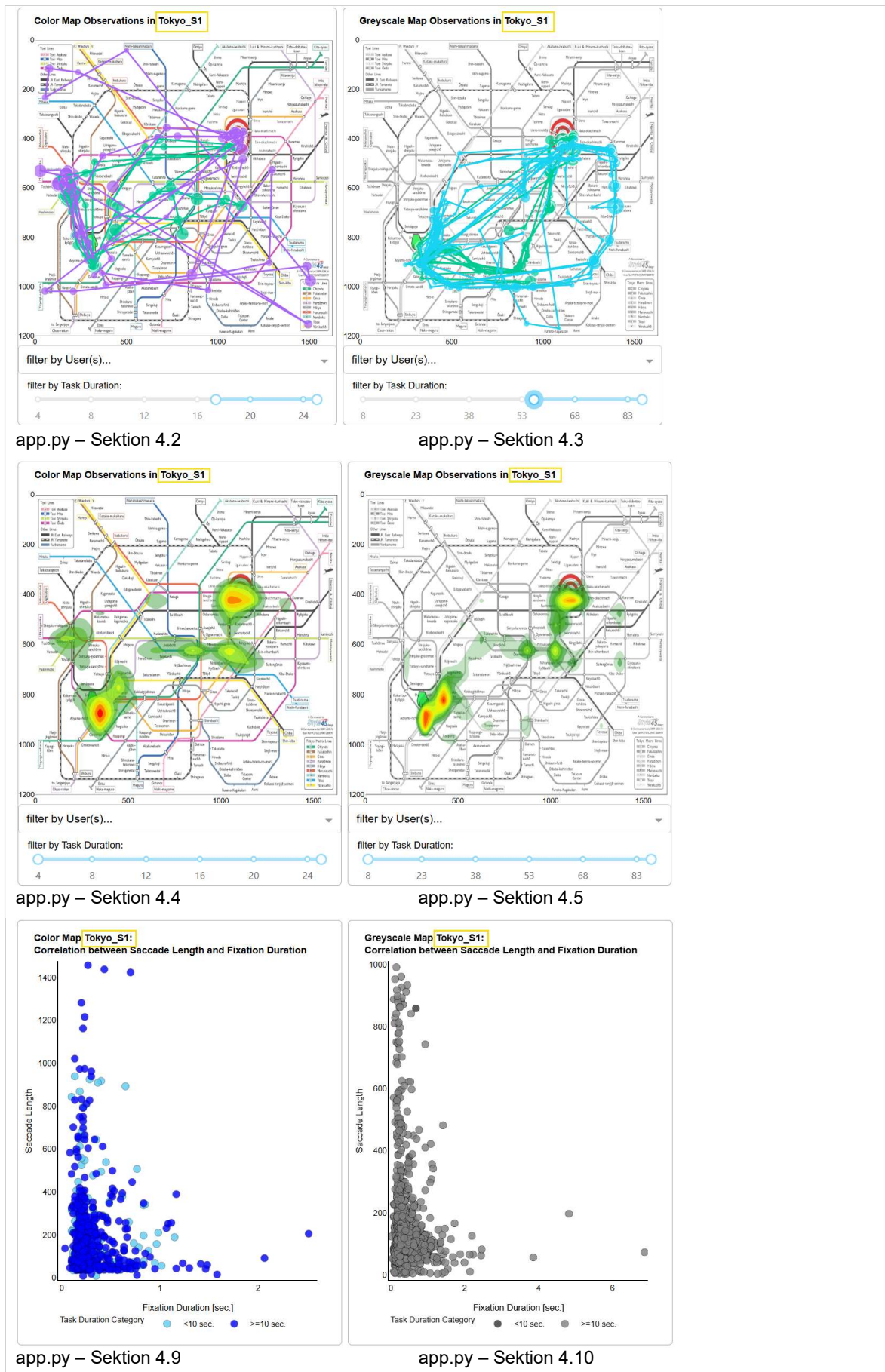
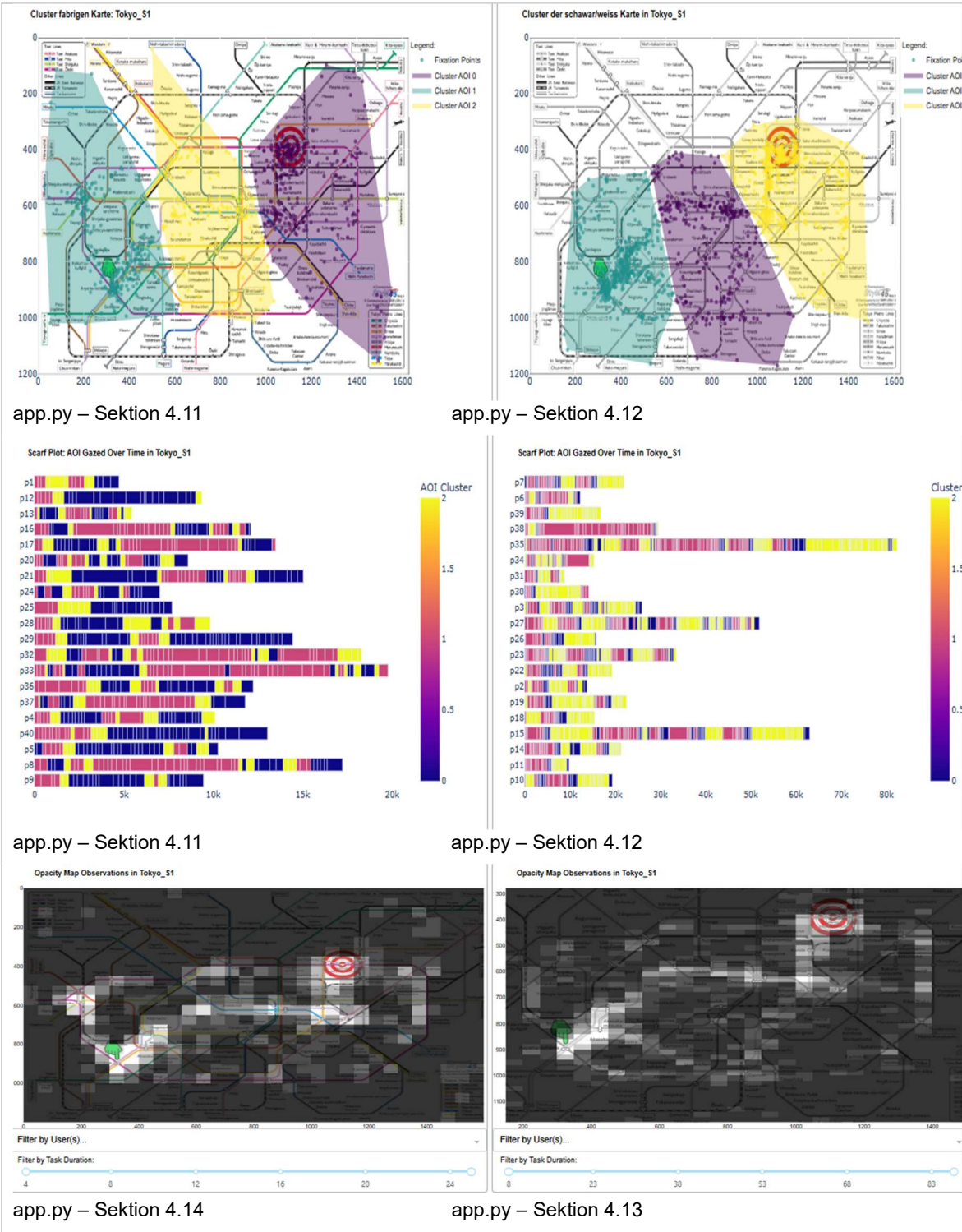


Abbildung 14 Neue Visualisierungen der Detail-Analyse (Beispiel Tokio)



Selbstständigkeitserklärung

Wir erklären hiermit, dass wir diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und erlaubten Hilfsmittel benutzt haben, einschliesslich der Verwendung von KI-Systemen. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen worden sind, haben wir als solche gekennzeichnet. Wir sind den Vorgaben des Leitfadens wissenschaftliches Arbeiten gefolgt. Uns ist bekannt, dass andernfalls die Hochschulleitung zum Entzug der aufgrund unserer Arbeit verliehenen Qualifikation oder des für unsere Arbeit verliehenen Titels berechtigt ist.

Winterthur, 20. Januar 2025



Sharon Reiser

Wetzikon, 20. Januar 2025



Serge Pellegatta