



Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάλυση χωρικών δεδομένων (ιχνών) χρησιμοποιώντας μέτρα ομοιότητας Analysis of spatial data (tracks) using similarity measurements
Όνοματεπώνυμο Φοιτητή	Βασίλειος-Θάνος Αναγνωστόπουλος
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΠΛ13002
Επιβλέπων	Γιάννης Θεοδωρίδης

Πειραιάς, 2015

©2015 Αναγνωστόπουλος Βασίλειος-Θάνος

Το κείμενο αυτό έχει γραφτεί σε $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Αυτό το κείμενο διανέμεται σύμφωνα με τους όρους της άδειας Creative Commons Attribution - ShareAlike Unported 3.0.

Εν συντομία: Είστε ελεύθεροι να διανέμετε και να τροποποιήσετε αυτό το κείμενο εφόσον αναφέρετε τον δημιουργό του και διατηρήσετε την ίδια άδεια χρήσης.

Το παρόν έγγραφο διανέμεται με την ελπίδα ότι θα είναι χρήσιμο, αλλά χωρίς καμία εγγύηση, χωρίς ακόμη και την έμμεση εγγύηση εμπορευσιμότητας ή καταλληλότητας για κάποιο συγκεκριμένο σκοπό.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν το Πανεπιστήμιο Πειραιώς.

Τριμελής Εξεταστική Επιτροπή

Γιάννης Θεοδωρίδης
Καθηγητής Πα.Πει

Νίκος Πελέκης
Λέκτορας Πα.Πει

Βασίλειος Βεσκούκης
Επίκουρος Καθηγητής Ε.Μ.Π.

Ό,τι και να κάνεις, στο τέλος θα το μετανιώσεις.

Acknowledgements

I would like to thank Vassilios Vescoukis and Vassilios Krassanakis for all their help that was given to me during this thesis. Also, I would like to thank Yannis Theodoris for accepting this thesis.

Abstract

The comprehension of phenomena related to movement - not only of people and vehicles but also of animals and other moving objects - has always been a key issue in many areas of scientific investigation and social analysis. The spread of tracking technology, not only provide us with a considerable tracking data source that enables us to study the habits of people, but also introduces open research challenges with respect to the exploitation of such wealth of data.

In the last years there is a growing popularity over the eye tracking data. Eye tracking is the recording of the eye movements and is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. The recording and the analysis of eye movements constitute an effective method for the exploration of several aspects related to human vision, perception and visual behaviour. It provides objective and quantitative evidences towards the examination of visual attention and a way to examine processes related to visual search, visual perception and cognitive process, which occurs during the observation of a stimulus.

Tracking data have similar structure with eye tracking data. The main difference is that tracking data are defined by coordinates (in geographical space) and maybe some additional attributes that may also be present and eye tracking data are defined by gaze coordinates on a screen. The structural similarity suggests that both classes of data may be analysed using the same methods.

When working with eye tracking data, the determination of some measurement of similarity between many subjects gaze is desirable. Object similarity or object dissimilarity or object matching or shape matching is the decision of the resemblance (similarity) between two objects. The object similarity functions of tracking data can be used in eye tracking data for extracting similarities between the gazes of two subjects. However, not all methods may be relevant since the movement properties are not the same for eye movements and for road or sea traffic, human mobility, or animal migration.

In this thesis we examine how the object similarity functions can be applied to eye tracking data and a toolkit is developed for this purpose. Also a case study, that use the software is presented and the results of this case study are shown.

Περίληψη

Η κατανόηση φαινομένων που σχετίζονται με την κίνηση - όχι μόνο των ανθρώπων και των οχημάτων αλλά επίσης και τον ζώων και άλλων κινούμενων αντικειμένων - ήταν πάντα ένα επίκαιρο θέμα στην επιστημονική και κοινωνική έρευνα. Η εξάπλωση της τεχνολογίας καταγραφής των ιχνών, όχι μόνο μας προσφέρουν μία πληθώρα πηγών από ίχνη αλλά μας επιτρέπουν και την μελέτη των συνηθειών του ανθρώπου, αλλά και εισάγουν ερευνητικές προκλήσεις που αφορούν την εκμετάλλευση μιας τόσο πλούσιας πηγής δεδομένων.

Τα τελευταία χρόνια υπάρχει ένα αυξανόμενο ενδιαφέρον όσο αφορά τα δεδομένα από την καταγραφή οφθαλμικών κινήσεων. Η καταγραφή των οφθαλμικών κινήσεων είναι η διαδικασία μέτρησης είτε του σημείου του βλέμματος (όπου κάποιος κοιτάει) είτε της κίνηση του οφθαλμού σε σύγκριση με το κεφάλι. Η καταγραφή και η ανάλυση των οφθαλμικών κινήσεων αποτελεί ένα αποτελεσματικό τρόπο για την εξερεύνηση διαφόρων πτυχών που σχετίζονται με την ανθρώπινη όραση, αντίληψη και συμπεριφορά. Παρέχει ένα αντικειμενικό και ποσοτικό τρόπο εξέτασης των διαδικασιών που σχετίζονται με την οπτική προσοχή, την οπτική αντίληψη και τις γνωστικές διαδικασίες, οι οποίες λαμβάνουν κατά την διάρκεια της παρατήρησης ενός ερεθίσματος.

Τα δεδομένα ιχνών έχουν παρεμφερή μορφή με τα δεδομένα από την καταγραφή οφθαλμικών κινήσεων με την κύρια διαφορά να είναι ότι τα δεδομένα ιχνών ορίζονται από συντεταγμένες (στο γεωγραφικό χώρο) και ίσως από μερικά επιπρόσθετα χαρακτηριστικά ενώ τα οφθαλμικά δεδομένα ορίζονται από συντεταγμένες βλέμματος πάνω σε μία οθόνη. Αυτή η δομική ομοιότητα υποδηλώνει ότι και οι δύο κατηγορίες δεδομένων μπορούν να αναλυθούν χρησιμοποιώντας τις ίδιες μεθόδους.

Πολλές φορές, ο προσδιορισμός της ομοιότητας μεταξύ των οφθαλμικών κινήσεων από πολλούς παρατηρητές είναι επιθυμητή. Η ομοιότητα των αντικειμένων ή η ομοιότητα σχημάτων είναι η απόφαση της ομοιότητας μεταξύ δύο αντικειμένων. Οι μέθοδοι που χρησιμοποιούνται στα ίχνη για τον προσδιορισμό της ομοιότητας τους μπορούν να χρησιμοποιηθούν και στις οφθαλμικές κινήσεις. Ωστόσο, δεν είναι όλες οι μέθοδοι κατάλληλες, μίας και οι ιδιότητες της κίνησης δεν είναι ίδιες για τις οφθαλμικές κινήσεις και τις κινήσεις που παρατηρούνται στους δρόμους ή στην θάλασσα, στην ανθρώπινη κινητικότητα και στις μεταναστεύσεις των ζώων.

Σε αυτή την διπλωματική, εξετάζουμε πως οι μέθοδοι ομοιότητας των ιχνών μπορούν να χρησιμοποιηθούν στις οφθαλμικές κινήσεις και γι' αυτό το σκοπό αναπτύχθηκε ένα πρόγραμμα. Ακόμα, τα αποτελέσματα μίας μελέτης περίπτωσης παρουσιάζονται στην οποία χρησιμοποιείται το πρόγραμμα και οι μέθοδοι ομοιότητας των ιχνών.

Preface

This thesis was prepared at the Department of Informatics of the University of Piraeus, in partial fulfillment of the requirements for acquiring the MSc degree in Informatics. This study has been conducted under the supervision of Professor Yannis Theodoridis and Vasilios Vescoukis.

This thesis presents a new approach of visualizing and analyzing eye tracking data using the APL and the TraceBundle algorithm. It comprises of 5 chapters and their structures is as follows:

Chapter 1: is the introduction of the thesis and defines the problem that this thesis tries to solve.

Chapter 2: describes what is object similarity and proceeds with describing the mathematical background of distance functions.

Chapter 3: proceeds with an analysis of the eye movements and eye tracking data.

Chapter 4: describes the toolkit that was develop during this thesis for the comparison of eye tracking data.

Chapter 5: which is the final chapter of this thesis, presents the conclusions that were exported from this thesis.

From this thesis the following paper was submitted to Spatial Cognition & Computation Journal:

V. Vescoukis, S. Karagiorgou, V. Anagnostopoulos, and V. Krassanakis. "Introducing APLs - average polylines for visualisation and analysis of eye-tracking data". Manuscript submitted for publication. 2015

Contents

Abstract	v
Περίληψη	vii
Preface	ix
List of Figures	xiv
List of Tables	xv
List of Algorithms	xvii
1 Introduction	1
1.1 Problem definition	2
1.2 Tracks	2
1.2.1 Geodetic coordinates	2
1.2.2 Trajectories	4
1.3 Problems arising from tracks	4
1.3.1 Inference of Road Networks from Tracking Data	5
1.3.2 Movement patterns in tracking data	5
1.3.3 Determining point of interests	6
1.3.4 Wildlife tracking data	6
1.3.5 Eye tracking data	7
2 Object similarity	9
2.1 Introduction	10
2.2 L_P norm	11
2.3 Discrete metric	11
2.4 Hausdorff distance	11
2.4.1 Partial Hausdorff distance	12
2.4.2 Modified Hausdorff distance	12
2.5 Bottleneck distance	13
2.6 Fréchet distance	13
2.7 Area of symmetric difference	14
2.8 Similarity of trajectories	15
2.9 Euclidean distance measure	15
2.10 Dynamic Time Warping - DTW	16

2.11	Longest Common SubSequence - LCSS	17
2.12	Summarizing	18
3	Eye tracking	19
3.1	Introduction	20
3.2	Eye movements	20
3.3	Visualization	22
3.3.1	Gaze plot	22
3.3.2	Heatmaps	22
3.3.3	Average Polyline - APL	24
3.4	Structure and properties of eye tracking data	24
3.5	Calculation of Average Polyline (APL)	26
4	Toolkit	29
4.1	Introduction	30
4.2	Concepts and work flow	30
4.3	Software	30
4.4	Requirements outline	33
4.4.1	Non-functional requirements	33
4.4.2	Functional requirements	34
4.5	Case study	35
4.5.1	Eye Tracker	37
4.5.2	Stimuli	37
4.6	Analysis of samples	38
4.6.1	Type 1 question	39
4.6.2	Type 2 question	44
5	Conclusions	47
5.1	Introduction	47
5.2	Conclusions	47
5.3	Proposals	47
	Bibliography	48
	Appendix A Peristera stimulus data	55
	Appendix B Thapa stimulus data	59
	Appendix C Submitted paper	61

List of Figures

1.1	Example path in a city	3
1.2	Tracking data in an example path	3
1.3	Geodetic coordinates [20].	4
1.4	Berlin road network [28]	5
1.5	Moving patterns [23]	6
1.6	The temporal profiles of the traffic jam occurrences [3]	6
1.7	Plot of hooded seal track data [26]	7
1.8	An Example of Eye Gaze Trace Overlay [61]	8
2.1	The direct Hausdorff distance between the blue and green point sets. The yellow arrows indicate the shortest distances from each point on the blue set to a point on the green set. The red arrow indicates the Hausdorff distance [6]	12
2.2	Hausdorff and Bottleneck distances [56]	13
2.3	Limitations of the Hausdorff distance [56]	14
2.4	Interpolation of two trajectories using the Euclidean distance norm (L_2)[42]	16
2.5	The mapping of a trajectory using either the Euclidean or the DTW function [42]	17
2.6	Point correspondence of two trajectories [14]	17
3.1	Histogram of all publications, relevant for eye tracking data visualization techniques. The number of published articles, conference papers, and books has strongly increased during the last decade [10]	20
3.2	Gaze plot visualizing the fixation and saccades movements occurring during the recording of eye movements ([52] adopted from Krassanakis [29])	23
3.3	Heatmap with different levels of intensity ([16] adopted from Krassanakis [29])	23
3.4	Different type of heatmaps (from left to right: Count, Absolute Duration, Relative Duration) [38] adopted from Krassanakis [29]	24
3.5	APL visualization. In the figure the raw and fixations points are also shown.	25
3.6	Calculation of the APL from 3 users	28
4.1	Structure of table T to used in analyzing APLS and individual's behavior.	33
4.2	The APL analysis toolkit	35
4.3	The MATLAB® toolkit	36
4.4	The Eye Tracking system, used in the NTUA experiments [8]	37
4.5	The geometry of the subject and the screen [8]	38
4.6	Stimuli	39
4.7	Heatmap of the stimuli [7]	40
4.8	APL of the stimuli	41

4.9	Subjects APL with reference line	42
4.10	Normalized distribution of Hausdorff distance across experiments	43
4.11	Normalized Hausdorff distance distribution across groups for the "Thapa" experiment	44
4.12	Normalized distribution of L_i/L_{ref} across groups for the "Thapa" experiment.	45

List of Tables

2.1	Comparison among the distance functions	18
3.1	Taxnomy of fixations identifications algorithms [44]	21
3.2	Parameters of APL algorithm.	28
4.1	An overview of the freely available tools for eyemovement analysis (modified from Krassanakis, Filippakopoulou, and Nakos [31]).	31
4.2	The normalized distribution of the two experiments	43
4.3	The normalized distribution of the two experiments	45
A.1	The data of Peristera stimulus	57
B.1	The data of Thapa stimulus	60

List of Algorithms

2.1	The intuitive algorithm for calculating the similarity between two trajectories [32]	15
3.1	Connecting hubs [57]	27
3.2	APL extraction [57]	27
4.1	APL analysis workflow [57]	30

CHAPTER 1

Introduction

1.1 Problem definition	2
1.2 Tracks	2
1.2.1 Geodetic coordinates	2
1.2.2 Trajectories	4
1.3 Problems arising from tracks	4
1.3.1 Inference of Road Networks from Tracking Data	5
1.3.2 Movement patterns in tracking data	5
1.3.3 Determining point of interests	6
1.3.4 Wildlife tracking data	6
1.3.5 Eye tracking data	7

Figures

1.1 Example path in a city	3
1.2 Tracking data in an example path	3
1.3 Geodetic coordinates [20].	4
1.4 Berlin road network [28]	5
1.5 Moving patterns [23]	6
1.6 The temporal profiles of the traffic jam occurrences [3]	6
1.7 Plot of hooded seal track data [26]	7
1.8 An Example of Eye Gaze Trace Overlay [61]	8

1.1 Problem definition

The comprehension of phenomena related to movement - not only of people and vehicles but also of animals and other moving objects - has always been a key issue in many areas of scientific investigation and social analysis. Advances in technology and particularly the miniaturization of electronic components, have allowed the widespread adoption and use of GPS enable devices which have opened up new opportunities for tracking the movement of various types of entities, including vehicles, humans and animals.

The spread of the tracking technology, such as the smart phones (and watches) applications, the GPS navigation applications, the social-media check-ins applications, activity trackers, e.t.c, not only provide us with a considerable tracking data source that enables us to study the habits of people, but also introduces new research challenges with respect to the exploitation of such wealth of data.

Trajectory data and similarity-based retrieval has attracted increasing interest in database and knowledge discovery communities because of its wide use in various applications. For example in figure 1.1 a path in city is shown. The question that nowadays researchers want to answer is how similar are the tracks shown in figure 1.2.

To address these new challenges this thesis deals with object similarity in tracks and presents a case study in eye tracking data.

1.2 Tracks

Tracking data is usually acquired by GPS enabled devices such as vehicles with GPS navigation, smart-phones and smart-watches. This sections proceeds by first explaining the Geodetic coordinate system which is used in GPS.

1.2.1 Geodetic coordinates

GPS tracking data is at least three dimensional, consisting of coordinates in space, a coordinate in time and occasionally, other associated informations are included, such as speed, direction, etc. Most of the modern GPS receivers provide the tracking data in geodetic coordinates followed by a timestamp of the recording. A typical example of data collected from GPS device is show below:

```

1 Time,X,Y
2 19.127,1138.5,657.87
3 19.277,837.07,602.43
4 19.444,727.82,640.31
5 .
6 .
7 .

```

The Earth's surface is approximated by an ellipsoid and locations near the surface are described in terms of geodetic coordinates (see fig. 1.3) and specifically:

longitude (ϕ): According to Wikipedia [63] longitude of a point on the Earth's surface is the angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth [63].

latitude (λ): According to Wikipedia [63] latitude of a point on the Earth's surface is the angle east or west from a reference meridian to another meridian that passes through that point [63].

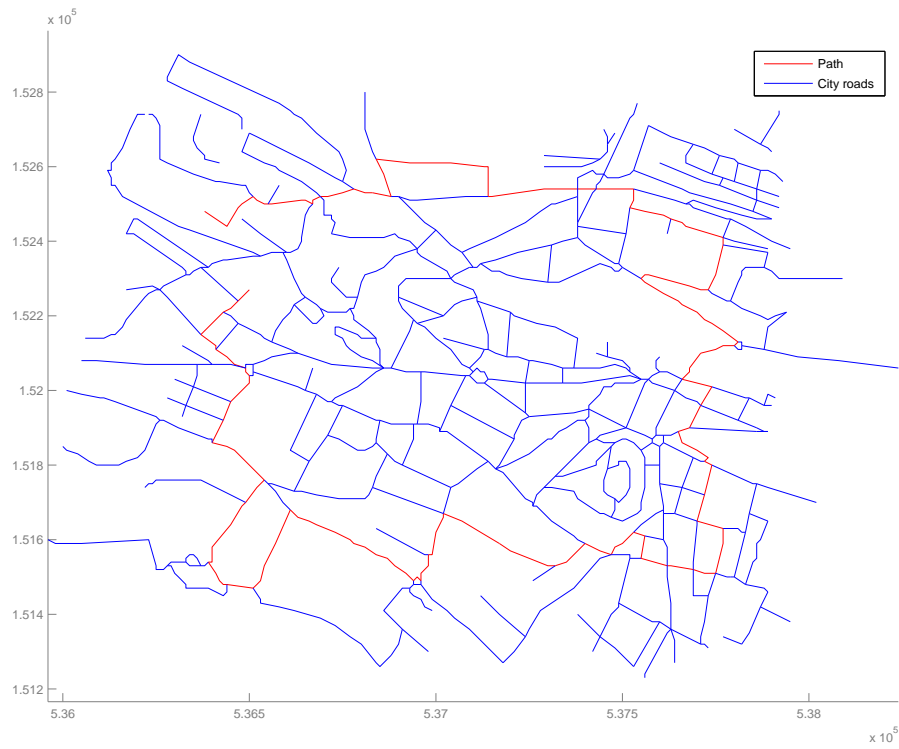


Figure 1.1: Example path in a city

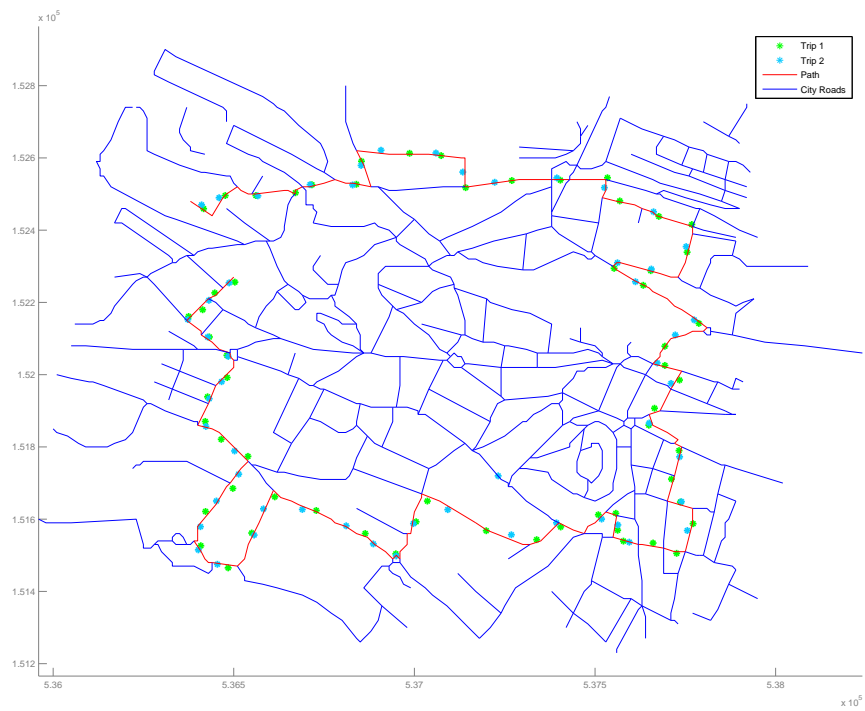


Figure 1.2: Tracking data in an example path

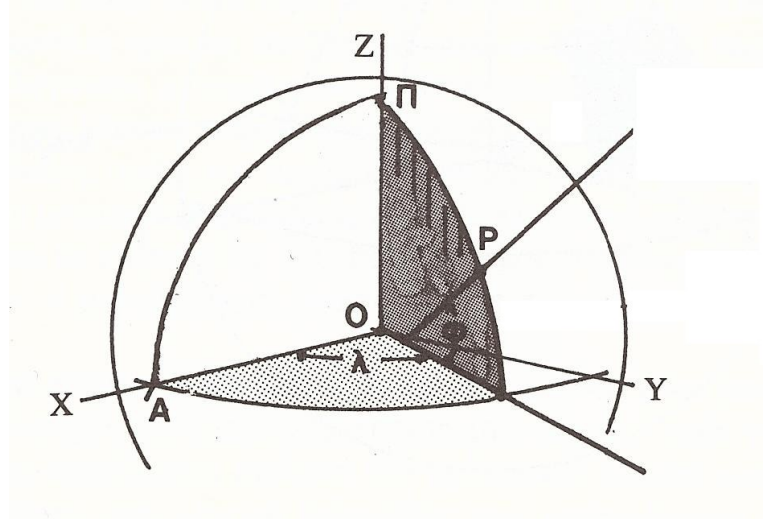


Figure 1.3: Geodetic coordinates [20].

height (h): of a point is the distance between the point and the earth's ellipsoid

The ellipsoid is a circular coordinate system and the main disadvantage of the coordinate system is that it is not rectangular, hence, the calculation of the distance between any two points identified by latitude and longitude is not straightforward. The advantage of this coordinate system is that it is continuous for the entire world.

1.2.2 Trajectories

A trajectory is the continuous path that a moving object follows through space as a function of time. However, due to the limitation of location positioning devices (e.g. sensors, GPS devices), a trajectory in the real world is a sequence of positions observed at discrete time instances. If t_0 is the time moment when the object starts moving and t_{end} is the moment when it ended its movement, for any moment $t_i, t_0 \leq t_i \leq t_{end}$, there is a position in space that was occupied by the entity at this moment, although the position may not be known. The trajectory of a continuous curve can be represented by a finite point sample, given as a finite sequence $T = \{p_0, \dots, p_n\}$ with $p_i = (x_i, y_i, t_i)$ and $x_i, y_i \in \mathbb{R}, t_i \in \mathbb{R}^+$ for $i = 0, 1, \dots, n$ and $t_0 < t_1 < \dots < t_m$. Each p_i minimally consists of a time stamp as well as a position measurement, but depending on the application may contain additional data such as speed or acceleration. [70, 28, 59, 5].

Trajectories are subject to different kinds of noise and the measurements of the position samples are only accurate within certain bounds (measurement error). The movement transition between position samples can be modeled with varying accuracies depending on the application (sampling error) [28].

1.3 Problems arising from tracks

In the following sections, we discuss related work in the area of moving objects and spatiotemporal data, outlining the problems arising from tracks.

1.3.1 Inference of Road Networks from Tracking Data

Road networks and generally transportation networks are of fundamental importance in a number of applications, in that they represent the principal data set in these applications, including GIS, location-based services and transportation systems.

In the past, the production of street maps required expensive field surveying and labor-intensive postprocessing. Although, the vast amount of existing tracking data produced over the last years, has given the possibility of inferring the road network from tracking data. This problem has two broad categories of application scenarios. The first potential refers to cases where the map of the network exists but needs to be maintained and updated, or enhanced with additional properties and the second scenario refers to cases where the entities move along specific trails, which, however, are not already mapped.

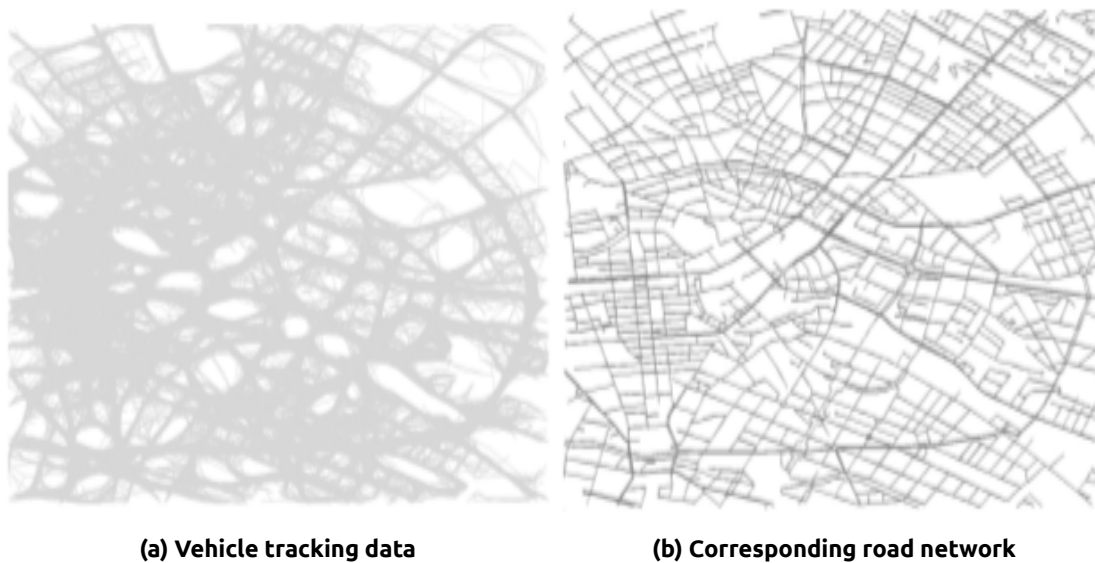


Figure 1.4: Berlin road network [28]

According to Karagiorgou [28] besides deriving road networks, the proposed approach can be used to identify implicit movement patterns in any kinds of spatiotemporal tracking data, e.g., animal migration, historic trade routes, etc.

1.3.2 Movement patterns in tracking data

A pattern is a discernible regularity in the world. Movement patterns are generally conceptualized as salient movement events or episodes in the geospatial representation of a certain number of moving entities. Pattern starts and ends at certain times (temporal footprint), and it might be restricted to a subset of space (spatial footprint) [66, 23].

For example in the case of moving animals, movement patterns can be viewed as the spatio-temporal expression of behaviours such as seasonal migration or in a transportation context, a movement pattern could be a traffic jam (see fig. 1.5).

The massive volumes of movement data currently available offer new insights in many dynamic processes of high socio-economic relevance and help us understand where, when and ultimately why the objects move the way they do [22]. For example, according to Gudmundsson, Laube, and Wolle [22], grazing sheep, may perform a certain movement pattern only when they are on a certain vegetation type and sea gulls may show certain flight patterns only when they are close to a salient landscape feature such as a river or a highway.

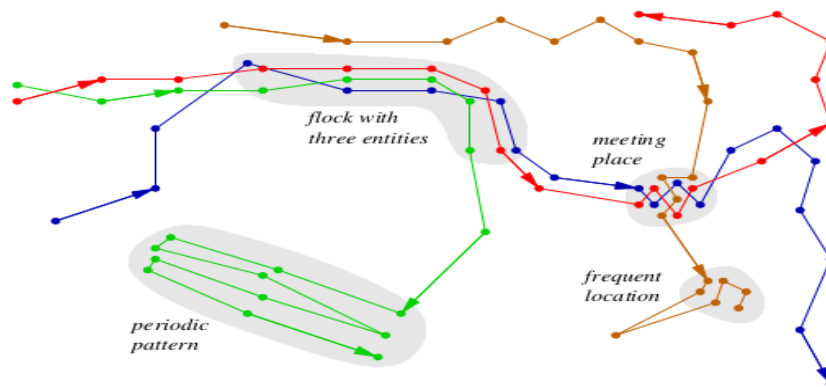


Figure 1.5: Moving patterns [23]



Figure 1.6: The temporal profiles of the traffic jam occurrences [3]

1.3.3 Determining point of interests

According to Wikipedia [67] a point of interest, or POI, is a specific point location that someone may find useful or interesting. Many problems need to determine places in which movements patterns of certain type occur repeatedly and then use these places in further analysis. One of the possible criteria for recognizing significant locations is the frequency of certain movement-related events occurring in these locations. Another possible criterion for assessing the significance of a location is the amount of time spent in it by visitors [3, 2].

For example, using tracks of multiple cars in a city, a traffic analyst can extract the places where traffic jams occurs (see fig. 1.6). From trails of migratory birds, an ornithologist may wish to extract places where the birds stop for resting and feeding. Andrienko et al. [3] has developed a generic procedure for analyzing mobility data, where relevant places are determined in order to study place-related patterns of events and movements.

1.3.4 Wildlife tracking data

According to Wikipedia [64] wildlife tracking is a process where researchers can remotely observe relative fine-scale movements or migratory patterns in a free-ranging wild animal using the GPS and optional environmental sensors or automated data-retrieval technologies such as GPRS and a range of analytical software tools.

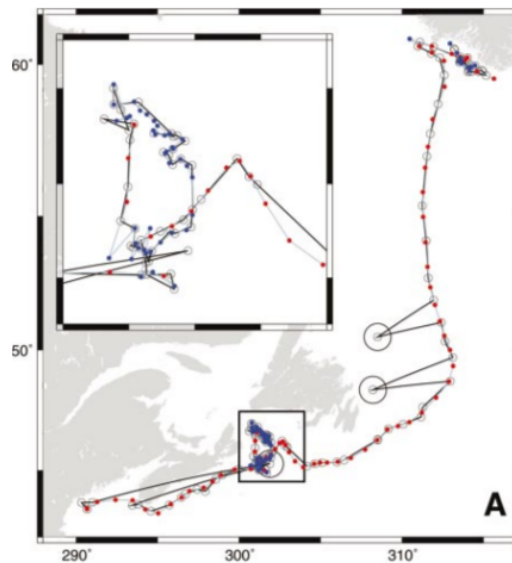


Figure 1.7: Plot of hooded seal track data [26]

In recent years, new wildlife tracking and telemetry technologies have become available, leading to substantial growth in the volume of wildlife tracking data. Large data sets generated by wildlife tracking equipment pose a number of challenges such as: *a)* coping with huge amount of data, *b)* automatic data acquisition, *c)* long-term storage, *d)* efficient management of spatial and temporal information.

More information about wildlife data can be found on [41]

1.3.5 Eye tracking data

In the last years there is a growing popularity over the eye tracking data. Eye tracking is the recording of the eye movements and is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. It provides objective and quantitative evidences towards the examination of visual attention and a way to examine processes related to visual search, visual perception and cognitive process, which occurs during the observation of a stimulus (for more details see chapter 3).

Eye tracking data consist of vectors about the positions and times of gaze data (see fig. 1.8). Tracking data have similar structure with eye tracking data with the main difference being that tracking data are defined by coordinates (in geographical space) and maybe some additional attributes that may also be present and eye tracking data is defined by gaze coordinates on a screen. The structural similarity suggests that both classes of data may be analyzed using the same methods.

A great deal of research [36, 35, 29] has gone into studies trying to analyzed eye tracking data and draw conclusions on several aspects related to visual behavior. Eye tracking has been used in psychology, in cognitive linguistics, marketing, as an input device for human computer interaction, and in product design.

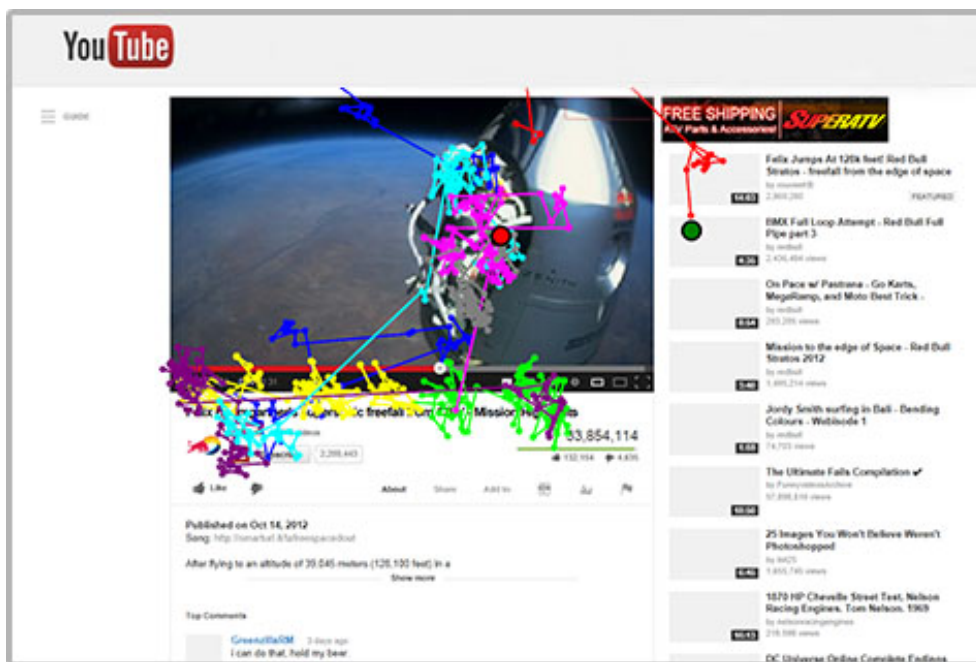


Figure 1.8: An Example of Eye Gaze Trace Overlay [61]

CHAPTER 2

Object similarity

2.1	Introduction	10
2.2	L_P norm	11
2.3	Discrete metric	11
2.4	Hausdorff distance	11
2.4.1	Partial Hausdorff distance	12
2.4.2	Modified Hausdorff distance	12
2.5	Bottleneck distance	13
2.6	Fréchet distance	13
2.7	Area of symmetric difference	14
2.8	Similarity of trajectories	15
2.9	Euclidean distance measure	15
2.10	Dynamic Time Warping - DTW	16
2.11	Longest Common SubSequence - LCSS	17
2.12	Summarizing	18

Figures

2.1	The direct Hausdorff distance between the blue and green point sets. The yellow arrows indicate the shortest distances from each point on the blue set to a point on the green set. The red arrow indicates the Hausdorff distance [6]	12
2.2	Hausdorff and Bottleneck distances [56]	13
2.3	Limitations of the Hausdorff distance [56]	14
2.4	Interpolation of two trajectories using the Euclidean distance norm (L_2)[42]	16
2.5	The mapping of a trajectory using either the Euclidean or the DTW function [42]	17
2.6	Point correspondence of two trajectories [14]	17

Tables

2.1	Comparison among the distance functions	18
-----	---	----

2.1 Introduction

Object similarity or object dissimilarity or object matching or shape matching is the decision of the resemblance (similarity) between two objects. In general, two objects A, B are given and the resemblance to each other is extracted. This can be quantitatively expressed as a distance between the two objects [48, 47, 55].

First it is necessary to formally define the notions of *a*) object and *b*) resemblance. There are many definitions on what constitutes a shape. For example according to Alt and Guibas [1] objects are a finite set of points ("point patterns") or "shapes" given in two dimensions by polygons. According to Venkatasubramanian [56] shape is a set of properties of an object that are invariant under rigid transformation (and scaling). Due to the nature of our data (i.e. points) we incline to agree more with Alt and Guibas [1].

The notion of resemblance (similarity) between shapes is even more complicated by the fact the similarity is based on the human perception and is not easily formalized [56]. Many attempts have been made to formalize the notion of similarity and most of them are based on metrics (or distance functions or shape similarity measures) that describe the degree of resemblance (difference) between two shapes.

A distance function, on a collection of shapes S is a nonnegative real value function $d : S \times S \rightarrow \mathbb{R}$ defined on pairs of patterns indicating the degree of resemblance of these patterns [48, 47, 24, 55]. If $x, y \in S$, then x is similar to y if $d(x, y) \leq \epsilon$, where ϵ is a predefined threshold.

According to Veltkamp [54] and Veltkamp and Hagedoorn [55] some of the desirable properties that a similarity measure should have are:

Metric properties: According to Mémoli and Sapiro [40] a set M is a metric space if for every pair of points $x, y \in M$ there is a well defined function $d_M(x, y)$ whose values are non-negative real numbers, such that *a*) $d_M(x, y) = 0 \Leftrightarrow x = y$ (uniqueness), and *b*) $d_M(x, y) \leq d_M(y, z) + d_M(z, x)$ for any $x, y, z \in M$ (strong triangle inequality). We call the function $d_M : M \times M \rightarrow \mathbb{R}$ the metric or distance.

More informations can be found by Hagedoorn [24] on the axioms of the metric and pseudometric spaces on which the similarity measures are based.

Continuity properties: A similarity measure is desirable to be robust against the effects of discretization [54].

Invariance: In some cases, it is often desirable that the similarity measure is invariant under transformations. What kind of geometric transformations are allowed to match objects A and B depends on the application. The most simple kind are certainly translations. The matching problem usually becomes much more difficult if we allow rotations and translations (these transformations are called rigid motions, or Euclidean transformations). In most cases reflections can be included as well without any further difficulty [1].

Distributive: A distance function is distributive in the shape space if the distance between one pattern and another does not exceed the sum of distance between the one and two parts of the other [54].

According to Chen [14] a distance function directly affects the matching quality of the retrieved results and is application and data dependent. It needs to be carefully designed to meet the application requirements.

The problem of object similarity arises in a variety of applications, including computer graphics, cartography and pattern recognition and it has been approached in a number of ways. It appears in various forms in many different domains such as *a*) model-based object recognition *b*) satellite image registration *c*) stock market trend analysis and prediction *d*) and protein structure and functions determination

In the following sections some similarity measures are described.

2.2 L_P norm

Many similarity measures on shapes are based on the L_p distance. For two points $r = (r_1, \dots, r_d)$ and $q = (q_1, \dots, q_d)$ in \mathbb{R}^d the L_P norm is calculated from equation 2.1.

$$L_P(r, q) = \|r - q\|_p = \left(\sum_{i=1}^d |r_i - q_i|^p \right)^{\frac{1}{p}} \quad (2.1)$$

Equation (2.1) is also often called the Minkowski distance. L_1 norm ($p = 1$) is named the Manhattan distance or the city block distance and L_2 norm ($p = 2$) is the well known Euclidean distance. For p approaching ∞ , we have the maximum norm: $\max(|r_i - q_i|)$. According to Veltkamp [54] for all $p \geq 1$ the L_P norms are metrics and for $0 < p < 1$ it is not a metric, since the triangle inequality is not satisfied.

Unless otherwise specified, in what follows, A and B will refer to point sets in \mathbb{R}^d , where $|A| = k$ and $|B| = n \geq k$ and the underlying metric on points in \mathbb{R}^d is that induced by the L_2 norm.

2.3 Discrete metric

The most simple metric is the discrete metric, which is also known as exact congruence metric. The discrete metric is describe from equation (2.2) but it lacks any usefulness since it finds matches only if the objects are equal. This metric is described in depth from Veltkamp [54] and Hagedoorn [24].

$$d(A, B) = \begin{cases} 0 & \text{if } A \text{ equals } B \\ 1 & \text{otherwise} \end{cases} \quad (2.2)$$

2.4 Hausdorff distance

The Hausdorff metric is the most studied similarity measure in computational geometry. The Hausdorff distance is defined for an arbitrary non-empty and closed set A and B . The directed Hausdorff distance simply assigns to each point of one set the distance to its closest point on the other and takes the maximum over all these values [47, 1] (see equation (2.3) and figure 2.1)

$$\tilde{\delta}_H(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \quad (2.3)$$

where $d(x, y)$, $x, y \in \mathbb{R}^2$ denotes the Euclidean distance between x and y . In this thesis we use the Euclidean distance but other norms can also be used in the general case of the Hausdorff metric.

According to Veltkamp [54] the direct Hausdorff distance is not a metric since it fails the triangle inequality but the bidirectional Hausdorff distance between A and B , which is defined in equation (2.4) is a metric [47, 39].

$$\delta_H(A, B) = \max(\tilde{\delta}_H(A, B), \tilde{\delta}_H(B, A)) \quad (2.4)$$

The Hausdorff distance is robust against small deformations and performs reasonably well in practice and the main advantages are that it does not require the solution of the correspondence problem and not all points from A need to have a corresponding point in B [19, 54].

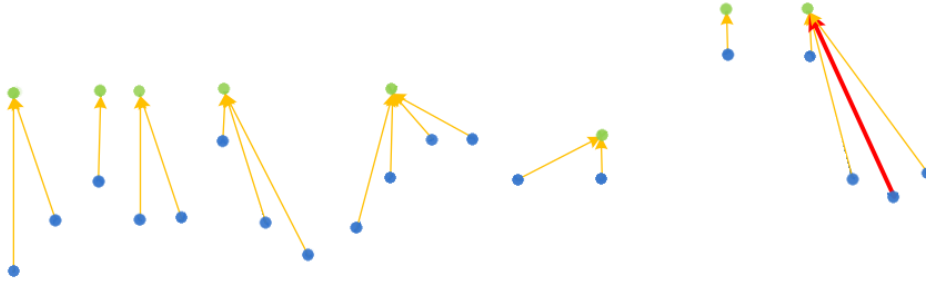


Figure 2.1: The direct Hausdorff distance between the blue and green point sets. The yellow arrows indicate the shortest distances from each point on the blue set to a point on the green set. The red arrow indicates the Hausdorff distance [6]

The main disadvantage of the Hausdorff distance is that it is sensitive to noise [47, 54, 24]. According to Dubuisson and Jain [19] the distance values are large even in the presence of small amount of noise and this is due to the fact the the Hausdorff distance value is set by the maximum distance among the two points sets. A single outlier can determine the distance value and this can be a real problem since a few outliers will perturb the distance measure greatly, even though the two objects might be very similar.

Finally, another disadvantage of the Hausdorff distance is that it only focus on the location of points and not their ordering along the the set of points. If the order of the points matters, just as in trajectories and curves, then the Hausdorff distance would not considering the ordering leading to inaccurate comparisons.

2.4.1 Partial Hausdorff distance

A similar measure that is not as sensitive as the original Hausdorff distance is the partial Hausdorff distance, which according to Veltkamp and Hagedoorn [55] is again the the maximum of the two directed partial Hausdorff distances (see equation (2.5)), where the directed distance are defined as the k -th value in increasing order of the distance from a point in A to B (see equation (2.6))

$$\delta_H^k(A, B) = \max(\tilde{\delta}_H^k(A, B), \tilde{\delta}_H^k(B, A)) \quad (2.5)$$

$$\tilde{\delta}_H^k(A, B) = k^{th}_{a \in A} \min_{b \in B} d(a, b) \quad (2.6)$$

The partial Hausdorff distance is not a metric since it fails the triangle inequality [55].

2.4.2 Modified Hausdorff distance

To solve the problem of sensitivity, Dubuisson and Jain [19] proposed a non-metric modified version of the Hausdorff distance shown in the equation (2.7). This modified version satisfies positivity and symmetry, but does not satisfy the triangle inequality.

$$\tilde{\delta}_H^{mod}(A, B) = \max(d^{mod}(A, B), d^{mod}(B, A)) \quad (2.7)$$

where $d^{mod}(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B)$, a a point in point set A and N_a the number of points

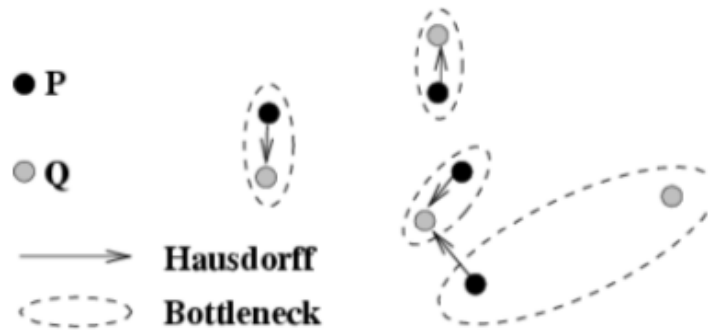


Figure 2.2: Hausdorff and Bottleneck distances [56]

in the set of points A .

Dubuisson and Jain [19] determined that, among the class of distance measures based on the Hausdorff distance, the modified version is best for matching two objects based on their edge points and it has two desirable properties: *a*) its value increases monotonically as the amount of difference between the two sets of edge points increases, and *b*) it is robust to outliers points that might result from segmentations errors.

2.5 Bottleneck distance

The Bottleneck distance is defined for two point sets A, B of the same size n (equal cardinality) as the minimum over all bijection between the sets over the maximum distance between each two points that are related in a bijection [24, 54] (see equation (2.8)).

According to Veltkamp [54] the bottleneck distance $F(A, B)$ is the minimum over all 1-1 correspondences f between A, B of the maximum distance $d(a, f(a))$

$$d_{B_\rho}(A, B) = \min_{f \in F(A, B)} \max_{a \in A} d(a, f(a)) \quad (2.8)$$

where A, B are finite subsets of a space X (usually \mathbb{R}^2) with metric ρ (usually the euclidean) and same cardinality, $F(A, B)$ the set of all bijections from A to B and $d(a, b)$ the distance between two points.

The main problem of this metric is that it needs a 1-1 correspondence and according to Venkatasubramanian [56] is more sensitive to outliers than the Hausdorff distance, since it attempts to match each point in the input. Figure 2.2 illustrates the difference between the Hausdorff distance and the bottleneck distance.

2.6 Fréchet distance

The simplicity of Hausdorff distance may lead to wrong conclusions, when used to calculate the distance between curves. According to Alt and Guibas [1] the reason for this problem is that the Hausdorff distance is only concerned with the point sets but not with the course of the curves (see figure 2.3). Notice that the Hausdorff distance between P and Q is small, because every point in P is close to some point of Q and vice versa. However the curves are not at all similar [56].

The Fréchet distance is a well-known metric to measure similarity of polygonal curves that takes into account the continuity of the shapes, the location and ordering of the points along

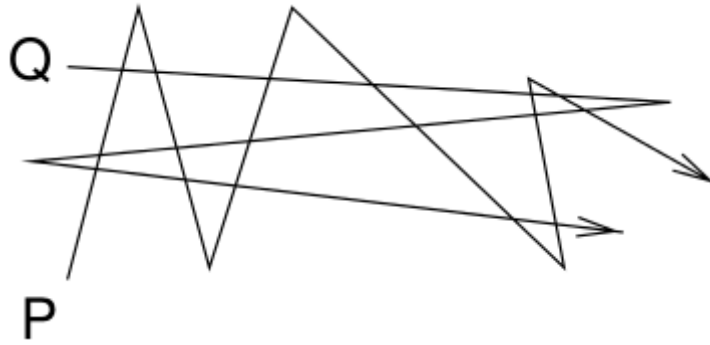


Figure 2.3: Limitations of the Hausdorff distance [56]

the curves. Wikipedia [62] describes the Fréchet distance between two curves as the minimum length of a leash required to connect a dog and its owner, constrained on two separate paths, as they walk without backtracking along their respective curves from one endpoint to the other.

The definition of the Fréchet distance given by Hagedoorn [24] is given in equation 2.9.

$$d_F(f, g) = \inf_{a, b \in \text{Hom}([0, 1])} \max_{t \in [0, 1]} \rho(f(a(t)), g(b(t))) \quad (2.9)$$

where X a space with metric ρ and let $\text{Hom}([0, 1])$ be the set of homeomorphisms from $[0, 1]$ onto itself and f, g two curves.

According to Hagedoorn [24] the Fréchet distance can be used only if the input consists of curves or closes curves.

2.7 Area of symmetric difference

For two compact sets A, B the area of symmetric difference, also called template metric, is defined as in equation (2.10). According to Veltkamp [54] and Hagedoorn [24] unlike the area of overlap, this measure is a metric.

$$s(A, B) = \text{vol}(A \Delta B) = \text{vol}((A - B) \cup (B - A)) \quad (2.10)$$

where the volume of symmetric difference is defined on the collection of Lebesgue measurable subsets of \mathbb{R}^k . For $k = 1, 2$ or 3 it coincides with the standard measure of length, area or volume.

The symmetric difference of two sets A, B is the set of all points in A that are not in B and all points in B that are not in A .

The normalized volume of symmetric difference is given by equation (2.11) [24]

$$s^*(A, B) = \frac{\text{vol}(A \Delta B)}{\text{vol}(A \cup B)} \quad (2.11)$$

The problem is that many times the set A, B are not compact and according to Hagedoorn [24] the symmetric difference is only suitable for matching regions (solid sets).

Notice that, according to De Berg et al. [18], maximizing the area of overlap of two polygon is equivalent to minimizing the area of the symmetric difference.

2.8 Similarity of trajectories

The process of calculating the similarity between two trajectories is more complicated than calculating the corresponding similarity between points. According to Laurinen, Siirtola, and Rönig [32] the reasons that makes the calculations more complicated for trajectories are:

1. that the values may be observed in equidistant or varying distant intervals
2. that the two trajectories may contain different number of measurement points
3. that the trajectories may be measured in 2 or multi-dimensional spaces.

2.9 Euclidean distance measure

Just as previous, a simple solution to calculate the similarity is to use an L_p norm to the sequences of points [42]. Euclidean distance has the advantage of being easy to compute (and specifically it has a linear computation cost [14]) and is easy to implement [42]. However, the Euclidean distance outliers have a great impact on the overall distance (due to the fact that is computed by square deviations) and it does not support local time shifting [42, 14].

Laurinen, Siirtola, and Rönig [32] describes an intuitive algorithm (see algorithm 2.1) for calculating the similarity of two trajectories with $\mathcal{O}(n^2)$ complexity but this algorithm can not compensate the problems described above.

Algorithm 2.1: The intuitive algorithm for calculating the similarity between two trajectories [32]

```

Input: trajectories  $traj_a$  and  $traj_b$  of size  $n(traj_a)$  and  $n(traj_b)$ 
Output: the distance between the trajectories,  $trajectory\_distance$ 
1  $trajectory\_distance = 0$ ;
2  $smallest\_distance = \infty$ ;
3 for  $i=1$  to  $n(traj_a)$  do                                     // scans all the points of the  $traj_a$ 
4   for  $i=1$  to  $n(traj_b)$  do                                     // scans all the points of the  $traj_b$ 
5     if  $d(traj_{a,i}, traj_{b,j}) < smallest\_distance$  then
6        $smallest\_distance = d(traj_{a,i}, traj_{b,j})$ ;
7     /* add the distance between the closest points to the distance between the
8       trajectories */
9      $trajectory\_distance += smallest\_distance$ ;
10     $smallest\_distance = \infty$ ;
11  $trajectory\_distance = (trajectory\_distance/n(traj_a))$ ;
12 return  $trajectory\_distance$ ;

```

Algorithm 2.1 resembles the logic of the modified version of the Hausdorff distance since it computes the average distance between the points of the trajectories.

To be able to calculate the measure of similarity for the trajectories, one needs to define *a*) how to find the matching pairs in the two trajectories and *b*) how to take into account the varying numbers of points in the trajectories [32]

For these reasons the L_P norms usually are combined with linear interpolation when the sampling times of the intermediate points do not match, or when the two trajectories have a different number of points (see fig. 2.4) [42].

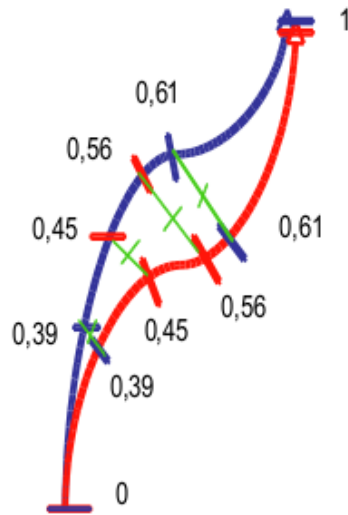


Figure 2.4: Interpolation of two trajectories using the Euclidean distance norm (L_2) [42]

2.10 Dynamic Time Warping - DTW

According to Sankararaman et al. [46], the Fréchet distance is not a good indicator of the correspondences between the trajectories due to the fact there could exist a large number of correspondences which yield the optimal Fréchet distance since the optimization criteria is the distance between the two farthest points in the coupling.

An alternative to Fréchet distance which is more appropriate to trajectories is the average Fréchet distance, aka dynamic time warping (DTW). The advantage of the DTW is that it allows a sequence to "stretch" or to "shrink" in order to better fit [42]. According to Wang et al. [59] the definition of DTW uses a recursive manner to search all possible point combinations between two trajectories for the one with minimal distance (see fig. 2.5 and eq. (2.12)).

DTW attempts to match each point with the most appropriate one and finally chooses the shortest distance. If the trajectories contain significant dissimilar portions, possibly due to actual deviations the results are not as meaningful due to the fact that DTW tries to find a correspondence for all points and thus, gives correspondences for points in the deviation for which no meaningful one exists [42, 46].

The DTW distance between two trajectories A, B of lengths m and n is defined in equation (2.12).

$$DTW(A, B) = L_P(r_n, q_m) + \min \left\{ \begin{array}{l} DTW(A, \text{Head}(B)), \\ DTW(\text{Head}(A), B) \\ DTW(\text{Head}(A), \text{Head}(B)) \end{array} \right\} \quad (2.12)$$

where $\text{Head}(A) = ((r_{1,x}, r_{1,y}) \cdots (r_{n-1,x}, r_{n-1,y}))$ (the subsequence A without the head element).

Another disadvantage of the DTW is that it does not follow triangle inequality [14] making it a non-metric function.



Figure 2.5: The mapping of a trajectory using either the Euclidean or the DTW function [42]

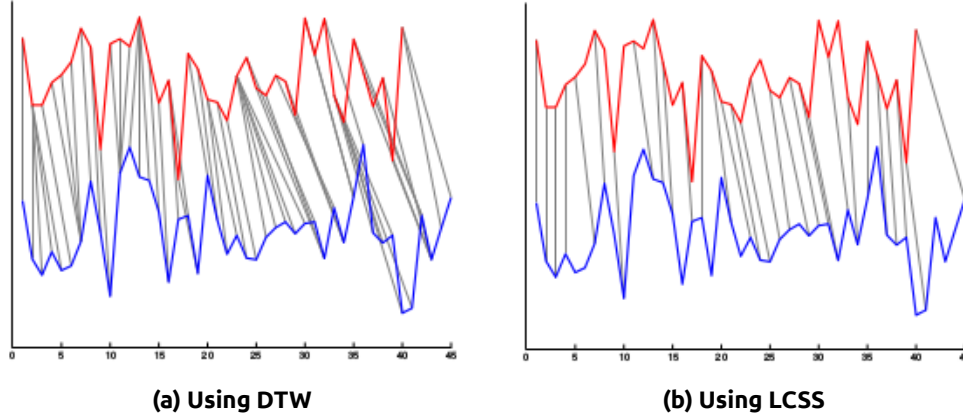


Figure 2.6: Point correspondence of two trajectories [14]

2.11 Longest Common SubSequence - LCSS

The Longest Common SubSequence (LCSS) is similar to DTW that it tries to match two trajectories by allowing their "stretching", however, without changing the sequence of elements, while allowing some elements of the sequences to be left unmatched. It can handle possible noise that may appear in data more efficient because it can disregard noisy points (see fig. 2.6) [42, 14, 59].

LCSS can be calculated by equation (2.13).

$$LCSS(A, B) = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ LCSS(Head(A), Head(S)) + 1 & \text{if } |r_n - s_m| < \epsilon \\ & \text{and } |n - m| \leq \delta \\ \max \left\{ \begin{array}{l} LCSS(Head(A), B), \\ LCSS(B, Head(B)) \end{array} \right\} & \text{otherwise} \end{cases} \quad (2.13)$$

where $\epsilon \in \mathbb{R}$ and $\delta \in \mathbb{Z}$. In LCSS, ϵ declares the threshold, which determines whether or not two elements match and δ is used to control how far in time we can go in order to match a given point from one trajectory to a point in another trajectory

LCSS is not a distance but can be converted into one by using equation (2.14)

$$LCSS_{dist}(A, B) = 1 - \frac{LCSS(A, B)}{\min(n, m)} \quad (2.14)$$

Another disadvantage of the LCSS is that it does not follow triangle inequality [14, 71] making it a non-metric function.

2.12 Summarizing

Table 2.1 is summing up the distance functions.

Distance Function	Different Lengths	Local Time Shifting	Noise	Metric
L_p norm	No	No	No	Yes
Discrete metric	No	No	No	Yes
Hausdorff	Yes	No	No	Yes
Partial Hausdorff	Yes	No	Yes	No
Modified Hausdorff	Yes	No	Yes	No
Bottleneck	No	No	No	Yes
Fréchet	No	No	No	Yes
Area of symmetric	Yes	No	Yes	Yes
Euclidean	No	No	No	Yes
DTW	Yes	Yes	No	No
LCSS	Yes	Yes	Yes	No

Table 2.1: Comparison among the distance functions

CHAPTER 3

Eye tracking

3.1	Introduction	20
3.2	Eye movements	20
3.3	Visualization	22
3.3.1	Gaze plot	22
3.3.2	Heatmaps	22
3.3.3	Average Polyline - APL	24
3.4	Structure and properties of eye tracking data	24
3.5	Calculation of Average Polyline (APL)	26

Figures

3.1	Histogram of all publications, relevant for eye tracking data visualization techniques. The number of published articles, conference papers, and books has strongly increased during the last decade [10]	20
3.2	Gaze plot visualizing the fixation and saccades movements occurring during the recording of eye movements ([52] adopted from Krassanakis [29])	23
3.3	Heatmap with different levels of intensity ([16] adopted from Krassanakis [29])	23
3.4	Different type of heatmaps (from left to right: Count, Absolute Duration, Relative Duration) [38] adopted from Krassanakis [29]	24
3.5	APL visualization. In the figure the raw and fixations points are also shown.	25
3.6	Calculation of the APL from 3 users	28

Tables

3.1	Taxnomy of fixations identifications algorithms [44]	21
3.2	Parameters of APL algorithm.	28

3.1 Introduction

Eyesight is one of the most important senses of a human. Through sight someone can perceive his surroundings. Unfortunately, the mechanisms of sight are not entirely known and there is a continuous effort to study and analyze these mechanisms.

The recording and the analysis of eye movements constitute an effective method for the exploration of several aspects related to human vision, perception and visual behavior. Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. It provides objective and quantitative evidences towards the examination of visual attention and a way to examine processes related to visual search, visual perception and cognitive process, which occurs during the observation of a stimulus [61, 28, 31, 57, 35]. Eye tracking has been gaining in popularity (see fig. 3.1) over the past decade as a window into observers' visual and cognitive processes.

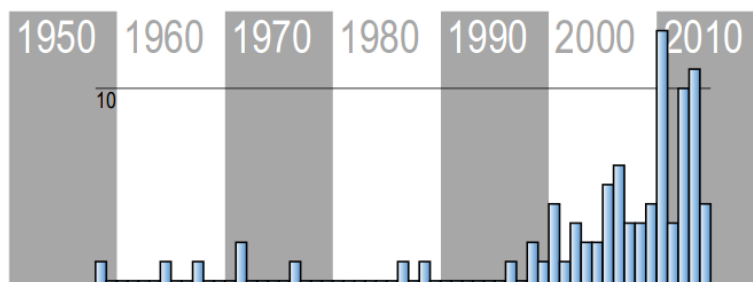


Figure 3.1: Histogram of all publications, relevant for eye tracking data visualization techniques. The number of published articles, conference papers, and books has strongly increased during the last decade [10]

The recording of the eye movement is performed using an apparatus. An eye tracker is the device responsible for making the measurements of the eye movements. Eye trackers are used in research of the visual system, in cognitive linguistics and in product design and have become very popular in a wide range of scientific disciplines from different research areas such as neuroscience, psychology, human and computer interaction (HCI). Specific applications include the tracking of eye movement in language reading, music reading, human activity recognition, perception of advertising, i.e. commercial eye tracking which includes web usability, marketing, automotive and the playing of sport [31, 57, 28, 35].

Eye tracking data have the same structure as data about movements of discrete objects in the geographic space. The main difference is that eye tracking data consist of gaze coordinates in a screen (and in the general case in the 3D world) and the tracking data consist of geodetic coordinates. So the object similarity functions mentioned in chapter 2 can be used in eye tracking data for extracting similarities between the gazes of two subjects. However, not all methods may be relevant since the movement properties and the possible questions of interest are not the same for eye movements and for road or sea traffic, human mobility, or animal migration. In the next sections the basic concepts of eye movements will be analyzed.

3.2 Eye movements

The main eye movements identified by the scientist are: *a*) saccades, *b*) fixations, *c*) smooth pursuits, *d*) vergence, *e*) vestibulo-ocular, *f*) nystagmus *g*) and conjugate [35]. But the key movements used in the analysis and recording of eye movements are only the first two (i.e. the saccades and the fixations) [29].

Criteria		Representative Algorithms				
		I-VT	I-HMM	I-DT	I-MST	I-AOI
Spatial	Velocity-based	X	X			
	Dispersion-based			X	X	
	Area-based					X
Temporal	Duration sensitive			X		X
	Locally adaptive		X	X	X	

Table 3.1: Taxonomy of fixations identifications algorithms [44]

A clear definition of the types of the main eye movements must be given, so the recording of the eye movements to be precise [35]. As already mention, the main eye movement can be classified to two categories:

Fixations: During the gaze of a scene, the eye focuses in various parts of the scene remaining relative still. Fixation occurs at the moment when the eyes are relative stationary in a position and the subject is focused on an object which attracts the attention. This moment is characterized by the miniature movements of tremors, drifts and microsaccades. Fixations represent the informations regarding the point that is being observed and their duration is between 100 msec and 1500 msec. 90% percent of the observation time is dedicated to fixations and they are important to the cognitive process. Fixations are able to indicate critical information regarding the cognitive processes that occur when a visual stimulus appears. During the gaze of the scene, the existence of many fixations points indicates that the subject is interested in the particular scene or that the scene appears to be complex. [35, 8, 43, 31, 33, 29]

Saccades: Saccades are eye movements, which are not part of the observation of a "critical" point. They are quick, simultaneous movement of both eyes that redefine the region of interest (leading to the next fixation). Their duration varies from 10ms to 100 ms. During a saccade little or no visual processing can be achieved, so these movements can not give any information regarding the complexity of the objects that are part of the visual scene, and thus the actual paths traveled during saccades are typically irrelevant for many research applications. However, regressive saccades (i.e. backtracking eye-movements) can act as a measure of processing difficulty [36, 35, 33, 8, 69, 29, 44, 43].

According to Salvucci and Goldberg [44] the process of fixation and saccades identification in eye-tracking is essential part of eye-movement data analysis and can have a dramatic impact on higher-level analyses.

There are numerous algorithms that identify fixations and saccades in eye tracking. In order to distinguish each type, the algorithms have several criteria based on spatial and temporal characteristics. Their criteria have velocity-based, dispersion-based and area-based attributes for spatial characteristics while the criteria that are related to temporal characteristics are the duration sensitivity and the local adaption of the algorithms (see table 3.1) [31].

Salvucci and Goldberg [44] have evaluated and compared fixation identification algorithms

and suggest that velocity-based and dispersion-based algorithms provide equivalent performance, while area-based algorithms seem to be more restrictive [31]. In a similar paper Shic, Scassellati, and Chawarska [49] have evaluated how outcome measures are impacted by changes to fixation algorithms.

3.3 Visualization

The visualization of the eye tracking data can reveal characteristics of the fixations, the saccades and the scanpath¹ structures [29, 10]. Several visualization methods for eye tracking data exists to help researchers to depict data collected in eye tracking experiments. Andrienko et al. [4] have done an extensive categorization of the visualisation techniques that exist and in the next sections we present the most widespread techniques.

3.3.1 Gaze plot

The most straightforward solution to visualize eye tracking data is with a simple plot of the pupil's horizontal and vertical coordinates against time. Similar techniques plot raw eye movements in 2-D with the stimulus image as the background. Traditionally, fixations are represented by circles, and saccades are represented by lines connecting the circles, indicating also the the order in which they occur. The center of each circle is placed in the position where the fixation happened and the size of the circle is proportional to the duration of the fixation. Larger radius of circle means bigger fixation duration (see fig. 3.2). Additionally, the third dimension may also be used in order to solve the problem of data overlapping [51, 29, 57]

This method is not suitable for large data due to enormous overplotting [4, 57].

3.3.2 Heatmaps

More advanced visualization techniques use the so-called heatmaps (aka fixation maps) to present the information in a more consistent manner (see fig. 3.3). A heatmap is a two-dimensional graphical visualization technique which represents points entities through different levels of intensity or different intensities of a color hue [51, 29, 51, 51, 57].

The visualization of the data with heatmaps is a tool through which complicated data, that would be difficult to be perceived with numerical data, is being summarized and understood. Heatmaps indicate the distribution of entities and this methodology can be quite useful in the direct observation of specific patterns or trends in the distribution of the gaze points [29].

Heatmaps can be easily generated using standard eye tracking software. The main categories of the heatmaps can be classified according to the properties of the fixations (see fig. 3.4) in which refer to and they can visualize counts of fixations, counts of different users who fixated on different areas, absolute gaze duration, and relative gaze duration (percentage to the total time spent) [29, 4]:

Heatmaps based on the number of fixations: In this case the visualization that is created is based on the number of fixations that occur in a visual scene. In this category the duration of the fixation is not considered.

Heatmaps based on the absolute duration of fixations: In this case, the heatmap is related to the absolute duration of the fixation that occur in the visual scene, without the number of fixations being considered.

¹A sequence of alternating fixations and saccades is called a scanpath. A scanpath can give information about the search behavior of a participant [10].

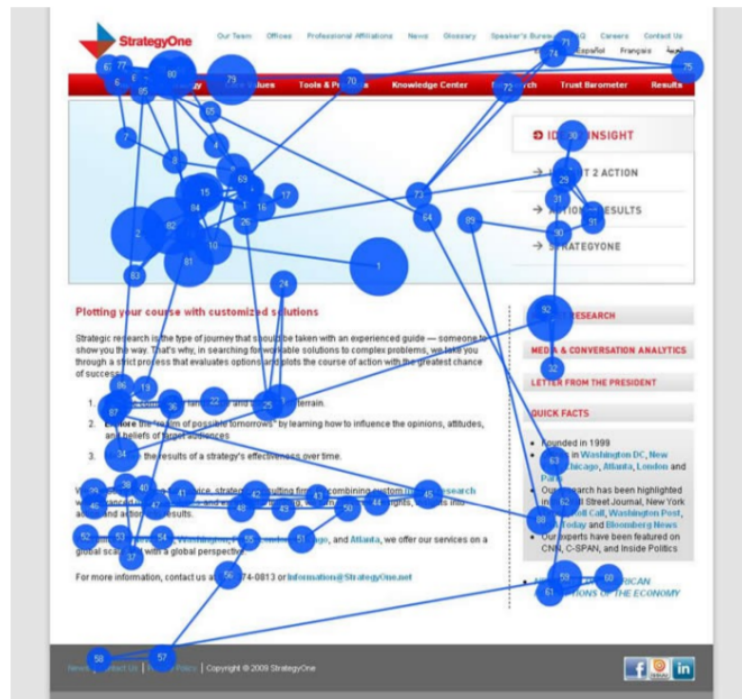


Figure 3.2: Gaze plot visualizing the fixation and saccades movements occurring during the recording of eye movements ([52] adopted from Krassanakis [29])

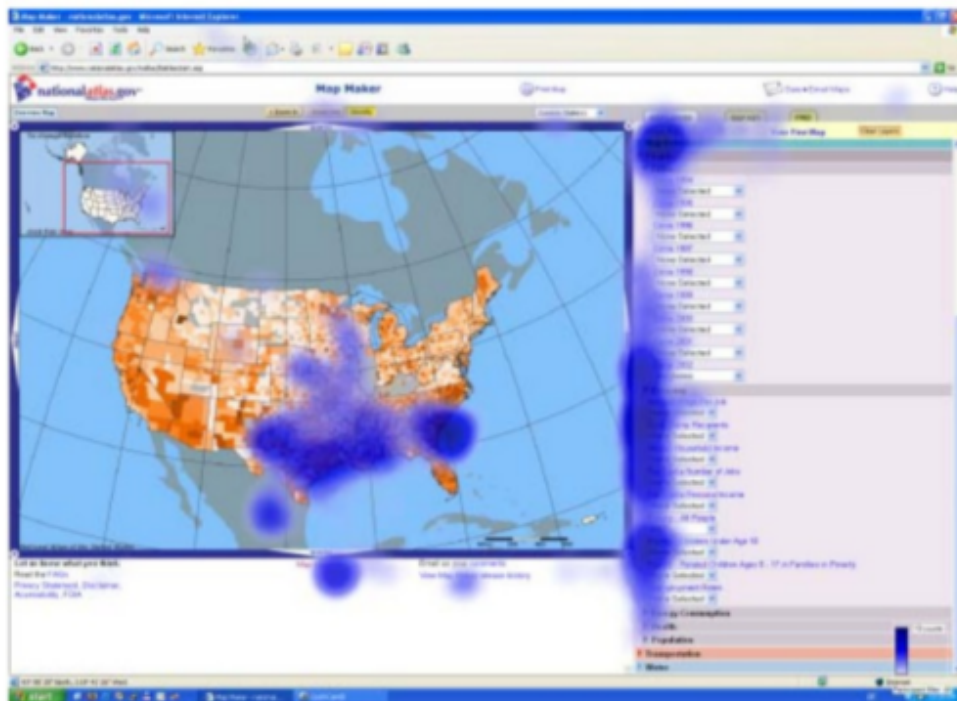


Figure 3.3: Heatmap with different levels of intensity ([16] adopted from Krassanakis [29])



Figure 3.4: Different type of heatmaps (from left to right: Count, Absolute Duration, Relative Duration) [38] adopted from Krassanakis [29]

Heatmaps based on the relative duration of fixations: In this case, the properties that the heatmap is based on is directly related to the relative duration of the fixations, i.e. the duration of each fixation relative to the total duration of all fixations of the visual scene.

Heatmaps based on the participants percentage: In this case, the heatmap is based on the percentage of the participants that gaze the different areas of the visual scene.

According to Andrienko et al. [4] "heatmaps may be useful for area of interest focused tasks. In comparative studies (different time intervals, different users, or different images) several heatmaps are compared. Eye tracking analysts also try to determine users' search strategies by analyzing series of heatmaps generated for consecutive time intervals, which show how the users' attention focus change over time. However, the characteristics of the eye movements, the links between the attention focus, and the paths followed during the search remain unclear".

3.3.3 Average Polyline - APL

APL is a new visualization and analysis tool introduced by Karagiorgou et al. [27] and Vescoukis et al. [57] and corresponds to the "average" line that is actually seen by subjects (see fig. 3.5 and section 3.5). Such a line might be useful in the study of various optical representations concepts, such as the assessment of the effects of alternative cartographic line attributes [27].

APL is the the depiction of the gaze route history using a polyline, which is feasible, as the visual trace is generated from sequential raw eye tracking data. This visualization is useful in cases where the context of eye tracking has reference to lines, paths, etc. that subjects are required or expected to follow [27].

3.4 Structure and properties of eye tracking data

Eye tracking data consist of vectors about the positions and times of gaze fixations. Each record usually includes the following two components: time, position in the display space (x- and y-coordinates). For fixations data also the fixation duration is included. The records may also include other attributes, e.g., stimulus identifier when different stimuli are used in the data collection. The temporally ordered sequence of records of one user referring to one stimulus is further called eye trajectory or scanpath, in the literature on eye tracking [4].

Tracking data have similar structure: moving object identifier, time, and position (in geographical space) defined by coordinates and maybe some additional attributes that may also be present. As already mentioned in 3.1 the structural similarity suggests that both classes of data may be analyzed using the same methods.

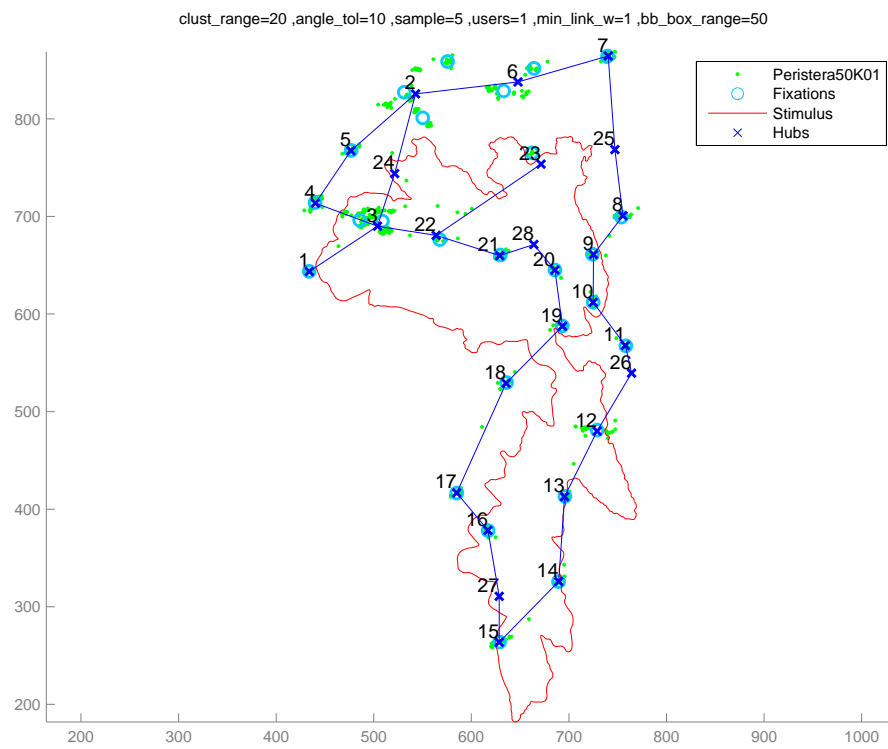


Figure 3.5: APL visualization. In the figure the raw and fixations points are also shown.

But according to Andrienko et al. [4], there is a significant difference between eye movements and movements of physical objects in the real world. Eye movements include instantaneous jumps (saccades) over relatively long distances, so the intermediate points between the start and end positions of a jump are not meaningful. It cannot be assumed that there exists a straight or curved line between two fixation positions such that the eye focus travels along it attending all intermediate points. This prohibits the use of methods involving interpolation between positions. Hence, not all movement analysis methods are valid for eye trajectories.

3.5 Calculation of Average Polyline (APL)

APL is produced by applying the TraceBundle [28] algorithm on eye tracking datasets, which was originally developed for the inference of graph geometries such as transportations networks, from GPS tracking data. The algorithm extract "hubs"² and constructs a polyline that correspond to the observed geometry and the aim of this approach is to derive a single polyline geometry from sampled eye tracking data from multiple users. This line is call the Average Polyline (APL) and represents the line that the subjects have followed on average. An APL can be used to help researchers understand how visualizations, distractions or even subjects' characteristics affect what has been seen in such an experiment [28, 57].

The proposed algorithm to derive the polylines from eye tracking data involves three steps: *a)* identifying hubs, *b)* connecting hubs and *c)* reducing the links into a single geometry to generate the APL [57, 27]

In the first step, hubs are inferred from spatial fixations on eye tracking data. According to Karagiorgou et al. [27] indicators for hub recognition are the number of tracking samples, the number of different users and the coverage of an extended area of focus. The algorithm takes as input the eye tracking data and determines the *k*-NNs of each tracking sample, which are filtered according to the number of users. On the filtered samples, DBSCAN clustering algorithm is applied using a distance threshold and a minimum number of samples. The centroids of the resulting clusters are the hubs. Hubs are very similar in nature with fixations.

Next, the hubs are connected by links. For each hub, the outgoing and/or incoming tracking portions connecting this hub to other is recorded by scanning all eye tracking data to discover sequences of hubs. The result of this step is the creation of a sample polyline set that connects hubs with links (this process is shown in algorithm 3.1). At this stage redundant links between hubs are introduced [57, 27, 28].

In the last step of the algorithm, the links are compacted to a reduced geometry expressed by polylines. The algorithm identifies tracking portions that are close to existing links by means of a buffer region and merges their geometry into the existing link geometry. In this step, only the geometry of existing links is adjusted using a three-step algorithm: *a)* sort existing link samples in a descending order according to their length (so to process longer links first as they may be more significant for polyline construction), *b)* determine relevant tracking portions using a buffer region around link samples and *c)* adjust the geometry of links based on the tracking data geometry (see algorithm 3.2) [57, 27, 28]. In table 3.2 the input parameters of the algorithm are summarized and in fig. 3.6 an example of the APL algorithm is shown.

According to Vesoukis et al. [57] "APLs provide a new means for visualizing eye-tracking data, which can be used along heatmaps and other methods to extract useful conclusions, depending on the specifics of the experiment".

²According to Karagiorgou et al. [27] a hub represents the spatial fixation that the eye creates near an area of interest.

Algorithm 3.1: Connecting hubs [57]

```

Input: Set of eye-tracks  $T$  and hubs  $H$ 
Output: A set of link samples  $L_S$ 
/* Connecting hubs using tracking data */
/* Hub sequence */
1  $H_S \leftarrow 0$ ;
/* Link samples */
2  $L_S \leftarrow 0$ ;
/* Identify hub sequences from tracks */
3 foreach  $t \in T$  do
4   foreach  $p \in t$  do
5     /* position sample is mapped to a hub */
6     if  $p \in H$  then
7       /* record prev., current hub */
8        $H_S \leftarrow h^-, h, p^-, p, t$ ;
/* Collect and merge link samples */
9 foreach  $h^-, h \in H$  do
10  /* add all track portions for this hub pair */
11  foreach  $t \in H_S$  do
12     $L_s \leftarrow t, p^-, \dots, p$ ;
/* cluster link samples */
13  $Width \leftarrow Width(L_S)$ ;
14  $Weight \leftarrow Weight(L_S)$ ;
15  $L_S \leftarrow SweepMerge(L_S)$ ;

```

Algorithm 3.2: APL extraction [57]

```

Input: A set of link samples  $LS$ 
Output: A set of links  $L$ 
/* Sorting link samples by length */
1  $LS \leftarrow sort(LS, length)$ ;
/* Candidate link samples */
2  $CLS \leftarrow 0$ ;
/* width of link samples */
3  $Width$ ;
/* direction threshold */
4  $Angle$ ;
5 foreach  $l \in LS$  do
6    $CLS \leftarrow Find(bbox(l, Width(l), Angle))$ ;
7   foreach  $cl \in CLS$  do
8     if  $Contains(l, cl)$  then
9        $LS \leftarrow SweepMerge(l, cl)$ ;
10    else
11      /* partial overlap */
12       $cl_{in}, cl_{out} \leftarrow Split(l, cl)$ ;
13       $CLS.add(cl_{out})$ ;
14       $L \leftarrow SweepMerge(l, cl_{in})$ ;

```

Parameters	Explanation
Proximity and angle difference	DBSCAN clustering algorithm is applied using a distance threshold and a minimum number of samples
Number of tracking samples, the number of different users and the coverage of an extended area of focus	Indicators for hub recognition are the number of tracking samples, the number of different users and the coverage of an extended area of focus. The algorithm takes as input the eye tracking data and determines the k-NNs of each tracking sample, which are subsequently filtered according to the number of users.
Bounding box	The algorithm identifies trajectory portions that are close to existing links by means of a bounding box and merges their geometry onto the existing link geometry, using a bounding box around link samples to determine relevant trajectory portions

Table 3.2: Parameters of APL algorithm.

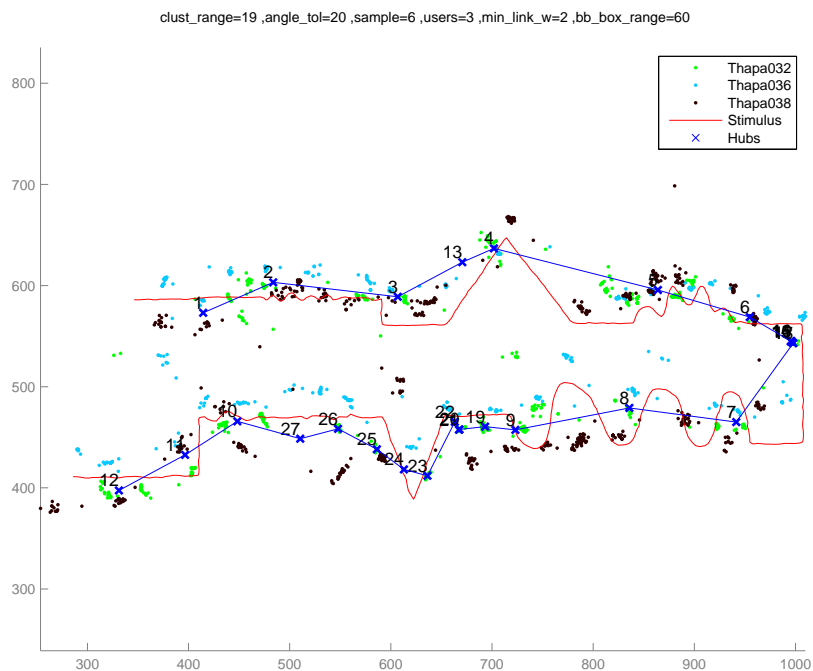


Figure 3.6: Calculation of the APL from 3 users

CHAPTER 4

Toolkit

4.1	Introduction	30
4.2	Concepts and work flow	30
4.3	Software	30
4.4	Requirements outline	33
4.4.1	Non-functional requirements	33
4.4.2	Functional requirements	34
4.5	Case study	35
4.5.1	Eye Tracker	37
4.5.2	Stimuli	37
4.6	Analysis of samples	38
4.6.1	Type 1 question	39
4.6.2	Type 2 question	44

Figures

4.1	Structure of table T to used in analyzing APLS and individual's behavior.	33
4.2	The APL analysis toolkit	35
4.3	The MATLAB® toolkit	36
4.4	The Eye Tracking system, used in the NTUA experiments [8]	37
4.5	The geometry of the subject and the screen [8]	38
4.6	Stimuli	39
4.7	Heatmap of the stimuli [7]	40
4.8	APL of the stimuli	41
4.9	Subjects APL with reference line	42
4.10	Normalized distribution of Hausdorff distance across experiments	43
4.11	Normalized Hausdorff distance distribution across groups for the "Thapa" experiment	44
4.12	Normalized distribution of L_i/L_{ref} across groups for the "Thapa" experiment.	45

Tables

4.1	An overview of the freely available tools for eyemovement analysis (modified from Krassanakis, Filippakopoulou, and Nakos [31]).	31
4.2	The normalized distribution of the two experiments	43
4.3	The normalized distribution of the two experiments	45

4.1 Introduction

When working with eye tracking data, it is common to determine some measurement of similarity between many subjects gaze sequences. APLs (see section 3.3.3) enable the quantitative analysis of the individuals' behavior compared to the average behavior, using established similarity measures. Other researchers, determined similarity by using simple visual analysis. This analysis is prone to error and bias, whereas computational analysis generates concrete and reproducible results [60, 57].

The purpose of this toolkit is to support the researchers in productively analyzing gaze data and producing similarity scores between gaze data in a quantitative way.

4.2 Concepts and work flow

The algorithm discussed in section 3.3.3, will be used to extract a set of hubs and connecting links which represent the APL. After the average polyline is extracted, the whole set will be further processed and the behavior of each individual will be analyzed and compared to the "average behavior" of the sample from which the APL was produced. By following this approach, two types of questions may be answered [57]:

Type 1: Visualization attributes How does a different stimulus affect the behavior of the same population?

Type 2: Individual's behavior How do groups of individual subjects with similar properties deviate from the average behavior when the same stimulus is presented?

According to Vescoukis et al. [57] "by being able to compare the behavior of each individual to the average behavior of the population participating in an experiment, we can perform quantitative statistical analysis on groups of subjects with common attributes".

The work flow to process a set of eye tracking data is shown in algorithm 4.1.

Algorithm 4.1: APL analysis workflow [57]

Input: Set of eye-tracking data for a given experiment E

Output: APL, table T containing set of individuals PL (PL_i), deviation metrics (D_i)

- 1 adjust the APL extraction algorithm parameters ;
 - 2 calculate the APL for E using the APL extraction algorithm ;
 - 3 **foreach** $S_i \in E$ **do**
 - 4 calculate PL_i by applying the APL extraction algorithm on data in S_i ;
 - 5 calculate deviation metrics D_i between PL_i and APL ;
 - 6 add PL_i, D_i to table T ;
 - 7 perform statistical analysis on table T ;
-

This approach can enable many new applications in education, behavioral and intention analysis, psychology, and elsewhere [57].

4.3 Software

Many tools have been developed in order to implement specific approaches in eye movement analysis. In table 4.1 there is an overview of the freely available software.

EyeMMV toolbox (Eye Movements Metrics & Visualizations toolbox) is a MATLAB® toolbox for post experimental eye movement analysis. It includes functions for fixations

Analysis tool	Developers	Site
EyeMMV toolbox	Krassanakis, Filippakopoulou, and Nakos [31]	https://github.com/krasvas/EyeMMV
ILAB	Gitelman [21]	Not available any more
Eyelink Toolbox	Cornelissen, Peters, and Palmer [17]	http://psychtoolbox.org
iComp	Heminghous and Duchowski [25]	http://andrewd.ces.clemson.edu/iComp/
openEyes	Li, Babcock, and Parkhurst [34]	http://www.openeyes.org.uk/
eyePatterns	West et al. [60]	http://sourceforge.net/projects/eyepatterns/
ASTEf	Camilli et al. [13]	www.astef.info
ETU-driver	Špakov [72]	http://www.sis.uta.fi/~csolsp/downloads.php
OGAMA	VoßKühler et al. [58]	http://www.ogama.net/
ITU Gaze Tracker	San Agustin et al. [45]	http://sourceforge.net/projects/gazetrackinglib/
GazeAlyze	Berger et al. [9]	http://gazealyze.sourceforge.net/
GazeParser	Sogo [50]	http://gazeparser.sourceforge.net/

Table 4.1: An overview of the freely available tools for eyemovement analysis (modified from Krassanakis, Filippakopoulou, and Nakos [31]).

identifications, metrics analysis, data visualization and ROI (region of interest) analysis [31].

ILAB is a MATLAB® toolbox which allows to *a)* display basic eye movement data, including scan paths, fixations, and saccades *b)* allow basic quantification of analyzed eye movement data, including tabulation of fixations and saccades and correlation of eye position with a region of interest (ROI) *c)* allow accurate mapping between the eye tracker and the computer screen *d)* enable display of eye movement data on the same images as those the subject was viewing *e)* display pupil size data *f)* enable export of data and results to facilitate analyses in other applications *g)* apply any of the preceding methods to a variety of eye-tracking systems and *h)* develop a system that allowed expandability over time [21].

Eyelink Toolbox is a MATLAB® toolbox that supports the measurement of eye movements. The Eyelink Toolbox, in combination with the Psychophysics Toolbox ¹, provides a fast, easy, interactive, and powerful means to develop research-grade eye-movement paradigms [17].

iComp is an open-source visualization tool that implements quantitative scanpath comparison in loci and sequence [25].

openEyes consists of an open-hardware design for a digital eye tracker that can be built from low-cost off-the-shelf components, and a set of open-source software tools for digital image capture, manipulation, and analysis in eye-tracking applications [34].

eyePatterns is a software for identifying patterns and similarities across fixation sequences. This software tool provides trusted sequence analysis techniques and an intuitive interface that guides the user from identifying which experimental variables may influence

¹<http://psychtoolbox.org/>

fixation sequences to discovering patterns shared by similar sequences. Additional sequence analysis techniques, such as compression and filtering of sequences are also planned [60].

ASTEF (A Simple Tool for Examining Fixations) is data analysis application for processing eye-movement information. ASTEF is developed in C#.NET and enables the visualization, analysis of fixation data and fixation identification [13].

ETU-driver (Eye-Tracking Universal Driver) has been developed as a software layer to be used between the actual eye tracker driver and the end-user application to provide device-independent data access and control. The ETU-Driver consists of a COM objects that implements the interface common for all eye trackers and a set of supporting DLL libraries (API-Converters), which "convert" original manufacturer's APIs into a common API used by ETU-Driver. The benefit of using ETU-Driver comes from the fact that any end-user application implemented on top of ETU-Driver can access data from a newly installed eye tracker simply by copying a corresponding API-converter to the ETU-Driver installation folder [72].

OGAMA (Open Gaze and Mouse Analyzer) is developed in C#.NET. Its main features include slideshow design, the recording of gaze and mouse data, database-driven preprocessing and filtering of gaze and mouse data, the creation of attention maps, areas-of-interest definition, and replay. Eyetracking and/or presentation soft- and hardware recordings in ASCII format can be imported. Data output is provided that can be used directly with different statistical software packages. Because it is open source, one can easily adapt it to suit one's needs [58].

ITU Gaze Tracker is developed in C#. There are three main components: *a)* the gaze tracking library, which implements all the methods to control a tracker such as extracting eye features, run a calibration procedure, estimate the gaze coordinates and detect the type of eye movement *b)* the camera class, responsible for initializing a generic camera and grabbing images that are then processed by the gaze-tracking library and *c)* the user interface, which provides the communication with the gaze-tracking library to set up the different parameters of the system [9].

GazeAlyze is a MATLAB® toolbox for the analysis of eye movement data and was developed to analyze gaze data for static visual stimuli and to obtain parameters from fixations and saccades for further visualization and statistical analysis. It includes detecting and filtering artifacts, detecting events, generating regions of interest, generating spread sheets for further statistical analysis, and providing methods for the visualization of results, such as path plots and fixation heat maps. GazeAlyze also includes functions for correcting eye movement data for the displacement of the head relative to the camera after calibration in fixed head mounts [9].

GazeParser is an open-source library for low-cost gaze position recoding, eye tracking and data analysis. The libraries used in GazeParser are written in Python and can be used in conjunction with PsychoPy² and VisionEgg³ experimental control libraries [50].

The only software that supports similarity analysis of gaze data is the eyePatterns [60]. To accomplish that it uses two algorithms to determine the mathematical similarity between string sequences. The first is the Levenshtein distance algorithm (a.k.a. the string-edit algorithm) and the second is the Needleman-Wunsch algorithm that, unlike the basic string-edit algorithm, allows the user to specify scoring parameters that are optimal for a given experiment [60].

Unfortunately eyePatterns is not suitable for our work due to the fact that the stimulus must be divided into Areas of Interest (AOIs). Each fixation that resides into a specific AOI is

²<http://www.psychopy.org/>

³<http://visionegg.org/>

assigned a one-character and the fixation sequence is converted to a string sequence. Afterwards the program compares the strings, effectively removing any geometrical informations that the fixations had.

4.4 Requirements outline

Software requirements in systems engineering and software engineering express the needs and constraints placed on a software product and encompasses those tasks that go into determining the needs or conditions for a new or altered product or project, taking account the conflicting requirements of the various stakeholders [11, 68].

As already mentioned, the purpose of the toolkit is to support the researchers in productively analyzing gaze data and producing the output of the suggested work flow (see section 4.2 and algorithm 4.1).

The user should be able to calculate the differences between the two polylines and extract the histogram of the differences between each PL_{*i*} and the APL to perform further analyses. With this toolkit we would like to be able to calculate a table *T* like in figure 4.1 needed to further analyze how specific subject attributes affect the gaze of the subject and the APL of a particular population. In the table, (*i*) denotes the subject ID and *k* denotes the deviation metric.

Structure of table <i>T</i>
<i>ExperimentID</i>
<i>SubjectID(i)</i>
<i>Subject classifications C_i</i>
<i>PL_i calculation parameters</i>
<i>PL_i length</i>
<i>Absolute Dist_k</i>
<i>Normalized Dist_k</i>

Figure 4.1: Structure of table T to used in analyzing APLS and individual's behavior.

The significance of each attribute in a population is shown in histograms. By filtering the results according to a specific attribute and creating each attributes histogram, someone will be able to extract the significance of the attribute in the gaze of the subject from the differences in the histograms.

The present sections describe the development of a new toolbox for post experimental analysis.

4.4.1 Non-functional requirements

In software system engineering, a software requirement that describes not what the software will do, but how the software will do it, is a non-functional requirements and are the ones that act to constrain the solution [11, 15]. The non-functional requirements that were in mind before the development of the toolkit were:

Cross-platform: smooth operation on all operating systems.

Modular architecture: in order to allow future extension and upgrades.

Graphical User Interface: Creation of a Graphical User Interface (GUI) for the efficient use of the toolkit and the ease of learning.

Error messages: for alerting users in terms familiar to them.

Documentation: of the toolkit by means of detailed description of each function in the code base and creation of manuals for the users.

To ensure the aforementioned in an easy way, it was decided the toolbox to be developed using MATLAB® [37]. MATLAB® (MATrix LABoratory) is a high-level and fourth-generation programming language designed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. It is used in many domains such as signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology [65, 37].

This toolkit could have been written in a low-level computer language such as C++, which would have freed it from dependency on other software. However, although low-level languages are very powerful and flexible, they are not as conducive to rapid program development or easy implementation of graphical interfaces. Similar with interpreted languages, such as Python, which allow easier program development than low-level languages do, still they do not include all the prepackaged algorithmic and graphical tools in the basic version of MATLAB®.

So even though MATLAB® is only available commercially, it is commonly used in research, and the advantages that offers overcome the disadvantage of being a commercial product. With MATLAB® the developed software can be executed in every operating system (Windows, Linux or Mac OS) where MATLAB® is installed and with GUIDE⁴ the creation of a users interface is being simplified. In a later stage someone could also replace MATLAB® with GNU Octave⁵.

Also, through the GUI the user must able to select the files that contains the raw data of the experiment and calculate the fixations, the hubs and the APL. By using the toolkit interface the user is able to calculate the APL for the population he chooses and visualize the results.

4.4.2 Functional requirements

A software requirement specifies a function that a software system or software component must be capable of performing. For example, formatting some text or modulating a signal. They are sometimes known as capabilities or features [11, 15].

With this toolkit the user must be able to:

Load multiple files: The toolkit must be able to load multiple files of raw eye tracking data. The files must have a .raw extension and the file must have a header and each line to be a triple of (Time, x, y). An example is shown below:

```

1 Time,X,Y
2 41.0074,430.2848,643.328000
3 41.0239,433.8688,644.096000
4 41.0406,431.2064,643.456000
5 41.0572,434.3808,643.456000
6 41.0739,433.2544,643.456000
7 41.0905,434.7904,642.560000
8 41.1072,432.6400,643.072000
9 41.1238,434.9952,643.456000
10 .
11 .
12 .
```

⁴<http://www.mathworks.com/discovery/matlab-gui.html>

⁵<http://www.gnu.org/software/octave/>

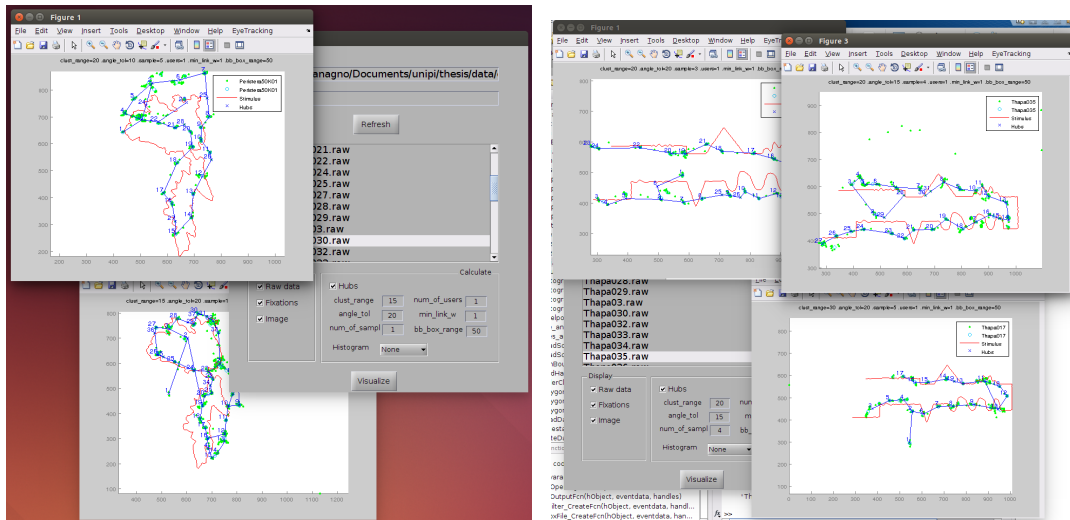


Figure 4.2: The APL analysis toolkit

Visualize stimulus: If in the same folder with the data exists a file with .stim extension, that files contains the stimulus of the experiment and must be given in pairs of (x,y). Also if a .bgd file exists that files contain the background of the stimulus and it has the same properties with .stim file. An example is shown below:

```

1 634.79,572.13
2 634.53,572
3 634.26,571.9
4 634.01,571.77
5 633.76,571.62
6 633.49,571.5
7 633.28,571.31
8 .
9 .
10 .

```

Fixations calculation: The toolkit must be able to calculate the fixations of the gaze data. If a file with .fix extension exists and the name of the file is the same with the name of .raw data file, it is assumed that the files contains the fixations of the particular raw file and they wont be recalculated.

APL caclulation: The toolkit must be able to calculate the *APL* (see section 3.5) for a batch of .raw files. The parameters of the algorithm must be provided by the user and the toolkit must export the hubs and the links calculated from the algorithm.

Similarity comparison: Finally the toolkit must be able to calculate the similarity function (see chapter 2) for *APL* and the PL_i . The user must be able to choose the data and the toolkit will calculate the distance function between the data.

In figure 4.2 and 4.3 the toolkit is being shown.

4.5 Case study

A case study is presented, that uses the software capabilities, to demonstrate the use of the APL and the similarity analysis of eye tracking data. The data used in this case study, comes

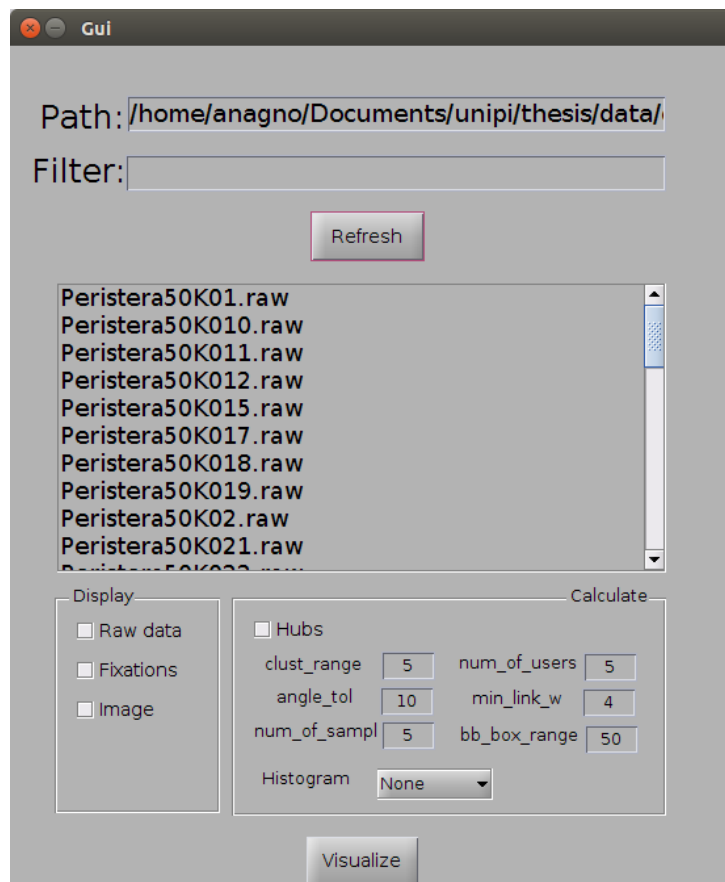


Figure 4.3: The MATLAB® toolkit

from the National Technical University of Athens (NTUA) and they were already used in two publications ([8, 7]). Specifically, the experiment took place in the laboratory of the School of Rural and Surveying Engineering.

In the case study to be discussed in the sequel, we have used two such metrics, namely the Hausdorff distance H and the relative to the APL length of the subject's PLi.

4.5.1 Eye Tracker

The system for recording the gaze of the subjects is the Viewpoint Eye Tracker® from Ar-rigton Research, which is installed at the Laboratory of Cartography of National Technical University of Athens (NTUA). Viewpoint Eye Tracker allows the recording of the position and movement of the subjects eyes on a visual scene displayed on a computer monitor connected with the system.

The operation of this system is based on the use of devices that are able to record the gaze on a visual scene by analyzing eye images (see fig. 4.4). The system consists of *a*) the recording device (hardware) and *b*) the processing tools (software). The system is embedded in a computer, which supports two display monitors. The primary monitor is used from the system's operator and the second monitor is used for the projection of the stimulus to the subject.

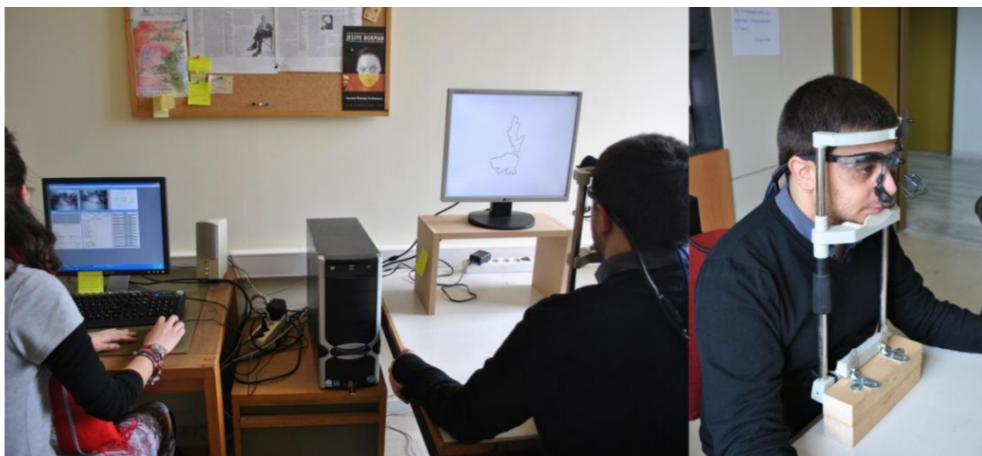


Figure 4.4: The Eye Tracking system, used in the NTUA experiments [8]

The recording device consists of a camera and an infrared Light-Emitting Diode (LED) for each eye, which are attached to the optical system of the subject. The infrared LED, which is located under the observer's eye, illuminates the eyeball (the illumination contributes to the discrimination between the regions of the pupil and the iris), while its image is being captured by the camera placed next to the infrared LED. The system's geometry is completed by a mechanism that immobilize the position of the observer's optical system [36, 29, 30].

The main disadvantages of the Viewpoint Eye Tracker are *a*) that for the successful recording of data, the immobilization of the head of the subject at a fixed position is needed (see fig. 4.4) and *b*) that the subjects that use the system must have accurate vision without the need for glasses or contact lenses, due to the limitations of the system.

4.5.2 Stimuli

In this experiment, the head of the subject was anchored 60 cm from the screen and only the movement of the right eye were recorded with a frequency of 60Hz and an accuracy that varies between 0.25-1.00 degrees of visual arc (see fig. 4.5)[8, 7]

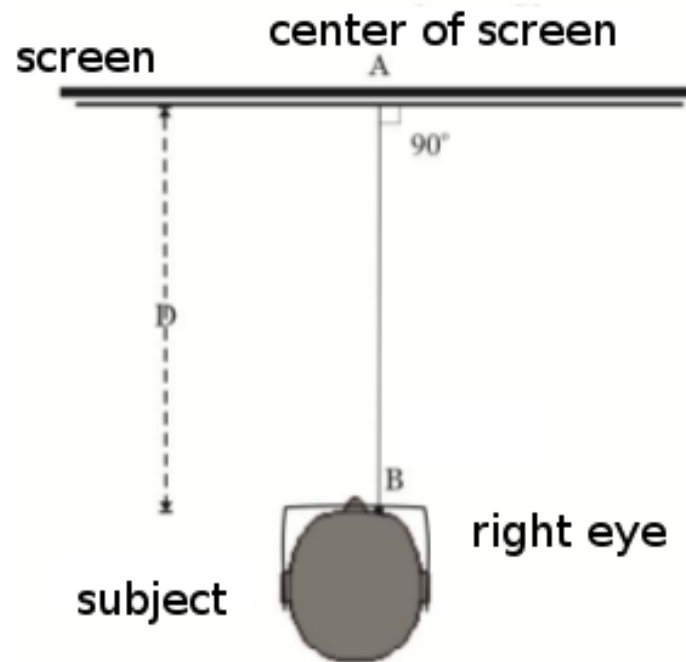


Figure 4.5: The geometry of the subject and the screen [8]

The subjects that participate in the experiments were persons aged from 18 years to 40 years and from various professional areas, with different educational levels and accurate vision.

The stimulus that were displayed to the subjects were:

- The coastline of the Peristera island, which is located in Sporades, at scale 1:50.000 approximately (see fig. 4.6a).
- The geometrical model consisted of several representative line shape samples introduced by Thapa [53] (see fig. 4.6b)

The line's width was modified taking into consideration the distance of the 60 cm between the subject and the projection's monitor. Stimuli have 1280 x 1024 pixels resolution. For more information about the original experiments the readers are referenced to Bargiota [8] and Bargiota et al. [7]

4.6 Analysis of samples

In this case study, we executed the work flow discussed in section 4.2 and created the table for each stimulus (T_P for Peristera stimulus and T_T for Thapa stimulus).

TraceBundle (see section 3.3.3) algorithm is used to create a polyline that the subjects have followed "on average" and a set of hubs and connecting link are extracted from the data provided (see section 3.5). Each subject in the experiments was asked to follow the linear entities described in section 4.5.2 and afterwards the analysis of the data followed using the Hausdorff distance and the Modified Hausdorff distance.

For each subject an average gaze line was produced using the APL (see table A.1 and B.1 in the Appendix) and that line was compared to the "average line" produced by many subjects.

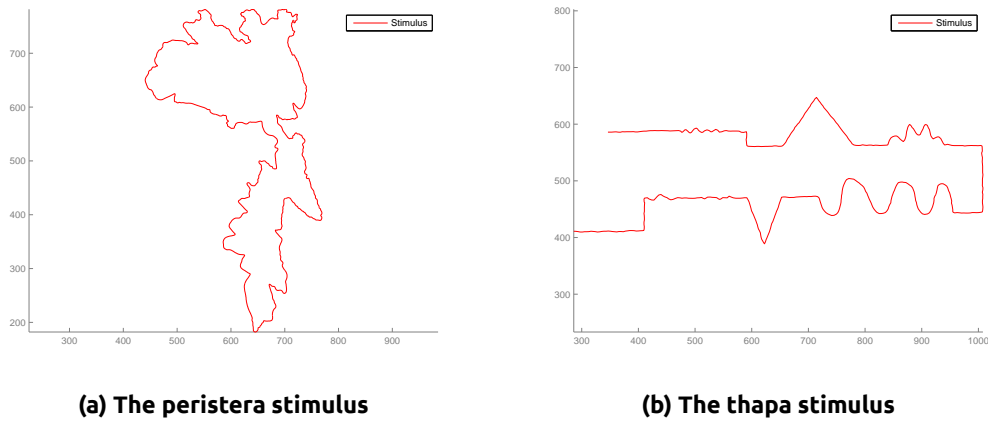


Figure 4.6: Stimuli

The visualization of the aforementioned datasets using heatmaps (see section 3.3.2) and APL (see section 3.3.3) are shown in fig. 4.7 and 4.8 respectively. As a reference line, in the case of the Peristera stimulus the APL introduced from Karagiorgou et al. [27] was used and in the case of the Thapa stimulus, 3 subjects were chosen and the APL was created.

Only two of the distance functions discussed in chapter 2 were used, namely the Hausdorff distance (see section 2.4) and the modified Hausdorff distance (see section 2.4.2). The reason for using only these two metrics were due to the limitations of our data. Although our original data did include a time dimension (when the gaze was recorded), after applying the TraceBundle, this information is lost since the algorithm only produces hubs and links (location of the processed gaze data). So algorithms that include the time dimension (i.e. trajectories similarity - see section 2.8) are not possible to be used. Also our sets are not compact and the cardinality of our data can't be consider equal (since most of the times we won't have the same number of observations) so the area of symmetric difference (see section 2.7) and the Bottleneck distance (see section 2.5) are not applicable metrics. Finally the Fréchet distance (see section 2.6) would have been a suitable metric but due to the nature of the produced data (points connected with edges), there wasn't a straightforward way of producing a correct curve, so the use of a Fréchet algorithm was impossible.

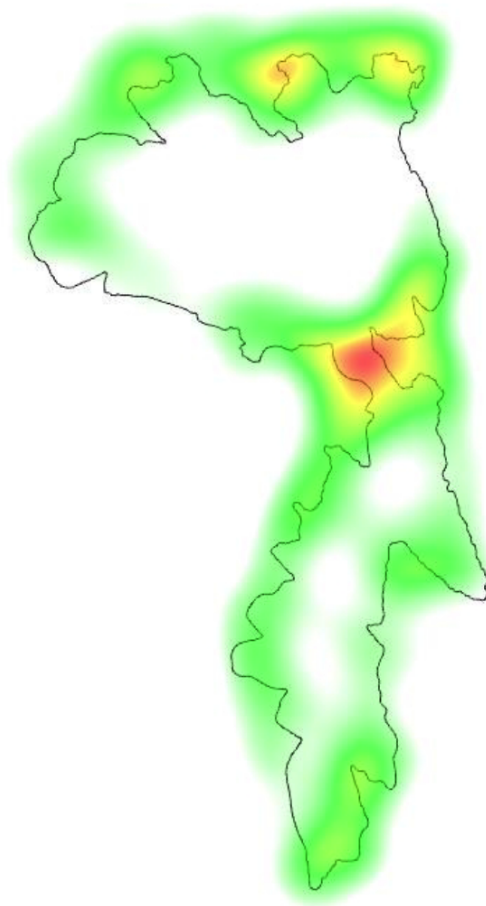
As already mentioned in chapter 2 the simplicity of the Hausdorff distance may lead from time to time to wrong conclusions since it does not consider the polylines as polylines but as point sets. In figure 4.9 three cases of the subjects APL and the reference line are shown. Although Hausdorff distance is simple it captures the differences between the polylines. The more different a polyline is, the bigger the Hausdorff distance and the Modified Hausdorff distance is.

4.6.1 Type 1 question

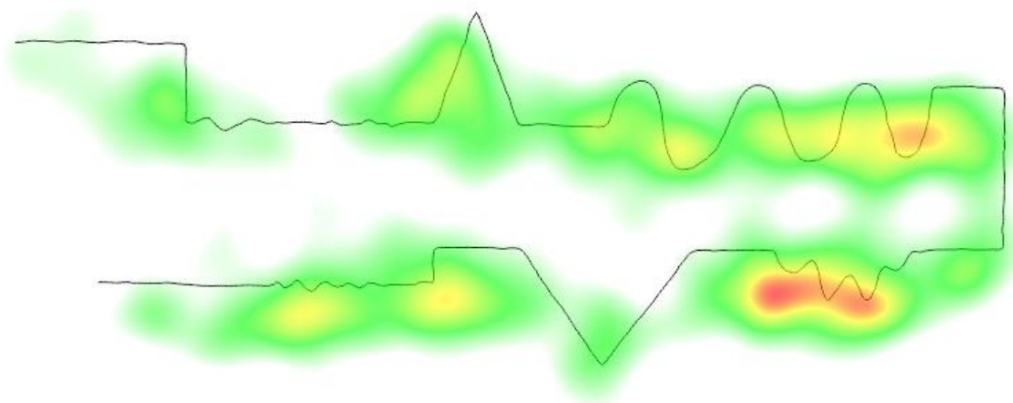
In figure 4.10 the normalized distribution of the RH_i is shown. RH_i is calculated as the Hausdorff distance of each sample PL_i from the APL, divided by the length of the APL (see equation (4.1) and table 4.2).

$$RH_i = \frac{\delta_H(PL_i, APL)}{length(APL)} \quad (4.1)$$

The two experiments have not been executed with this kind of analysis in mind, but it seems from the results that the subjects deviate less from the "average line" in the simpler cartographic line of Thapa.

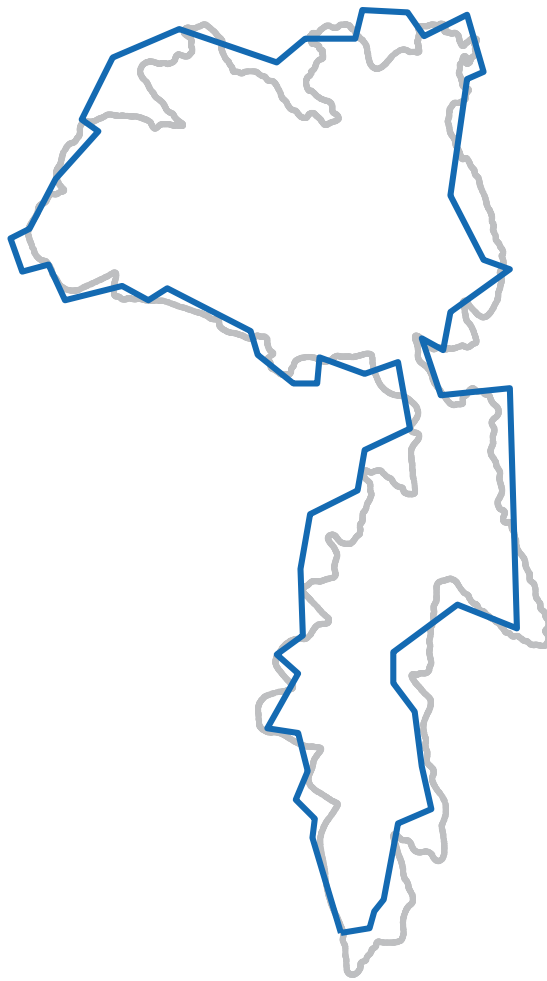


(a) The Peristera heatmap

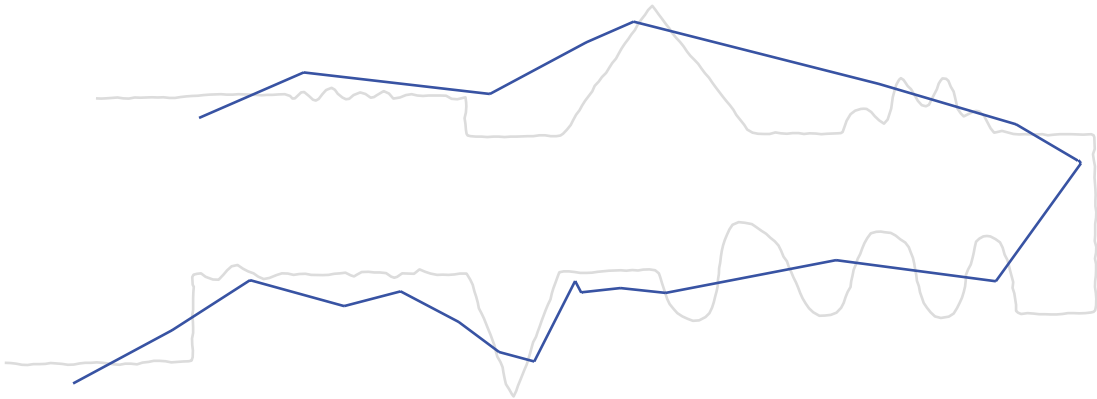


(b) The thapa heatmap

Figure 4.7: Heatmap of the stimuli [7]

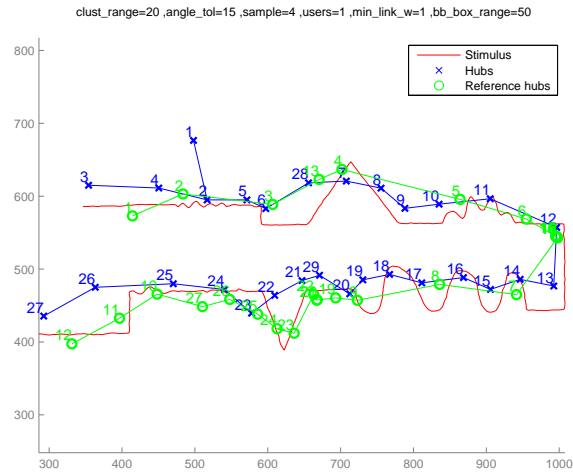


(a) The Peristera APL [27]

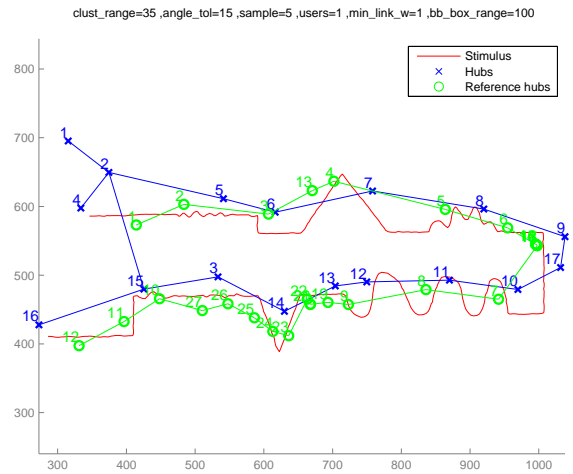


(b) The Thapa APL

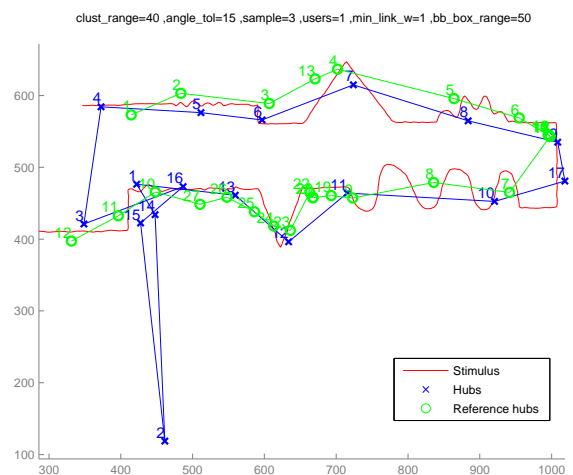
Figure 4.8: APL of the stimuli



(a) Subject APL with small hausdorff distance (76.717) from reference line



(b) Subject APL with medium distance (157.3236) from reference line



(c) Subject APL with large distance (307.6057) from reference line

Figure 4.9: Subjects APL with reference line

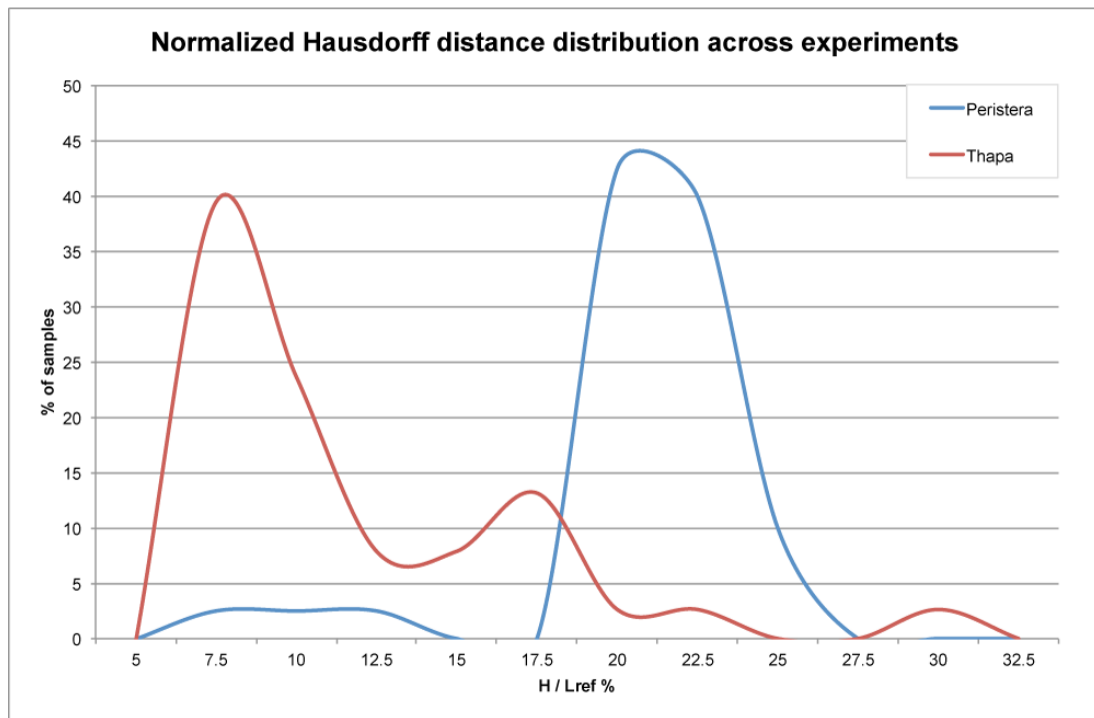


Figure 4.10: Normalized distribution of Hausdorff distance across experiments

Frequencies	Peristera RH_i	Peristera $RH_i\%$	Thapa RH_i	Thapa $RH_i\%$
5	0	0	0	0
7.5	1	2.5	15	39
10	1	2.5	9	24
12.5	1	2.5	3	8
15	0	0	3	8
17.5	0	0	5	13
20	17	42.5	1	3
22.5	16	40	1	3
25	4	10	0	0
27.5	0	0	0	0
30	0	0	1	3
32.5	0	0	0	0

Table 4.2: The normalized distribution of the two experiments

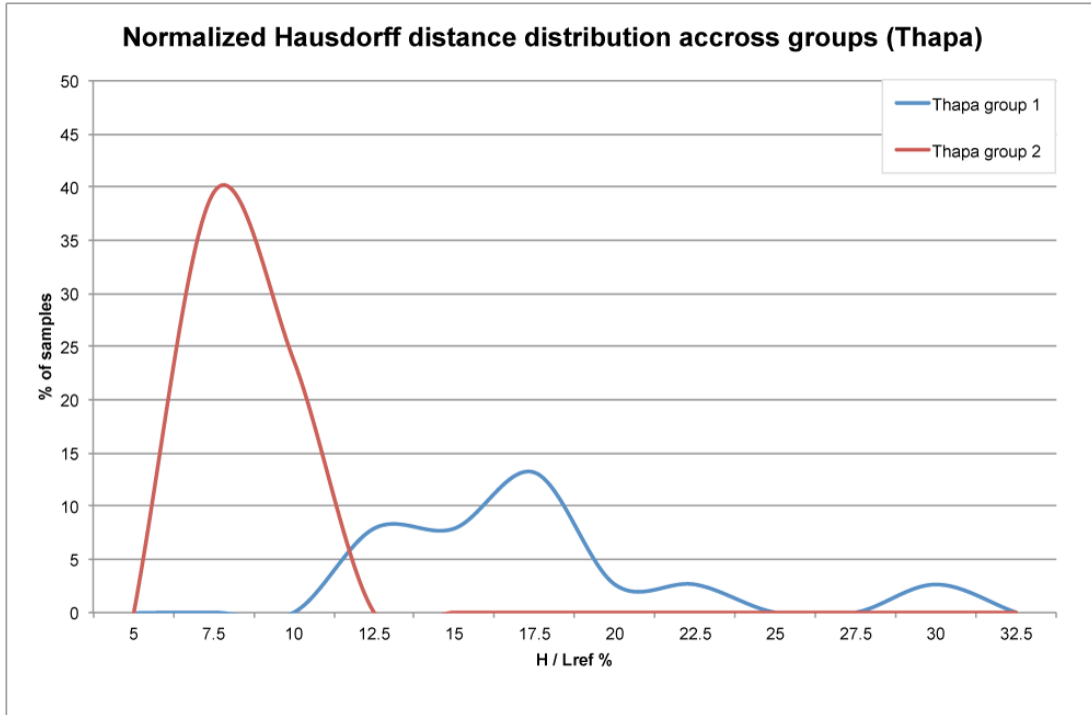


Figure 4.11: Normalized Hausdorff distance distribution across groups for the "Thapa" experiment

4.6.2 Type 2 question

To study how attributes of different individuals influence the results of the APL, we studied the distribution of two metrics. The first is the one presented in type 1 question and equation 4.1. The second is based on ratio calculated from the length of each PL_i divided by the length of the APL (see equation 4.2)

$$R_i = \frac{L_i}{L_{ref}} = \frac{\text{length}(PL_i)}{\text{length}(APL)} \quad (4.2)$$

The distributions were calculated for 2 groups of subjects. The first group consists of 14 people aged over 45 where the 24 members of group 2 are aged below 40. The distributions are shown in fig. 4.11 and 4.12 accordingly.

From the diagrams is noticed that the deviations of individuals observations in group 1 are noticeably higher than the corresponding deviations of group 2.

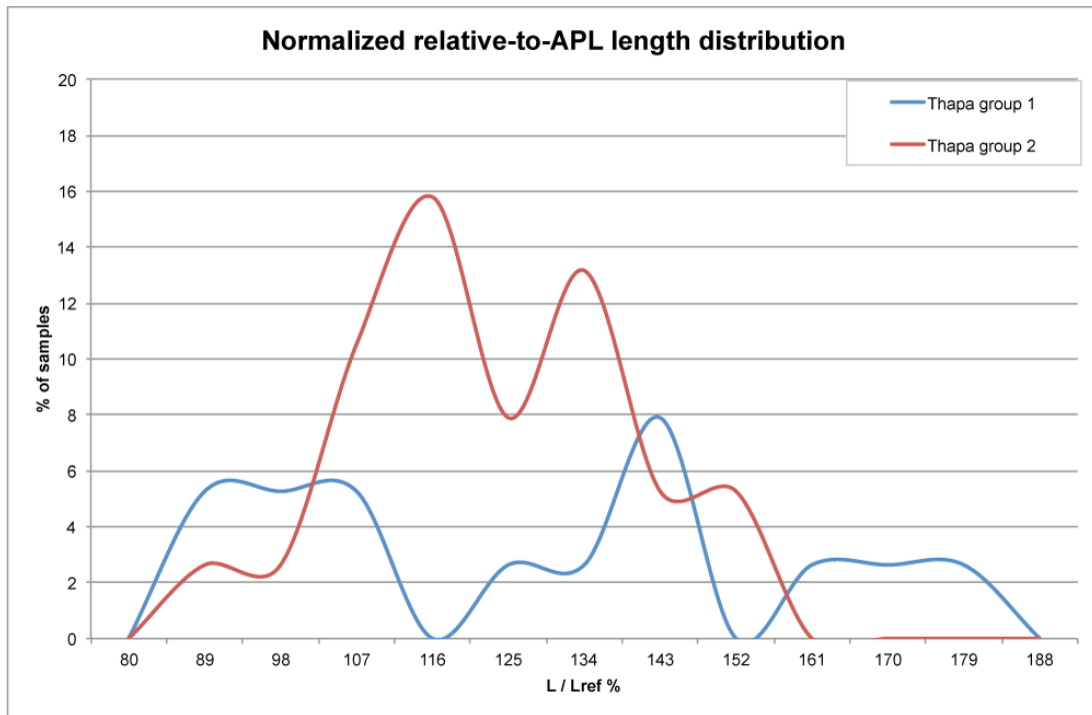


Figure 4.12: Normalized distribution of L_i/L_{ref} across groups for the "Thapa" experiment.

Frequencies	Thapa group #1 RH_i	Thapa group #1 $RH_i\%$	Thapa group #2 RH_i	Thapa group #2 $RH_i\%$
5	0	0	0	0
7.5	0	0	15	39
10	0	0	9	24
12.5	3	8	0	0
15	3	8	0	0
17.5	5	13	0	0
20	1	3	0	0
22.5	1	3	0	0
25	0	0	0	0
27.5	0	0	0	0
30	1	3	0	0
32.5	0	0	0	0

Table 4.3: The normalized distribution of the two experiments

5.1 Introduction

The huge amount of gaze data collected in eye tracking studies requires analysis and visualization in order to help researchers to answer questions posed in every case. The majority of the well-established eye tracking visualization techniques, such as heatmaps, are mainly used for qualitative evaluations.

The purpose of the thesis is to present a new visualization method introduced by Vescoukis et al. [57] for the representation of the average gaze behavior during the observation of stimulus in qualitative way. The approach is based on the TraceBundle algorithm [28] and the aim of this approach is to derive the reference average line (APL).

5.2 Conclusions

The main advantage of the new visualization technique is that it allows the depiction of the average gaze behavior of a population and this approach enables the comparison of the visual reaction of individual subjects in a group through analytical techniques used in similarity analysis. During this thesis, except from the theoretical background a case study was also presented.

The proposed method extends the capability of eye tracking visualizations allowing group comparison reported by specific metrics such as these used in the present study. According to Vescoukis et al. [57] this idea may be applied in domains where eye tracking is already used, to enhance the analysis of eye tracking data. Both a visualization and a framework for quantitative analysis, the APL approach can be useful in cases where the context of eye tracking has reference to lines, paths, etc. that subjects are required or expected to follow. Several other applications may also benefit from the possibility of quantitative analysis of gaze data, such as fields in education, behavioral and intention analysis, psychology and ophthalmology.

5.3 Proposals

After the completion of this thesis some proposals for future work should be noted:

- The approach of the introduced method is based on the generation of a reference average line that we refer as APL. Currently the APL is a flat polyline, but in the future it can be further improved by adding color attributes to the line segments of this polyline,

using calculations such as data density of eye tracking samples near the line, or other metrics.

- Another promising application area might be medical education, where the behavior of medical students in the observation of images where a linear geometry is dominant, such as a e.g. cardiogram, might be an indication of how they perceive attributes of the cardiogram that lead to a diagnosis. Also, deviations from the average or the expert's behavior might be indications that, combined with other data, might be useful in the evaluation (grading) of medical students asked to observe a cardiogram. Needless to say, all of the above possible areas for applying the APL visualization and analytics approach introduced in this paper, need to be individually addressed and specific experiments need to be designed for each case with APL visualization and analysis in mind.
- Fréchet distance (see section 2.6) is a suitable metric for comparison of gaze data. So a suitable experiment or a transformation of the original data that would allow the use of the Fréchet algorithm should be found.
- It would be interesting to see the results of similarity functions to raw gaze data or fixations points. The Hausdorff distance is not a suitable metric since it does not consider the duration of a fixations but Earth Mover's Distance (EMD) between two weighted point sets [12] can be used instead.

Bibliography

- [1] Helmut Alt and Leonidas J Guibas. "Discrete geometric shapes: Matching, interpolation, and approximation". In: *Handbook of computational geometry* 1 (1999), pp. 121–153.
- [2] Gennady Andrienko, Nathaliya Andrienko, Peter Bak, Daniel Keim, Slava Kisilevich, and Stefan Wrobel. "A conceptual framework and taxonomy of techniques for analyzing movement". In: *Journal of Visual Languages & Computing* 22.3 (2011), pp. 213–232.
- [3] Gennady Andrienko, Natalia Andrienko, Christophe Hurter, Salvatore Rinzivillo, and Stefan Wrobel. "From movement tracks through events to places: Extracting and characterizing significant places from mobility data". In: *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*. IEEE, 2011, pp. 161–170.
- [4] Gennady Andrienko, Natalia Andrienko, Michael Burch, and Daniel Weiskopf. "Visual analytics methodology for eye movement studies". In: *Visualization and Computer Graphics, IEEE Transactions on* 18.12 (2012), pp. 2889–2898.
- [5] N. Andrienko, G. Andrienko, N. Pelekis, and S. Spaccapietra. "Basic Concepts of Movement Data". In: Springer Science & Business Media, 2008, pp. 15–38. isbn: 978-3-540-75176-2. doi: 10.1007/978-3-540-75177-9_2. url: http://dx.doi.org/10.1007/978-3-540-75177-9_2.
- [6] Stefan Emilov Atev. "Using asymmetry in the spectral clustering of trajectories". PhD thesis. University of Minnesota, 2011.
- [7] T. Bargiota, V. Mitropoulos, V. Krassanakis, and B. Nakos. "Measuring locations of critical points along cartographic lines". In: *Proceedings of the 26th International Cartographic Conference, Dresden*. talk: 26th International Cartographic Conference ICC, Dresden; 2013-08-25 – 2013-08-30. 2013. isbn: 978-1-907075-06-3.
- [8] Theodora I. Bargiota. "Measuring critical points along cartographic lines using eye movements (in Greek)". MA thesis. National Technical University of Athens, 2013.
- [9] Christoph Berger, Martin Winkels, Alexander Lischke, and Jacqueline Höppner. "GazeAlyze: a MATLAB toolbox for the analysis of eye movement data". In: *Behavior research methods* 44.2 (2012), pp. 404–419.
- [10] T Blascheck, K Kurzhals, M Raschke, M Burch, D Weiskopf, and T Ertl. "State-of-the-art of visualization for eye tracking data". In: *Proceedings of EuroVis*. Vol. 2014. 2014.
- [11] Pierre Bourque, Richard E Fairley, et al. *Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press, 2014.
- [12] Sergio Cabello, Panos Giannopoulos, Christian Knauer, and Günter Rote. *Matching point sets with respect to the earth mover's distance*. Springer, 2005.
- [13] Marco Camilli, Roberto Nacchia, Michela Terenzi, and Francesco Di Nocera. "ASTEF: A simple tool for examining fixations". English. In: *Behavior Research Methods* 40.2 (2008), pp. 373–382. issn: 1554-351X. doi: 10.3758/BRM.40.2.373. url: <http://dx.doi.org/10.3758/BRM.40.2.373>.
- [14] Lei Chen. "Similarity search over time series and trajectory data". PhD thesis. University of Waterloo, 2005.

- [15] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media, 2012.
- [16] Arzu Çöltekin, Simone Garlandini, Benedikt Heil, and SI Fabrikant. "Evaluating the effectiveness of interactive map interface designs: a case study with eye movement analysis". In: *Cartography and GIS* 36 (2008), p. 1.
- [17] Frans W Cornelissen, Enno M Peters, and John Palmer. "The Eyelink Toolbox: eye tracking with MATLAB and the Psychophysics Toolbox". In: *Behavior Research Methods, Instruments, & Computers* 34.4 (2002), pp. 613–617.
- [18] Mark De Berg, Otfried Cheong, Olivier Devillers, Marc Van Kreveld, and Monique Teilaud. "Computing the maximum overlap of two convex polygons under translations". In: *Theory of computing systems* 31.5 (1998), pp. 613–628.
- [19] M-P Dubuisson and Anil K Jain. "A modified Hausdorff distance for object matching". In: *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*. Vol. 1. IEEE, 1994, pp. 566–568.
- [20] A. Fotiou and E. Livieratos. *Geometric Geodesy and Networks (in Greek)*. Ziti, 2000.
- [21] Darren R Gitelman. "ILAB: a program for postexperimental eye movement analysis". In: *Behavior Research Methods, Instruments, & Computers* 34.4 (2002), pp. 605–612.
- [22] Joachim Gudmundsson, Patrick Laube, and Thomas Wolle. "Computational movement analysis". In: *Springer handbook of geographic information*. Springer, 2012, pp. 423–438.
- [23] Joachim Gudmundsson, Patrick Laube, and Thomas Wolle. "Movement Patterns in Spatio-temporal Data". In: *Encyclopedia of GIS*. Springer, 2008, pp. 726–732.
- [24] Michiel Hagedoorn. "Pattern matching using similarity measures". PhD thesis. Universiteit Utrecht, 2010.
- [25] John Heminghous and Andrew T. Duchowski. "iComp: A Tool for Scanpath Visualization and Comparison". In: *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*. APGV '06. Boston, Massachusetts, USA: ACM, 2006, pp. 152–152. isbn: 1-59593-429-4. doi: 10.1145/1140491.1140529. url: <http://doi.acm.org/10.1145/1140491.1140529>.
- [26] Ian D Jonsen, Joanna Mills Flemming, and Ransom A Myers. "Robust state-space modeling of animal movement data". In: *Ecology* 86.11 (2005), pp. 2874–2880.
- [27] Sophia Karagiorgou, Vassilios Krassanakis, Vassilios Vescoukis, and Byron Nakos. "Experimenting with polylines on the visualization of eye tracking data from observations of cartographic lines". In: *2nd International Workshop on Eye Tracking for Spatial Research, co-located with the 8th International Conference on Geographic Information Science (GIScience 2014), Vienna, Austria*. 2014, pp. 22–26.
- [28] Sophia L Karagiorgou. "Inference of transportation networks from sparse tracking data". PhD thesis. National Technical University of Athens, 2014.
- [29] Vassilios Krassanakis. "Development of a methodology of eye movement analysis for the study of visual perception in animated maps (in Greek)". PhD thesis. National Technical University of Athens, 2014.
- [30] Vassilios Krassanakis, Vassiliki Filippakopoulou, and Byron Nakos. "An Application of Eye Tracking Methodology in Cartographic Research". In: *Proceedings of the EyeTrack-Behavior* (2011).
- [31] Vassilios Krassanakis, Vassiliki Filippakopoulou, and Byron Nakos. "EyeMMV toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification". In: *Journal of Eye Movement Research* 7.1 (2014), pp. 1–1.
- [32] Perttu Laurinen, Pekka Siirtola, and Juha Röning. "Efficient Algorithm for Calculating Similarity between Trajectories Containing an Increasing Dimension." In: *Artificial Intelligence and Applications*. 2006, pp. 392–399.

- [33] Alexandra K. Lelli. "Investigating dynamic variables with eye movement analysis on visual scenes (in Greek)". MA thesis. National Technical University of Athens.
- [34] Dongheng Li, Jason Babcock, and Derrick J Parkhurst. "openEyes: a low-cost head-mounted eye-tracking solution". In: *Proceedings of the 2006 symposium on Eye tracking research & applications*. ACM. 2006, pp. 95–100.
- [35] Ismini-Eleni F Lokka. "Investigating dynamic variables with eye movement analysis on topographic map (in Greek)". MA thesis. National Technical University of Athens, 2014.
- [36] Paraskevi A. Lotidi. "Study of the dynamic variable rate of change by using eye-tracking methods in maps of varying intensity (in Greek)". MA thesis. National Technical University of Athens, 2014.
- [37] MATLAB. *version 8.3.0 (R2014a)*. Natick, Massachusetts: The MathWorks Inc., 2014.
- [38] R. Matos. *Designing eye tracking experiments to measure behavior*. Presentation in Post Conference Course, EyeTrackBehavior Conference, Tobii Technology, Frankfurt. 2011.
- [39] Facundo Mémoli. "On the use of Gromov-Hausdorff distances for shape comparison". In: (2007).
- [40] Facundo Mémoli and Guillermo Sapiro. "A theoretical and computational framework for isometry invariant recognition of point cloud data". In: *Foundations of Computational Mathematics* 5.3 (2005), pp. 313–347.
- [41] Movebank. *Movebank*. <https://www.movebank.org/>. [Online; accessed 25-June-2015]. 2015. url: <https://www.movebank.org/>.
- [42] Nikos Pelekis and Yannis Theodoridis. "Mobility Data Management and Exploration". In: Springer, 2014. Chap. 6.
- [43] Alex Poole and Linden J. Ball. "Eye Tracking in Human-Computer Interaction and Usability Research: Current Status and Future". In: *Prospects*, Chapter in C. Ghaoui (Ed.): *Encyclopedia of Human-Computer Interaction*. Pennsylvania: Idea Group, Inc. 2005.
- [44] Dario D Salvucci and Joseph H Goldberg. "Identifying fixations and saccades in eye-tracking protocols". In: *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM. 2000, pp. 71–78.
- [45] Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin Hansen. "Evaluation of a low-cost open-source gaze tracker". In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. ACM. 2010, pp. 77–80.
- [46] Swaminathan Sankararaman, Pankaj K. Agarwal, Thomas Mølhave, and Arnold P. Boedihardjo. "Computing Similarity between a Pair of Trajectories". In: *CoRR* abs/1303.1585 (2013). url: <http://arxiv.org/abs/1303.1585>.
- [47] Ludmila Scharf. "Computing the Hausdorff distance between sets of curves". PhD thesis. Thesis, Freie Universität, Berlin, 2003.
- [48] Michael D Shapiro and Matthew B Blaschko. "On hausdorff distance measures". In: *Computer Vision Laboratory University of Massachusetts Amherst, MA 1003* (2004).
- [49] Frederick Shic, Brian Scassellati, and Katarzyna Chawarska. "The incomplete fixation measure". In: *Proceedings of the 2008 symposium on Eye tracking research & applications*. ACM. 2008, pp. 111–114.
- [50] Hiroyuki Sogo. "GazeParser: an open-source and multiplatform library for low-cost eye tracking and analysis". In: *Behavior research methods* 45.3 (2013), pp. 684–695.
- [51] O Špakov and Darius Miniotas. "Visualization of eye gaze data using heat maps". In: *Elektronika ir Elektrotechnika* 74.2 (2015), pp. 55–58.
- [52] strategyone.com. *strategyone*. <http://www.strategyone.com/strategyonehomepageazeplot.html>. 2015. url: <http://www.strategyone.com/strategyonehomepageazeplot.html>.

- [53] K Thapa. "Critical points detection and automatic line generalisation in raster data using zero-crossings". In: *The Cartographic Journal* 25.1 (1988), pp. 58–68.
- [54] Remco C Veltkamp. "Shape matching: Similarity measures and algorithms". In: *Shape Modeling and Applications, SMI 2001 International Conference on*. IEEE. 2001, pp. 188–197.
- [55] Remco C Veltkamp and Michiel Hagedoorn. "Shape similarity measures, properties and constructions". In: *Advances in visual information systems*. Springer, 2000, pp. 467–476.
- [56] Suresh Venkatasubramanian. "Geometric shape matching and drug design". PhD thesis. Stanford University, 1999.
- [57] V. Vescoukis, S. Karagiorgou, V. Anagnostopoulos, and V. Krassanakis. "Introducing APLs - average polylines for visualisation and analysis of eye-tracking data". Manuscript submitted for publication. 2015.
- [58] Adrian VoßKühler, Volkhard Nordmeier, Lars Kuchinke, and Arthur M Jacobs. "OGAMA (Open Gaze and Mouse Analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs". In: *Behavior Research Methods* 40.4 (2008), pp. 1150–1162.
- [59] Haozhou Wang, Han Su, Kai Zheng, Shazia Sadiq, and Xiaofang Zhou. "An effectiveness study on trajectory similarity measures". In: *Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137*. Australian Computer Society, Inc. 2013, pp. 13–22.
- [60] Julia M. West, Anne R. Haake, Evelyn P. Rozanski, and Keith S. Karn. "eyePatterns: Software for Identifying Patterns and Similarities Across Fixation Sequences". In: *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*. ETRA '06. San Diego, California: ACM, 2006, pp. 149–154. isbn: 1-59593-305-0. doi: 10.1145/1117309.1117360. url: <http://doi.acm.org/10.1145/1117309.1117360>.
- [61] Wikipedia. *Eye tracking* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Eye_tracking. [Online; accessed 26-June-2015]. 2015. url: https://en.wikipedia.org/wiki/Eye_tracking.
- [62] Wikipedia. *Fréchet distance* — *Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/wiki/Fr%C3%A9chet_distance. [Online; accessed 30-May-2015]. 2015. url: [http://en.wikipedia.org/wiki/Fr\C3\%A9chet_distance](http://en.wikipedia.org/wiki/Fr%C3%A9chet_distance).
- [63] Wikipedia. *Geographic coordinate system* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Geographic_coordinate_system. [Online; accessed 24-June-2015]. 2015. url: https://en.wikipedia.org/wiki/Geographic_coordinate_system.
- [64] Wikipedia. *GPS wildlife tracking* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/GPS_wildlife_tracking. [Online; accessed 24-June-2015]. 2015. url: https://en.wikipedia.org/wiki/GPS_wildlife_tracking.
- [65] Wikipedia. *MATLAB* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/MATLAB>. [Online; accessed 27-June-2015]. 2015. url: <https://en.wikipedia.org/wiki/MATLAB>.
- [66] Wikipedia. *Pattern* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/Pattern>. [Online; accessed 4-July-2015]. 2015. url: <https://en.wikipedia.org/wiki/Pattern>.
- [67] Wikipedia. *Point of interest* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Point_of_interest. [Online; accessed 29-June-2015]. 2015. url: https://en.wikipedia.org/wiki/Point_of_interest.
- [68] Wikipedia. *Requirements analysis* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Requirements_analysis. [Online; accessed 30-May-2015]. 2015. url: https://en.wikipedia.org/wiki/Requirements_analysis.
- [69] Wikipedia. *Saccade* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/Saccade>. [Online; accessed 26-June-2015]. 2015. url: <https://en.wikipedia.org/wiki/Saccade>.

- [70] Wikipedia. *Trajectory* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/wiki/Trajectory>. [Online; accessed 24-June-2015]. 2015. url: <https://en.wikipedia.org/wiki/Trajectory>.
- [71] Yu Zheng and Xiaofang Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.
- [72] Oleg Špakov. *ETU-Driver*. <http://www.sis.uta.fi/~csolsp/downloads.php?id=ETUDriver>. [Online; accessed 5-July-2015]. 2015. url: <http://www.sis.uta.fi/~csolsp/downloads.php?id=ETUDriver>.

APPENDIX A

Peristera stimulus data

Raw file	Cluster range	Angle tolerance	Num. of samples	Num. of users	Minimum linkage	Bounding box	Length	Hausdorff Distance	Modified Hausdorff Distance
Peristera50K01	20	10	5	1	1	50	2,072.93	355.28	40.17
Peristera50K010	25	20	5	1	1	50	1,977.04	362.29	30.95
Peristera50K011	15	15	6	1	1	60	2,100.64	419.04	37.07
Peristera50K012	20	10	8	1	1	50	1,773.33	378.64	39.25
Peristera50K015	20	10	8	1	1	50	1,823.30	439.40	41.85
Peristera50K017	20	20	5	1	1	50	2,598.18	340.74	30.56
Peristera50K018	25	15	6	1	1	60	1,929.37	412.72	32.67
Peristera50K019	30	24	7	1	1	50	1,483.20	423.51	50.18
Peristera50K02	12	10	2	1	1	70	2,838.31	370.38	33.83
Peristera50K021	15	16	4	1	1	50	1,969.11	389.08	28.87
Peristera50K022	15	10	5	1	1	50	2,044.26	392.82	29.52
continues on the next page									

Raw file	Cluster range	Angle tolerance	Num. of samples	Num. of users	Minimum linkage	Bounding box	Length	Hausdorff Distance	Modified Hausdorff Distance
Peristera50K024	25	10	4	1	1	70	1,880.32	373.83	39.43
Peristera50K025	20	10	4	1	1	50	1,930.01	440.10	47.96
Peristera50K027	25	15	3	1	1	60	1,912.45	445.76	51.46
Peristera50K028	20	15	1	1	1	60	2,613.38	159.11	35.46
Peristera50K029	15	20	1	1	1	50	2,440.88	402.93	34.70
Peristera50K03	17	5	7	1	1	70	2,101.34	376.29	29.23
Peristera50K030	15	20	4	1	1	50	2,051.11	368.49	29.07
Peristera50K032	15	10	5	1	1	50	1,526.11	369.67	33.04
Peristera50K033	15	15	5	1	1	50	1,825.40	429.52	35.04
Peristera50K034	22	10	3	1	1	50	1,707.33	389.65	32.43
Peristera50K035	10	10	2	1	1	50	1,934.55	391.05	29.22
Peristera50K036	30	30	3	1	1	50	2,279.82	200.14	27.69
Peristera50K038	20	15	2	1	1	50	2,416.00	395.32	40.12
Peristera50K039	15	10	5	1	1	50	2,012.49	396.14	28.91
Peristera50K04	10	10	5	1	1	60	1,871.40	355.35	26.74
Peristera50K040	14	10	6	1	1	50	1,869.96	355.84	33.35
Peristera50K041	20	15	6	1	1	50	2,762.20	368.69	29.60
Peristera50K042	10	10	3	1	1	50	1,931.51	358.94	50.54
Peristera50K043	17	15	7	1	1	50	1,776.23	427.05	36.21
Peristera50K044	20	10	5	1	1	50	2,238.23	373.17	32.30
Peristera50K045	15	10	5	1	1	50	2,212.17	409.48	30.61
Peristera50K046	10	10	2	1	1	50	1,861.04	353.62	41.35
Peristera50K047	15	10	5	1	1	50	1,927.98	383.44	26.33
Peristera50K048	20	10	4	1	1	50	1,945.59	378.09	30.97
Peristera50K049	20	15	4	1	1	50	1,849.31	391.19	29.07
Peristera50K05	15	10	4	1	1	80	2,558.89	434.36	37.94
Peristera50K052	10	5	5	1	1	50	2,025.05	109.46	18.10
Peristera50K054	30	8	8	1	1	30	1,847.03	335.41	33.54

continues on the next page

Raw file	Cluster range	Angle tolerance	Num. of samples	Num. of users	Minimum linkage	Bounding box	Length	Hausdorff Distance	Modified Hausdorff Distance
Peristera50K07	27	20	5	1	1	30	1,600.98	421.88	41.15

Table A.1: The data of Peristera stimulus

APPENDIX B

Thapa stimulus data

Raw file	Group	Cluster range	Angle tolerance	Num. of samples	Num. of users	Minimum linkage	Bounding box	Length	Hausdorff Distance	Modified Hausdorff Distance
Thapa01	2	20	20	3	1	1	50	1,569.30	126.6795	45.3346
Thapa010	1	20	20	3	1	1	50	1,325.05	221.5554	52.5078
Thapa011	1	35	15	5	1	1	100	1,844.81	157.3236	52.4924
Thapa012	1	28	15	5	1	1	80	1,208.66	194.5716	52.7633
Thapa015	1	40	15	3	1	1	50	2,424.26	307.6057	45.8666
Thapa017	2	30	20	5	1	1	50	1,480.72	140.0413	38.0124
Thapa018	1	20	15	4	1	1	50	1,248.80	141.3833	49.2364
Thapa019	2	25	20	5	1	1	20	2,068.51	95.7864	38.4021
Thapa02	2	30	15	3	1	1	50	1,862.98	83.1184	37.5432
Thapa021	2	25	15	5	1	1	50	1,963.16	94.9938	35.5297
Thapa022	2	20	15	5	1	1	50	1,459.63	119.6408	36.8787
Thapa024	2	20	15	4	1	1	50	1,846.49	85.8959	34.3455

continues on the next page

Raw file	Group	Cluster range	Angle tolerance	Num. of samples	Num. of users	Minimum linkage	Bounding box	Length	Hausdorff Distance	Modified Hausdorff Distance
Thapa025	2	20	15	4	1	1	50	1,657.25	131.3555	46.1202
Thapa027	1	15	10	5	1	1	50	1,687.97	238.182	58.8283
Thapa029	1	20	10	1	1	1	30	1,939.52	163.9836	53.7433
Thapa03	1	20	10	2	1	1	30	1,988.26	179.6805	40.4697
Thapa030	2	30	40	4	1	1	50	1,586.66	117.3248	68.7824
Thapa032	2	40	30	4	1	1	50	1,456.26	84.38	14.9827
Thapa033	2	30	20	5	1	1	50	1,656.61	96.6059	34.9138
Thapa034	2	20	20	4	1	1	50	1,520.70	120.1628	46.8895
Thapa035	2	20	15	4	1	1	50	1,818.77	71.0811	34.1994
Thapa036	2	25	15	3	1	1	50	2,092.01	79.0998	33.2847
Thapa038	2	20	15	3	1	1	50	1,992.51	77.2737	31.6131
Thapa039	2	20	15	4	1	1	50	1,597.10	109.8914	36.8604
Thapa04	1	15	30	2	1	1	50	2,219.84	259.906	45.5792
Thapa040	1	10	20	4	1	1	30	1,189.35	214.4671	78.0874
Thapa042	1	20	30	4	1	1	70	1,902.62	201.9069	52.9746
Thapa043	2	20	20	4	1	1	50	1,784.23	137.7386	34.5301
Thapa044	2	20	20	4	1	1	50	1,481.83	131.5853	50.4244
Thapa045	2	30	20	4	1	1	50	1,244.53	79.5988	34.6919
Thapa046	1	20	20	4	1	1	50	1,442.55	220.4169	56.5583
Thapa047	1	20	40	3	1	1	50	2,288.80	389.7367	40.6293
Thapa048	2	20	30	2	1	1	50	1,682.90	95.2711	33.0134
Thapa049	2	45	30	6	1	1	50	1,341.65	91.185	42.01
Thapa05	2	20	10	5	1	1	50	1,588.30	101.7074	45.1123
Thapa052	1	15	10	5	1	1	50	1,436.66	242.5805	47.6775
Thapa054	2	20	15	4	1	1	50	1,835.19	99.8973	38.9004
Thapa07	2	20	15	4	1	1	50	1,603.67	76.717	35.4925

Table B.1: The data of Thapa stimulus

APPENDIX C

Submitted paper

Introducing APLs - average polylines for visualisation and analysis of eye-tracking data

V.Vescoukis¹, S.Karagiorgou², V.Anagnostopoulos¹ and V.Krassanakis¹

Corresponding author: Vassilios Vescoukis, v.vescoukis@cs.ntua.gr

(1) National Technical University of Athens, Department of Surveying Engineering

(2) Athena-Research and Innovation Center in Information, Communication and Knowledge Technologies

Abstract. Several visualization methods for eye tracking data exist to help researchers from many disciplines depict data collected in eye tracking experiments. Expanding previous work on visualizing eye tracking data from observations of cartographic lines, in this paper we introduce APLs, as a new visualization and analysis tool of eye tracking data using polylines inferred from the analysis of samples. An APL corresponds to the “average” line that is actually seen by subjects, which can be useful in the study of various optical representation concepts, such as the assessment of the effects of alternative cartographic line attributes, distractions, abstraction levels and more. Furthermore, APLs enable the quantitative analysis of the individuals' behavior compared to the average behavior of members of the same group in an eye-tracking experiment, using established similarity measures such as the Hausdorff distance or other metrics. This approach can enable many new applications in education, behavioral and intention analysis, psychology, and elsewhere.

1 Introduction

The recording and the analysis of eye movements constitute an effective method for the exploration of several aspects related to visual behavior. Eye tracking provides objective and quantitative evidences towards the examination of visual attention (Duchowski, 2002) and has become very popular in a wide range of scientific disciplines (see also Duchowski, 2002; Richardson, 2004). Eye movement analysis techniques are based on the computation of fundamental and derived metrics produced by the recorded eye tracking protocols (Goldberg & Kotval, 1998; Jacob & Karn, 2003; Poole & Ball, 2005; Ehmke & Wilson, 2007; Krassanakis, Filippakopoulou, & Nakos, 2014) while the visualization of eye tracking data allows the analysis through a qualitative approach (Blascheck, Kurzhals, Raschke, Burch, Weiskopf, & Ertl, 2014).

Visualization techniques may be summarized into two main categories including visualizations referred to each individual subject or all subjects which participate in an experimental process. Obviously, the selection of the suitable methods depends on the research question of each study. In any case, a first approach includes the visualization of human gazes as simple trajectories produced by the recorded gaze points (Špakov & Miniotas, 2007). In the most of the cases this technique is considered to be insufficient taking into account the amount of the collected data. Hence, the implementation of clustering techniques towards the identification of fixation and saccade events (see also Salvucci & Goldberg, 2000; Duchowski, 2007; Holmqvist, Nyström, Andersson, Dewhurst, Jarodzka, & Van de Weijer, 2011) contributes to a more effective way for the production of eye tracking visualizations. Scan path visualization is a common technique where fixation events are presented as circles, which radical values are connected with their absolute or relative durations, while saccades are depicted as the line segments between fixations. Additionally, the third dimension may also be used in order to solve the problem of data overlapping (Räihä, Aula, Majaranta, Rantala, & Koivunen, 2005).

One of the most popular techniques to visualize eye tracking data, referred to the gaze behavior of either individuals or all subjects of an experiment, is heatmap visualization. Heatmaps are two-dimensional representations where the values of a variable are presented using colors (Bojko, 2009) or different intensities of a color hue. In eye tracking, heatmap visualization may be produced using as variables the duration, the relative duration and the number of fixations or the participant percentage who fixated on the areas of the stimulus (Bojko, 2009). Additionally, a recent study (Krassanakis et al., 2014) proposes the generation of heatmap based on the density of gaze data recordings. Furthermore, dotplots are also two-dimensional graphical techniques, which are used for the visualization of sequences produced by modeled scan paths (Goldberg & Helfman, 2009). Extending the typical representations in two dimensions, Li, Çöltiken, & Kraak (2010) suggest also the exploration of eye tracking data using the dynamic environment of a Space-Time-Cube (STC).

As mentioned above, the majority of the aforementioned visualization techniques may be applied to depict eye-tracking data produced by one subject or to model the total visual behavior of all subjects of an experiment. In many studies, the visualization of an average gaze behavior is also considered very important as it may reveals a representative image of subjects' visual reaction. Specifically, this approach is very critical in cases that the

reconstruction of the gaze route history is meaningful for the examination of a research question. For instance, Bargiota, Mitropoulos, Krassanakis, and Nakos (2013) performed an experimental study in order to examine subjects' eye movements during the observation of cartographic lines. In this study, the reconstruction of gaze route history allows the comparison of the eye measurements with the distribution of cartographic lines' critical points, which are used for the performance of cartographic generalization methods.

In this paper, we present further development on earlier work on the depiction of the gaze route history using polylines (Karagiorgou et al, 2014). We present a workflow supported by a toolkit for the extraction and analysis of what is now called an APL - average polyline from eye-tracking data. The description of the APL extraction algorithm is done in section 2. Section 3 discusses the workflow for the extraction, similarity assessment and analysis of APLs from eye tracking data, along with a Matlab toolkit that has been developed to support the process. Application of this workflow in two case studies and demonstration of the similarity metrics and analysis is discussed in Section 4. Finally Section 5 concludes by discussing on current status and ideas for further applications of the introduced approach for visualization and analysis of eye tracking data in other domains.

2 Inference of APLs - average polylines

In earlier work (Karagiorgou, Krassanakis, Vescoukis, & Nakos 2014), we have experimented with a first version of an algorithm to derive APLs from eye tracking data. That particular algorithm had been originally developed in a different research context where the objective was the extraction of road networks from sparse tracking data. Building on that experience, we have applied the algorithm to two larger data sets from eye tracking experiments and did some fine tuning, especially regarding the parameters, as the algorithm was initially intended to receive parameter values that map to the physical space instead of the computer screen space.

The algorithm involves three steps;

- (i) identifying hubs,
- (ii) connecting hubs, and

(iii) reducing the links to generate the APL,

These steps will be discussed in more detail in the sequel.

Step 1: Hubs and spatial fixation. A hub represents the spatial fixation that the eye creates near an area of interest. Indicators for hub recognition are the number of tracking samples, the number of different users and the coverage of an extended area of focus. The algorithm takes as input the eye tracking data and determines the k -Nearest Neighbors (k -NN) of each sample, which are subsequently filtered according to the number of users. On these filtered samples, we apply the DBSCAN clustering algorithm using a distance threshold and a minimum number of samples, depending on the specifics of the experiment. The centroids of the resulting clusters are what we call hubs. Hubs are very similar to fixations. Hence, they can also be calculated using already established fixation identification algorithms, such as dispersion-based (I-DT) ones (see also Salvucci & Goldberg, 2000).

Step 2: Connecting hubs. Next, we connect hubs by links. A fringe benefit of the hubs computation based on spatial fixation is that for all data we know which samples helped in identifying hubs. To derive links we exploit this knowledge: for each hub we record the outgoing and/or incoming tracking portions connecting this hub to others by scanning all eye tracking data to discover sequences of hubs. The result of this step is the creation of a sample polyline set that connects hubs with links, which contains many links that need to be further simplified in the sequel. In our representation of eye tracking data, all tracking samples that are also hubs are marked as such. Hence, performing a linear scan of all tracking data reveals the respective tracking portions that connect hubs.

Essentially, two hubs in question will typically be connected by a number of eye tracks, i.e., tracks from more than one subjects from one hub to the other. In terms of network geometry, at this stage of the overall link inference process, we introduce redundant links between hubs as we simply identify how eye tracks connect hubs. Merging these links will be the next step. We refer to track portions connecting hubs at this stage as link samples. This process is shown in Figure 1.

Phase 2: Connecting Hubs

Input: Set of eye-tracks T and hubs H

Output: A set of link samples L_S

```
begin
  // Connecting hubs using track data
   $H_S \leftarrow \emptyset$  // Hub sequence
   $L_S \leftarrow \emptyset$  // Link samples
  // Identify hub sequences from tracks
  foreach  $t \in T$  do
    foreach  $p \in t$  do
      // position sample is mapped to a hub
      if  $p \in H$  then
        // record prev., current hub
         $H_S \leftarrow \{h^-, h, p^-, p, t\}$ 
      end
    end
  end
  // Collect and merge link samples
  foreach  $\{h^-, h\} \in H_S$  do
    // add all track portions for this hub pair
    foreach  $t \in H_S$  do
       $L_S \leftarrow \{t, p^-, \dots, p\}$ 
    end
    // cluster link samples
     $Width \leftarrow Width(L_S)$ 
     $Weight \leftarrow Weight(L_S)$ 
     $L_S \leftarrow SweepMerge(L_S)$ 
  end
end
```

Figure 1. Algorithm for initial extraction of links from hubs

Step 3: Compacting links. To this point, we have hubs connected by links derived from eye tracking data that exhibit spatial fixation at these hubs. In a nutshell, the algorithm identifies tracking portions that are close to existing links by means of a buffer region and merges their geometry into the existing link geometry. The size of the buffer region depends on the specifics of the data; in our case we used 15 pixels as buffer region, however this may vary depending on the specifics of the experiment. In this step, we neither introduce new hubs nor do we add new links. We only adjust the geometry of existing links using a three-step algorithm: (i) sort existing link samples, (ii) determine relevant tracking portions using a buffer region around link samples, and (iii) adjust the geometry of links based on the tracking data geometry.

The first step is to sort all links according to their length so as to process longer links first as they are more significant for link construction. I.e., the longer a link, the more selective will be the match for a longer trajectory portion to fit in a bounding box. In trying to identify portions of link samples that match other link samples expressed by spatial proximity and direction similarity, the algorithm uses a bounding box around the examined link sample and retrieves all intersecting portions of other links. The size of the bounding box is determined by the width of the respective link sample. In addition to containment in a bounding box, a threshold is used to assess direction similarity. In our experimentation, an adequate measure for direction similarity is equals to 45° . Figure 2 shows in black the bounding box of an examined link. The examined link is shown in grey and respective portions of other candidate links are shown in light grey. The algorithm for compacting links is shown in Figure 3.

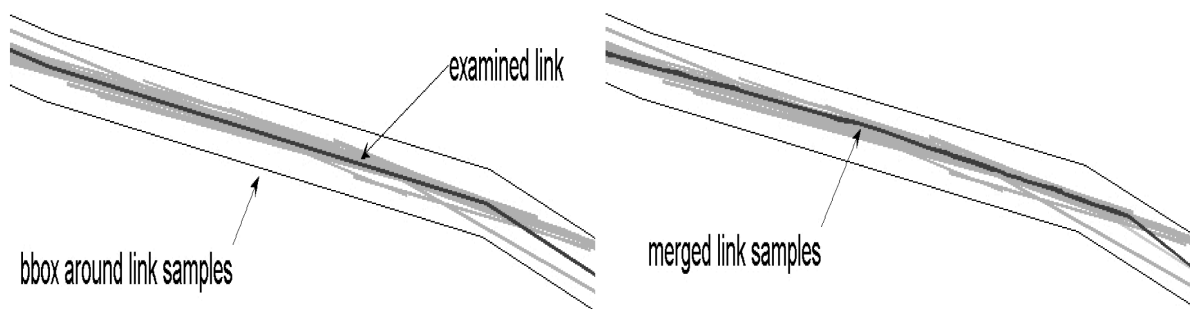


Figure 2. Compacting links

As a pre-cursor to merging link samples, we record for each examined link its similar links and the portions that exhibit similarity. The latter is important in order to manage partially similar link samples. As the similar link samples can be located at the beginning, the end, or the middle, the remaining portions are preserved by splitting the respective links. The method is applied to every portion of the examined link that exhibits partial similarity to other links. New links are created by interpolating link samples and introducing intersection nodes. In addition, new links preserve a weight that is the sum of the weights of the merged links. Link samples are updated several times during this stage. While the examined links are reconstructed, new link samples are created and the existing are removed, i.e., additions, deletions and updates to the connectivity of the APL.

Phase 3: APL extraction

Input: A set of link samples LS

Output: A set of links L

begin

LS \leftarrow sort(LS, length) // Sorting link samples by length

CLS \leftarrow \emptyset // Candidate link samples

Width // width of link samples

Angle // direction threshold

foreach l \in LS **do**

 CLS \leftarrow Find(bbox(l, Width(l), Angle))

foreach cl \in CLS **do**

if Contains(l, cl) **then**

 L \leftarrow SweepMerge(l, cl)

end

else

 // partial overlap

 cl_{in}, cl_{out} \leftarrow Split(l, cl)

 CLS.add(clout)

 L \leftarrow SweepMerge(l, clin)

end

end

end

end

Figure 3. Algorithm for compacting links to form the final APL

3 Analysis of samples/APL

3.1 Concepts and workflow

The algorithms discussed so far are used to extract a set of hubs and connecting links from eye tracking data from many subjects who have been asked to follow a linear entity such as a route in a road map or a cartographic line. We call this final set of links the "Average Polyline" (APL). APLs represent the line that the subjects have followed "on average". Depending on the purpose of the experiment, an APL can be used to help researchers understand how visualizations, distractions or even subjects' characteristics affect what has been seen in such an experiment.

To do this, after the initial APL has been extracted, we need to further process the whole set of eye tracking data to analyze the behavior of each individual and compare it to the "average behavior" of the whole sample. Assuming that the experiments have been designed with this

in mind, two types of questions may be answered by this approach:

- **Visualization attributes:** How does a different visualization affect the "average" behavior of the same population?
- **Individuals' behavior:** How do groups of individual subjects with similar selected characteristics deviate from the "average" behavior in the same experiment?

There are several potential applications of each of the above type of questions. In the former case, APLs provide a new means for visualizing eye-tracking data, which can be used along heatmaps and other methods to extract useful conclusions, depending on the specifics of the experiment. The latter type of questions is more challenging: by being able to compare the behavior of each individual to the average behavior of the population participating in an experiment, we can perform quantitative statistical analysis on groups of subjects with common attributes. This has several potential applications in cognitive science, ophthalmology, education, psychology and elsewhere, which remain to be explored as will be further discussed in Section 5.

The workflow to process a set of eye tracking data to perform the analysis discussed above is shown in Figure 4.

APL analysis workflow

Input: Set of eye-tracking data for a given experiment E

Output: APL, table T containing set of individuals' PL (PL_i), deviation metrics (D_i)

begin

 adjust the APL extraction algorithm parameters

 calculate the APL for E using the APL extraction algorithm

foreach S_i **in** E

 calculate PL_i by applying the APL extraction algorithm on data in S_i

 calculate deviation metrics D_i between PL_i and APL

 add PL_i, D_i to table T

end

 perform statistical analysis on table T

.end

Figure 4. A workflow for creating and analyzing APLs

In the sequel we present a Matlab toolkit that has been developed to support the workflow presented in Figure 4.

3.2 The APL Toolkit

The purpose of the toolkit is to support the researchers in productively analyzing gaze data and producing the output of the suggested workflow (APL and T) needed to further analyze how specific subject attributes affect the gaze of the subject. In Figure 5 the structure of table T is shown. That can be achieved by filtering the results according to a specific attribute and comparing it afterwards with another attribute or with the whole subject population. The significance of each attribute is shown in histograms. From the differences in the histograms, someone will be able to extract the significance of the attribute in the gaze of the subject.

Structure of table T (columns)

Experiment ID
Subject ID (i)
Subject classification Ci
PLi calculation parameters
PLi length
Absolute Dik
Normalized Dik

Figure 5. Structure of table T to be used in analyzing APLs and individuals' behavior

In the above (i) denotes the subject ID and k denotes the deviation metric, assuming that more than one deviation metrics will be calculated. In the case study to be discussed in the sequel, we have used two such metrics, namely the Hausdorff distance H and the relative to the APL length of the subject's PLi.

Through the GUI the user is able to select the files that contain the raw data of the experiment and calculate the fixations (based on Krassanakis et al. (2014) fixation detection algorithm), the hubs and the APL. By using the toolkit interface the user is able to calculate the APL for the population he chooses and visualize the results; if the all the raw samples from the experiment are selected, then the APL that will be used as reference in the next steps, will be extracted; if one single or any subset of samples is selected, then the PLi or the APL for that specific subset will be extracted, respectively. A screenshot of the APL toolkit is shown in Figure 6.

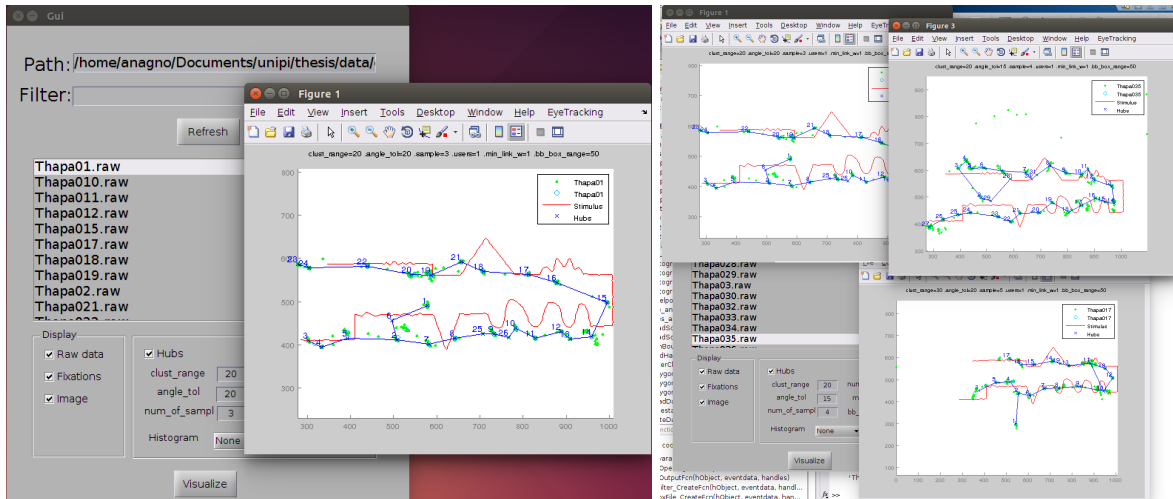


Figure 6. The APL analysis toolkit

By combining the results and using a suitable metric, such as the Hausdorff distance as will be discussed below, the user is able to calculate the differences between the two polylines and extract the histogram of the differences between each PL_i and the APL to perform further analyses.

3.3 Similarity analysis

Object similarity (referred also as object matching or object dissimilarity or shape matching) is the decision on the similarity between two objects. In general, a set of objects A, B is given and the resemblance to each other is quantitatively expressed as a distance between the two objects (Scharf, 2003 and Veltkamp, 2001). The problem of object similarity arises in a variety of applications, including computer graphics, cartography and pattern recognition. Object similarity relies on metrics that describe the degree of difference between the two objects. A shape similarity measure on a collection of shapes S is a nonnegative value $d: S \times S \rightarrow \mathfrak{R}$ defined on pairs of patterns indicating the degree of resemblance of these patterns (Hagedoorn, 2000, Scharf, 2003 and Shapiro et al., 2004). According to Veltkamp (2013) some of the desirable properties that a similarity measure should have are a) metric properties (i.e. non negativity, the identity property and a strong triangle inequality) b) continuity properties and c) invariance.

In this work we are interested only in the first property since we are comparing sets of points

that constitute polylines. For this reason the Hausdorff distance that is widely used as metric in similar cases (Sofia Karagiorgou PhD), has been selected. The Hausdorff distance is the most studied similarity measure in computational geometry (Scharf, 2003 and Alt et al, 1999). It is defined for an arbitrary non-empty and closed set A and B in equation (1).

$$d_H(A, B) = \max \left\{ \sup_{x \in A} \inf_{y \in B} d(x, y), \sup_{y \in B} \inf_{x \in A} d(x, y) \right\} \quad (1)$$

where $d(x, y), x, y \in \mathbb{R}^2$ denotes the Euclidean distance between x and y . The closer to zero, the more the shapes resemble to each other. In this work we use the Euclidean distance but other norms can also be considered in the future, depending on the nature of the experiment, the research question, or the application.

The main disadvantage of the Hausdorff distance is that it is sensitive to noise (Scharf,2003 and Veltkamp, 2013). According to (Dubuisson et all, 1994) “the distance values are large even in the presence of small amount of noise and this is due to the fact the Hausdorff distance value is set by the maximum distance among the two points sets”. To solve this problem, Dubuisson et all (1994) proposed a non-metric modified version of the Hausdorff distance, which satisfies positivity and symmetry, but does not satisfy the triangle inequality. The same work determined that among the class of distance measures based on Hausdorff distance, the modified version is best for matching a set of objects based on their edge points and it has two desirable properties: a) its value increases monotonically as the amount of differences between the two sets increases, and b) it is robust to outliers points.

Our experimentation so far has shown that although the absolute values of the Hausdorff and the modified Hausdorff distances are very different, the distribution of the deviations is not affected. Moreover, what is considered as noise by by Dubuisson et all (1994) may not actually be noise in the case of eye-tracking samples, as it may represent unwanted behavior caused by visual distractions or other anomalies, depending on the experiment. A further discussion on this will follow in section 5.

4 Case studies

Eye tracking datasets used for the testing of the proposed approach have been collected in one previous experimental study (Bargiota et al., 2013) where eye tracking data were collected during the observation of two cartographic lines. In the sequel, these experiments will be

referred to as "Peristera" and "Thapa" (first and the second line correspondingly). Subjects' eye movements were recorded with a sampling frequency of 60Hz. Totally 40 subjects participated in the experimental process in both cases. The experimental conditions are fully described by Bargiota et al. (2013) while more information about the eye tracking equipment can be reached through a previous study (Krassanakis, Filippakopoulou, & Nakos, 2011). Peristera consists a real cartographic line that depicts the coastline of Peristera's Island at scale 1:50.000 approx. (Nakos & Mitropoulos, 2005), while Thapa is a geometrical model consisted of representative line shape samples introduced by Thapa (1988).

The visualization of the aforementioned datasets using heatmaps and APLs are shown in Figure 7.

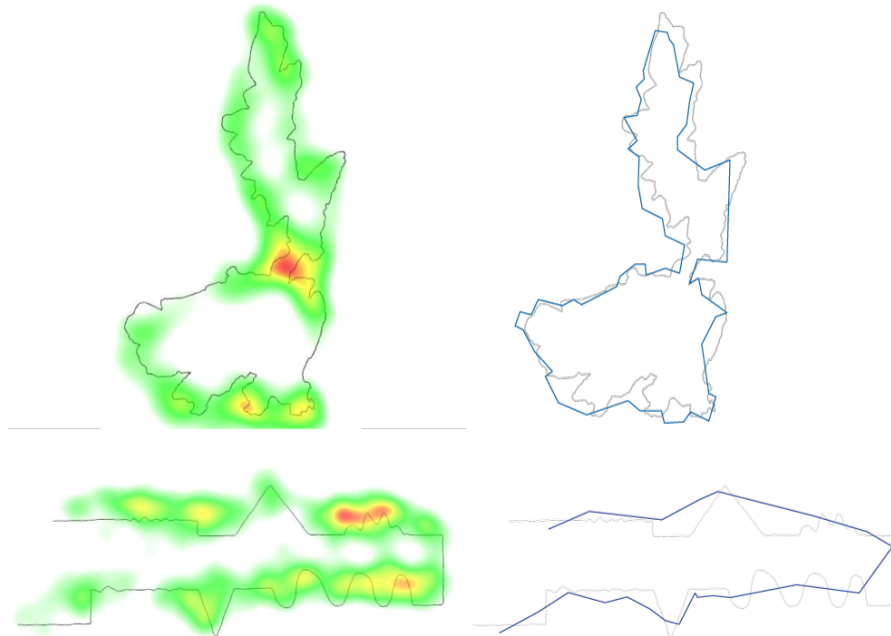


Figure 7. Visualizations of eye tracking data from Peristera (top) and Thapa (bottom) using heatmaps (left) and inferred APLs (right).

In the sequel, we executed the workflow discussed in section 3.1 and came up with one table T for each experiment, T_p and T_t , for Peristera and Thapa, respectively. Data in T_p and T_t was used in statistical analysis to demonstrate how research questions of the two types mentioned in Section 3.1 can be answered. As discussed, two metrics have been used: the Hausdorff distance and the relative length calculated as the ratio of the length of PL_i of each sample, divided by the length of the APL for each experiment.

Type 1 questions: visualizations and line complexity. In Figure 8 the normalized distribution of RHi , calculated as the Hausdorff distance of each sample PL_i from the APL, divided by the length of the APL, is shown. Although the two experiments have not been

executed with this kind of analysis in mind, it is clear that the statistical distribution of the deviations of individuals' observations in the simpler cartographic line of Thapa is considerably better than the corresponding distribution of Peristera.

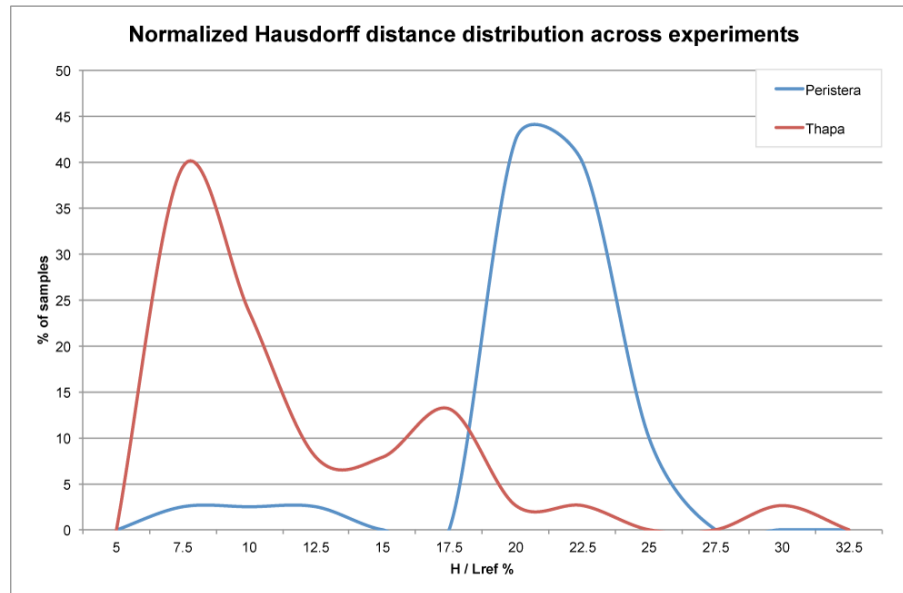


Figure 8. Normalized distribution of Hausdorff distance across experiments

Type 2 questions: behavior of different groups. To study how attributes of different groups can be related to results extracted by the approach introduced in this work, we studied the distribution of two metrics applied to the Thapa experiment data: RHi based on the Hausdorff distance as discussed above, as well as Ri calculated as the ratio of the length of each PLi divided by the length of the APL.

The normalized distribution of RHi was calculated for two groups of subjects participating in the Thapa experiment, as shown in Figure 9. Group 1 consists of 14 people aged over 45 where the 24 members of group 2 are aged below 40. All the subjects declared that their vision is accurate without the need for glasses or contact lenses. The corresponding distributions shown in Figure 9, indicate that the deviations of individuals' observations in group 1 is noticeably higher than the corresponding deviations of group 2. Of course, whether this is any indication of any kind of not diagnosed refractive errors is out of the scope of this work.

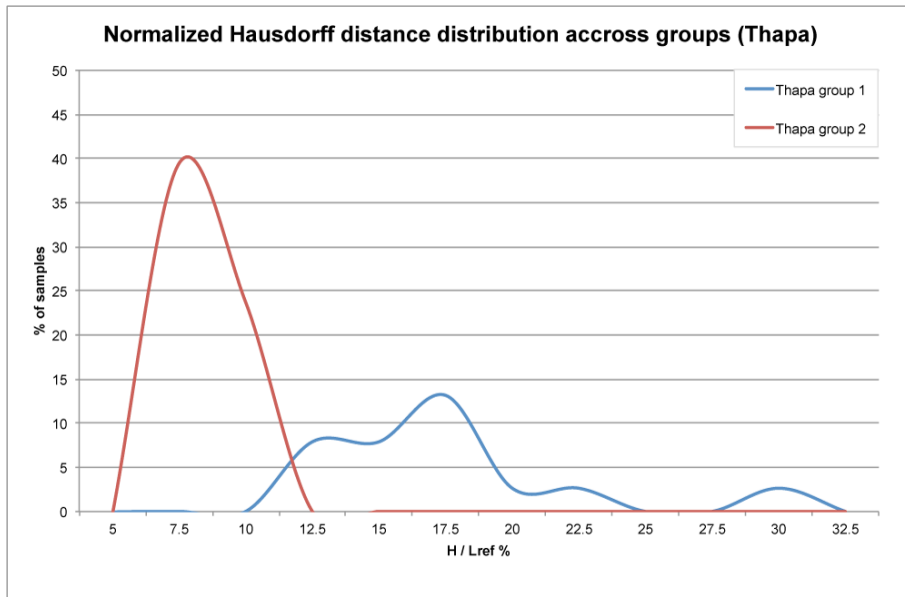


Figure 9. Normalized Hausdorff distance distribution across groups for the "Thapa" experiment

Finally, the normalized distribution of Ri for the aforementioned groups is shown in Figure 10. Although further experimentation is needed to jump to definite conclusions, it is also evident that the behavior of the two groups is noticeably different.

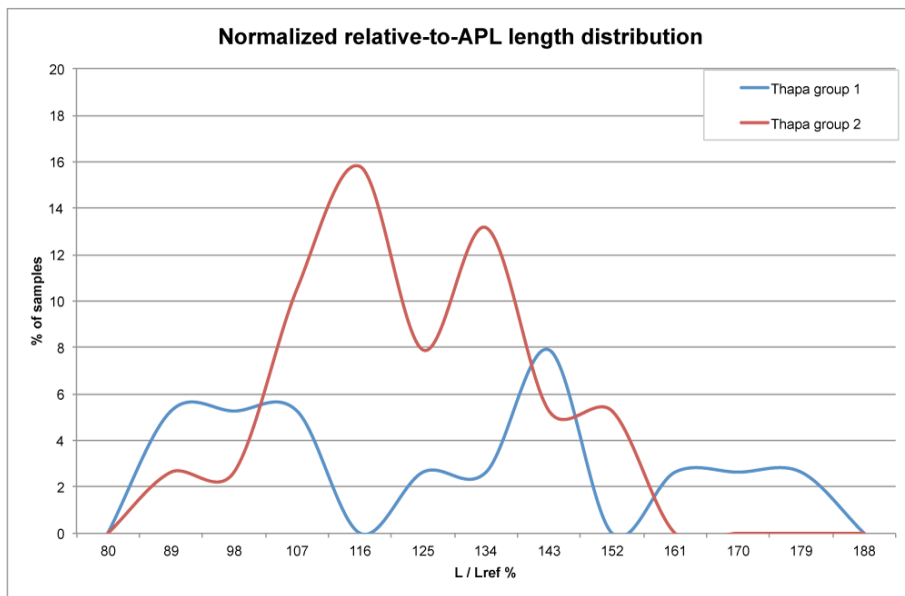


Figure 10. Normalized distribution of L_i/L_{ref} across groups for the "Thapa" experiment.

5 Conclusion and discussion

The huge amount of gaze data collected in eye tracking studies requires their analysis and visualization in order to answer the research questions posed in every case. The majority of the well-established eye tracking visualization techniques, such as heatmaps, are mainly used for qualitative evaluations. In this paper we introduce a new visualization method for the representation of the average gaze behavior during the observation of stimuli in which linear entities such as lines, routes, paths etc. exist, and the research question requires the quantitative study of their observation, including the study of the behavior of each individual. The approach of the introduced method is based on the generation of a reference average line that we refer as APL. Currently the APL is a flat polyline, but in the future it can be further improved by adding color attributes to the line segments of this polyline, using calculations such as data density of eye tracking samples near the line, or other metrics.

Except from the main advantage that allows the depiction of the average gaze behavior of a population, this approach enables us to compare the visual reaction of individual subjects in a group through analytical techniques used in similarity analysis. Hence, the proposed method gives an alternative point of view that extends the capability of eye tracking visualizations allowing group comparison reported by specific metrics such these used in the present study.

This idea may be applied in domains where eye tracking is already used, to enhance the analysis of eye tracking data. Both a visualization and a framework for quantitative analysis, the APL approach can be useful in cases where the context of eye tracking has reference to lines, paths, etc. that subjects are required or expected to follow. One such case is cartography where borders, navigation routes and all kinds of curves, are used to represent useful information on a map and the research question is about the effects of different visualization attributes of cartographic lines in the concentration of the eye's attention to a "main" linear entity.

Several other applications of the possibility of quantitative analysis that are very challenging also come into mind. These include education, behavioral and intention analysis, psychology and ophthalmology, to mention a few. For example, analyzing how a drawing is observed by members of groups with specific attributes, may lead to new methods for early diagnosis of situations in psychology or ophthalmology.

Another promising application area might be medical education, where the behavior of

medical students in the observation of images where a linear geometry is dominant, such as a e.g. cardiogram, might be an indication of how they perceive attributes of the cardiogram that lead to a diagnosis; also, deviations from the average or the expert's behavior might be indications that, combined with other data, might be useful in the evaluation (grading) of medical students asked to observe a cardiogram. Needless to say, all of the above possible areas for applying the APL visualization and analytics approach introduced in this paper, need to be individually addressed and specific experiments need to be designed for each case with APL visualization and analysis in mind.

6 References

- Ahmed, M., & Wenk, C. (2012). Constructing street networks from gps trajectories. In *Proceedings of the 20th Annual European Symposium on Algorithms*, 60-71, Ljubljana, Slovenia.
- Alt Helmut and Guibas J Leonidas. "Discrete geometric shapes: Matching, interpolation, and approximation". In: *Handbook of computational geometry 1* (1999), pp. 121–153.
- Bargiota, T., Mitropoulos, V., Krassanakis, V., & Nakos, B. (2013). Measuring locations of critical points along cartographic lines. In *Proceedings of the 26th International Cartographic Conference*, Dresden.
- Biagioni, J., & Eriksson, J. (2012). Map inference in the face of noise and disparity. In *Proceedings of the 20th ACM SIGSPATIAL GIS International Conference*, 79-88, Redondo Beach, CA.
- Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., & Ertl, T. (2014). State-of-the-art of visualization for eye tracking data. In R. Borgo, R. Maciejewski, & I. Viola (Eds) *Proceedings of Eurographics Conference on Visualization (EuroVis)*.
- Bojko, A. (2009). Informative or Misleading? Heatmaps Decostruction. In Jacko, J.A. (Eds.) *Human-Computer Interaction*. pp. 30-39, Springer-Verlag, Berlin.
- de Bruin J.A., Malan K.M., & Eloff J.H.P. (2013). Saccade deviation indicators for automated

eye tracking analysis. In *Proceedings of the 2013 Conference on Eye Tracking South Africa*, 47-54, Cape Town.

Dubuisson M-P and Anil K Jain. "A modified Hausdorff distance for object matching". In: *Pattern Recognition, 1994. Vol.1-Conference A: Computer Vision & Image Processing*, Proceedings of the 12th IAPR International Conference on. Vol.1. IEEE. 1994, pp. 566–568.

Duchowski, A. T. (2002). A breath-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34 (4), 455-470.

Duchowski, A.T. (2007). *Eye Tracking Methodology: Theory & Practice* (2nd ed.). London: Springer-Verlag.

Edelkamp, S., & Schroedl, S. (2003). Route planning and map inference with global positioning traces, In: *Computer Science in Perspective*, Springer-Verlag, 128-151, New York.

Ehmke, C., & Wilson, S. (2007). Identifying web usability problems from eye-tracking data. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it*, Volume 1 (pp. 119-128). British Computer Society.

Ester, M., Kriegel, H.P., S, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 226-231, Portland, OR.

Goldberg J. & Helfman J. (2009), Pattern Analysis in Eye Tracking Data: Dotplots Revisited, In *Proceedings of the CHI 2009*, Boston, MA, USA.

Goldberg, J. H. & Kotval, X. P. (1999). Computer inter- face evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24, 631-645.

Hagedoorn Michiel. "Pattern matching using similarity measures". PhD thesis. Faculteit Wiskunde, 2001.

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press.

Jacob, R.J.K., Karn, K.S. (2003). Eye Tracking in Human-Computer and Usability Research: Ready to Deliver the Promises. In: Hyona et al. (Eds.) *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements*. pp.573-605. Elsevier Science, Oxford.

Karagiorgou S, Krassanakis V., Vescoukis V., & Nakos B., (2014). Experimenting with polylines on the visualization on eye tracking data from observation of cartographic lines. In P. Kiefer, I. Giannopoulos, M. Raubal, A. Krüger (eds.), *Proceedings of the 2nd International Workshop on Eye Tracking for Spatial Research*, Vienna, Austria, pp. 22-26.

Karagiorgou, S. (2014), Inference of transportation networks from sparse tracking data, Ph.D. Thesis, School of Surveying Engineering, National Technical University of Athens.

Karagiorgou, S., & Pfoser, D. (2012). On vehicle tracking data-based road network generation. In *Proceedings of the 20th ACM SIGSPATIAL GIS International Conference*, 89-98, Redondo Beach, CA.

Krassanakis, V., Filippakopoulou, V., & Nakos, B. (2011). An Application of Eye Tracking

Krassanakis, V., Filippakopoulou, V., Nakos, B. (2014). EyeMMV toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification. *Journal of Eye Movement Research*, 7(1):1, 1-10.

Li, X., Çöltiken, A., Kraak, M.J. (2010). Visual exploration of eye movement data using Space-Time-Cube. In: Fabrikant et al. (Eds.) *Geographic Information Science*. pp.295-309. Springer-Verlag, Berlin.

Methodology in Cartographic Research. In *Proceedings of the EyeTrackBehavior (Tobii)*, Frankfurt.

Poole, A., & Ball, L. J. (2005). Eye Tracking in Human- Computer Interaction and Usability Research: Current Status and Future Prospects. In C. Ghaoui (Eds.), *Encyclopedia of human computer interaction*. pp. 211- 219. Pennsylvania: Idea Group.

Räihä, K. J., Aula, A., Majaranta P., Rantala, H., Koivunen, K. (2005). Static Visualization of Temporal Eye-Tracking Data. In: Costabile, M.F. And Paternò, F. (Eds.) *INTERACT 2005*. pp 946-949.

Richardson, D. C. (2004). Eye tracking: Research areas and applications. In G. Wnek & G. Bowlin (Eds.), *Encyclopedia of biomaterials and biomedical engineering* (pp. 573-582). New York: Marcel Dekker.

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying Fixations and Saccades in Eye-Tracking Protocols. In *Proceedings of the Symposium on Eye Tracking Research and*

Applications, 71-78.

Scharf Ludmila. "Computing the Hausdorff distance between sets of curves". PhD thesis. Thesis, Freie Universität, Berlin, 2003.

Shapiro D Michael and Matthew B Blaschko. "On hausdorff distance measures". In: Computer Vision Laboratory University of Massachusetts Amherst, MA 1003 (2004).

Špakov O, Miniotas D (2007). Visualization of Eye Gaze Data using Heat Maps. *Electronics and electrical engineering*, 2(74), 55-58.

Thapa K (1988). Automatic Line Generalization in Raster Data Using Zero-Crossings. *Photogrammetric Engineering and Remote Sensing*, 54(4): 511-517

Veltkamp C. Remco. "Shape matching: Similarity measures and algorithms". In: Shape Modeling and Applications, SMI 2001 International Conference on. IEEE. 2001, pp. 188–197.

Zhang, T., Ramakrishnan, R., Livny, & Birch M. (1996). An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 103-114, Montreal, CA.

