RECURRENCE QUANTIFICATION ANALYSIS OF GROUP EYE TRACKING DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

MANI TAJADDINI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCES

AUGUST 2018

RECURRENCE QUANTIFICATION ANALYSIS OF GROUP EYE TRACKING DATA

Submitted by Mani Tajaddini in partial fulfillment of the requirements for the degree of **Master of Science in the Department of Cognitive Sciences, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics** _____

Prof. Dr. Cem Bozsahin
Head of Department, **Cognitive Science** _____

Assoc. Prof. Dr. Cengiz Acarturk
Supervisor, **Cognitive Science Dept., METU** _____

Examining Committee Members:

Prof. Dr. Cem Bozşahin
Cognitive Science Dept., METU _____

Assoc. Prof. Dr. Cengiz Acartürk
Cognitive Science Dept., METU _____

Prof. Dr. Deniz Zeyrek Bozşahin
Cognitive Science Dept., METU _____

Prof. Dr. Murat Perit Çakır
Cognitive Science Dept., METU _____

Asst. Prof. Dr. Özkan Kılıç
Computer Engineering Dept., Yildirim Beyazit University _____

Date: **28.08.2018**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mani Tajaddini

Signature : _____

ABSTRACT

RECURRENCE QUANTIFICATION ANALYSIS OF GROUP EYE TRACKING DATA

Tajaddini, Mani

MSc., Department of Cognitive Sciences

Supervisor: Assoc. Prof. Dr. Cengiz Acarturk

August 2018, 87 pages

Eye movements can provide insight into the cognitive processes behind them. To study group cognition through investigating eye movements, we have developed a software tool (*GETapp*) to collect eye movement data from groups of participants performing a task on a set of computers in the scope of the *Group Eye Tracking (GET)* paradigm. Like many real-world systems, the data from group eye tracking experiments are non-linear. We have developed a software tool, a package for the R programming language, called the `generalRQA` package for analyzing these non-linear data. The methods used in this package are based on recurrences of time-series and implement a part of the theory of *Recurrence Quantification Analysis (RQA)*. In this thesis we describe both the software tools and apply them to propose measures that differentiate between different group formation patterns in a group eye tracking search task.

Keywords: Group Eye Tracking, Recurrence Quantification Analysis, Nonlinear

ÖZ

GROUP EYE TRACKING VERILERI UZERINDE RECURRENCE QUANTIFICATION ANALYSIS

Tajaddini, Mani

Yüksek Lisans, Bilişsel Bilimler Bölümü

Tez Yöneticisi: Doç. Dr. Cengiz Acartürk

Ağustos 2018, 87 sayfa

Göz hareketleri arkalarındaki bilişsel proselere içgörü sağlayabilir. Göz hareketlerini incelemekle gurup bilişimini araştırmak için bir yazılım geliştirdik. Bu yazılım aracılığıyla *group eye tracking (GET)* kapsamında yer alan bir gurup denekle ve bir gurup bilgisayar üzerinde yapılan deneylerde göz verileri topladık. *Group eye tracking* verileri, her gerçek-dünya sisteminde olduğu gibi, nonlineerdirler. Bu nonlineer verileri analiz etmek için `generalRQA` isimli ve R package'i formunda bir yazılım geliştirdik. Bu yazılımdaki kullanan yöntemler *recurrence* üzerinde bina olup *Recurrence Quantification Analysis (RQA)* teorisinin bir kısmını uygulamaktadır. Bu tezde her iki yazılımı tanımlayıp onları farklı gurup oluşum desenlerini birbirinden ayırt edebilen ölçümler bulmak için uyguladık.

Anahtar Sözcükler: Group Eye Tracking, Recurrence Quantification Analysis, Nonlineer

To My Beloved Ones

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude towards my advisor, Assist. Prof. Dr. Cengiz Acarturk for his guidance and encouragement.

Besides my supervisor, I would like to thank all my beloved ones who helped supported me through my studies and through life.

I would also like to thank all the students of the department who assisted me in carrying out the data collection and software development.

This research was partially supported by TUBITAK 1001 Project, Grant no 116E570, Multiuser Eyetracking Platform for Social Gaze

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PSEUDOCODES

LIST OF ABBREVIATIONS

RQA	Recurrence Quantification Analysis
CRQA	Cross Recurrence Quantification Analysis
JRQA	Joint Recurrence Quantification Analysis
sts	Server Time Stamp
cts	Client Time Stamp

CHAPTER 1

INTRODUCTION

Coming together in groups may be regarded as one of the most important characteristics of humans. The majority of human activities, including learning, working, playing, sleeping, worshiping, etc. are done in relation to others in the context of a group. Humans live in highly sophisticated societies and therefore they need to frequently make decisions and act on them collectively. From small household management to state legislation, most is done in the context of group and is dependent on group consensus. Likewise, interpersonal conflicts in human societies may result from not being able to reach a group consensus (Conradt & List, 2009; Forsyth, 2014).

With the advent of internet and social media, coming together in online groups has become much easier than before. Whereas in the past an individual may have been a member of a handful of groups, like the family and friends, nowadays, each person belongs to a myriad of different groups under various contexts. The group action effects, constrains and its guiding the individual members, have become more dynamic and complex in the context of social media as the rate of information sharing has skyrocketed in the recent era. This has brought advantages and challenges. Even though coming together in groups and sharing information have become much easier than before, an understanding of how digital media may conciliate human collaboration is still rudimentary (Ludvigsen, Law, Rose, & Stahl, 2017; Stahl, 2006).

Accordingly, it is necessary to understand how actions, processes and changes, occur inside a group and between groups. This is called the *group dynamics* or the study of how groups change over time. Group dynamics is a broad area as there are many aspects to a group that can be studied. In this thesis, the dynamics of group formation will be investigated in the context of a visual search task. We have developed a software tool to measure group dynamics in gaze behavior. A set of measures have been developed, which were used for data analysis in a visual search experiment. The experiment was carried out in a lab setting and the search task was performed by participants on a group of computers. The data consisted of gaze locations of the participants, which were collected using eye tracking devices.

We studied the dynamics of group formation in eye tracking using a non-linear time-series analysis technique, called the *recurrence quantification analysis (RQA)* (Webber Jr & Marwan, 2015). This method, besides giving us insights into the dynamics of group formation in a collective search task, introduces new measures in the group eye tracking domain. The goal of the current study is to explore these measures by comparing various conditions in the experiment design. The experiment conducted was in three conditions where the search target of the task (a dot with a fixed radius of few pixels) was *dark*, *light* and *absent* and in two conditions where a marker (a red circle with its center corresponding to the mean collective gaze location and its radius corresponding to the standard deviation of the gaze locations) indicating the mean location of the collective gaze was present or absent. In essence, the research goals of this study involve developing a software tool which is able to:

- investigate RQA measures among the three conditions of the search target (*dark*, *light*, and *absent*)
- investigate RQA measures when a marker–as a means of communication among group members–is present or absent
- investigate if the previous two findings hold for various larger size groups

The initial goal of the thesis is to develop the software tool for the analysis. As for the experimental predictions, our hypothesis is that firstly, as the search becomes more difficult due to reduced visual saliency of the object on the display, the *correlation* among the time series of participants (referring to the time series composed of sequences of gaze locations per participant) will decrease, whereas the *complexity* of the time series will increase. Secondly, the time series will show less structure when there is no marker signaling the mean of the collective gaze location. This is because, when there is a marker present on the display while the participants search for a target actively, they will tend to look for hints from the group. Lastly, we expect that the effects studied through the RQA measures under different conditions will be similar for different group sizes.

Chapter 2 introduces eye tracking along with an outline of the concept of recurrence analysis in dynamical systems by introducing the Poincaré recurrence theorem, the Takens' embedding theorem and the theory of recurrence quantification analysis which describes the mathematical basis for the measures which were used in the analysis of the experimental data in this study. Chapter 3 provides a description of the software tools we have developed for collecting data (the *GETapp*) and analyzing the data (the *generalRQA* package) using an example dynamical system. Chapter 4 explains the actual data collection procedure and presents the results of the data analysis and chapter 5 provides a discussion on the results, the limitations of the methods used in the current study and the possible future work.

CHAPTER 2

LITERATURE REVIEW

2.1.Analysis of Recurrences

The process of understanding the physical world through scientific inquiry involves observation, measurement, analysis and possibly prediction. Linear systems have characteristics that make them ideal subjects for scientific investigation; however, nature is abundant with processes and systems which are non-linear and non-predictable (Charles L Webber & Zbilut, 2005). The study of systems through time is done in virtue of systems dynamics. The trajectories of such systems contain phase singularities and noise. These trajectories have been studied by linear tools in the temporal and frequency domain. However, these tools are sometimes not enough to capture non-linear characteristics of a system under study and often consider the non-linearities as noise. Basic to the field of non-linear analysis is the representation of system trajectories in a *phase space*. Successful characterization of these non-linear systems has been done by non-linear analysis tools such as the Liyapunov exponent, Kolmogorov-Sinai entropy and correlation dimension. One assumption in these methods is that the system is stationary. A stationary system is a system that retains its average characteristic measures over time. Nonetheless, most of the systems that are ubiquitous in nature are non-stationary and non-linear. A practice that has emerged in the recent years to tackle this limitation of the mainstream methods of analysis is the analysis of recurrences (Webber Jr & Marwan, 2015).

Recurrence is a fundamental feature of systems evolving in time which can be used to characterize the system's behavior in phase space. The idea was first proposed by Poincaré, known as Poincaré's recurrence theorem. Poincaré's theorem was merely qualitative; however, further studies and experiments have shown that recurrences contain all the relevant information about a system's behavior. In 1980's a powerful method was proposed to visualize recurrences in a dynamical system, called *recurrence plot* (Eckmann, Kamphorst, & Ruelle, 1987). The quantification of the recurrence plot is what is now called *Recurrence Quantification Analysis (RQA)*. This method of non-linear analysis does not require the system to be stationary and it is efficient in detecting transitions in the dynamics of systems from time series (Norbert Marwan, Carmen Romano, Thiel, & Kurths, 2007a).

Recurrence quantification analysis has been successfully used in various fields like economy, neuroscience, physiology, psychology, behavioral sciences, health sciences, earth sciences, astrophysics and engineering among others (Webber, Ioana, & Marwan, 2016). The current study takes advantage of the fact that recurrence quantification analysis

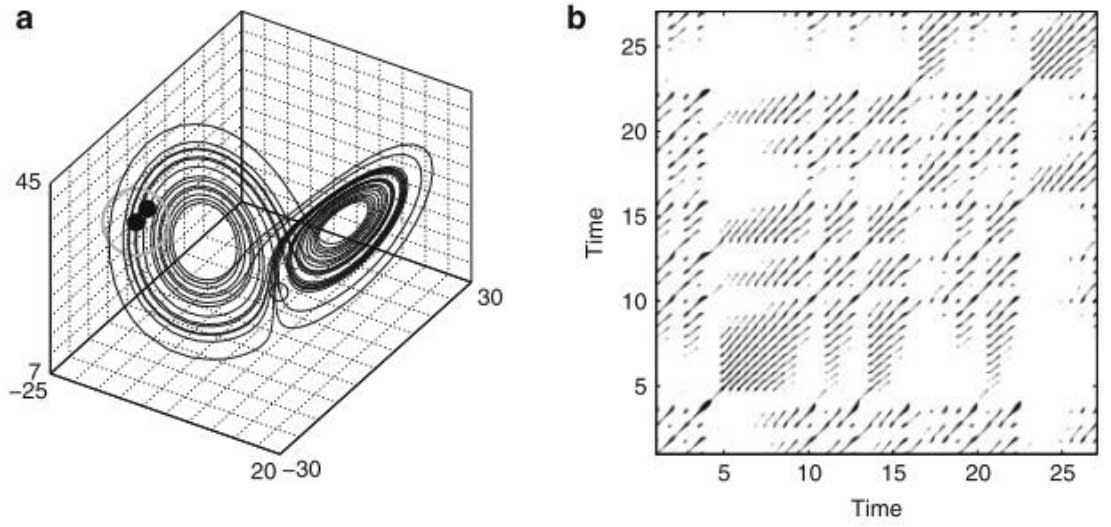


Figure 1 a) a Lorenz attractor in 3-dimensional phase space. b) the Recurrence Plot of the Lorenz attractor. A point of the trajectory at j which falls into the gray circle in (a) of a given point at i is considered as a recurrence point [black point on the trajectory in (a)]. This is marked with a black point in the RP at the location (i,j). A point outside the small circle in (a) causes a white point in the RP. The radius of the circle for the RP is D=5. The recurrence plot shows short diagonal line structures representative of a chaotic behavior (Webber et al., 2015, p. 7)

allows studying and quantifying interactions of coupled systems and is an appropriate tool

for the study of synchronization of complex systems (Norbert Marwan et al., 2007a), like conversational dynamics and natural semiotics (Fusaroli & Tylén, 2016; Orsucci et al., 2006). In the field of eye tracking, there is research based on RQA, investigating subjects such as expert vs. novice behavior and characterization of fixation sequences (Anderson, Bischof, Laidlaw, Risko, & Kingstone, 2013; Vaidyanathan, Pelz, Alm, Shi, & Haake, 2014); however, as of our knowledge, it has not been used for investigating group eye tracking as a measure of group dynamics.

In the next section the Poincaré recurrence theorem and subsequently the Takens' embedding theorem will be introduced. These two theorems make the basis for recurrence quantification analysis.

2.2. Poincaré Recurrence Theorem

In the early 1890's the study of celestial mechanics was based on finding individual solutions to differential equations. The analysts tried to extract as much information as possible about the individual solutions to a system of differential equations by analytical methods. Poincaré shifted this paradigm towards the global analysis of phase space. His ideas resulted in the establishment of two new fields in the study of dynamics: topological dynamics and ergodic theory. In topological dynamics the phase space is abstracted into the topological space and the topological structures of the trajectories and attractors are studied. In ergodic theory, the phase space is replaced by an abstract measure space and the dynamics of the system are studied as the behavior of a group of measure-preserving transformations (Furstenberg, 1981).

In short, Poincaré's recurrence theorem states that in certain systems, the system will come arbitrarily close (recur) to an arbitrary state given enough time. The time that it takes for the system to recur to a state is called the recurrence time (Poincaré, 1890; Wikipedia contributors, 2018). Poincaré's theorem is merely qualitative and does not provide quantitative measures. In the next two sections, efforts to quantify Poincaré's theorem are outlined.

2.3. Taken's Theorem

A *phase space*, necessary for modeling dynamical systems, is the collection of possible states of a dynamical system. A system's state at time *t* is a collection of all the information needed to define that state uniquely. The phase space of the mathematically modeled systems is known from its time dependent equations; however, for naturally occurring chaotic dynamical systems, the mathematical description of the system is usually unknown. In these situations, phase space reconstruction methods are employed. The phase space reconstruction methods use one or more observed time signals from the system to construct a surrogate of the original phase space.

The Whitney Embedding Theorem (Whitney, 1936) defines embedding as a mapping from an n-dimensional space to 2n+1 dimensional Euclidean space. After the embedding the n-manifold has unfolded completely in the larger space. This means that no two points in the n-manifold map to the same point in the larger space. In general, Whitney Embedding Theorem maintains that 2n+1 independent signals measured from a system, can be considered as a map from the set of states to 2n+1-dimensional space. In this space, each state can be identified uniquely by a vector of 2n+1 measurements, thereby reconstructing the phase space.

The Takens Embedding Theorem (Takens, 1981) supplemented the Whitney Theorem by maintaining that the same goal can be reached by a single observed time signal from the system. Takens in his seminal work, proved that a series of 2n+1 time-delayed signals acquired from a single signal, instead of 2n+1 signals, can be used to embed the n-manifold (an n-dimensional space). The idea of using time-delayed signals to reconstruct a phase space is reminiscent of an existence theorem of ordinary differential equations, which states that a unique solution exists for a series of time derivatives of the dependent variable.

With the advent of chaos theory and fractal geometry, a reassessment of the original theory, concerning smooth manifold attractors was proposed. It was shown that an attractor with box-counting dimension of d can be reconstructed with m observed signals or m time-delayed versions of a single observed signal from the system, where m is any integer greater than 2d (Sauer, Yorke, & Casdagli, 1991).

An embedding is not always successful and depends strongly on the specific application. If there are only weak connections between the degrees of freedom of the system and the aspect of the system being observed, it can affect the adherence of the reconstructed trajectory. To add to this, the difference of time scales in different parts of the system and the system and observational noise may be limiting factors for the practice of embedding as well.

A system's state is specified by its *n* state variables. If we represent the state of the system at time *t* as a vector in a *n-dimensional* space, that space is called a *phase space*.

$$\boldsymbol{X(t)} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{1}$$

This vector moves in time and in the direction of its velocity vector $\partial X/\partial t$. The temporal succession of the phase space vectors forms a trajectory. This trajectory, its topology and structure and its time evolution explain the dynamics of the system. The phase trajectory can be specified by integrating the velocity vector over time; however, insights about the system can be obtained from the shape of the trajectory by visualizing it without the need for integration (Webber Jr & Marwan, 2015).

In real-life situations, observations do not yield all possible state variables. It is usually the case that either not all the variables are known or not all of them are measurable; most often, only one state variable is available for observation. Taken's theorem states that because of the coupling between system's components, each single component contains essential information about the dynamics of the whole system. Because of this, by using only one time series, it is possible to reconstruct a phase space that conserves the topological structures of the original phase space (Packard, Crutchfield, Farmer, & Shaw, 1980; Takens, 1981).

When collecting data, measurement of a state variable $u(t)$ results in a discrete time series $\boldsymbol{u}_i$ where $t = i\Delta t$ and Δt is the sampling rate (here the continuous variables are denoted by braces and the discrete variables by subscripts). A method frequently used for reconstructing a trajectory $\widehat{\boldsymbol{X}}(t)$ from the original time series $u(t)$ such that it conserves the topological structures is called the *time delay* method. Having the measured original time series $\boldsymbol{u}_i$, we define the reconstructed trajectory as:

$$\widehat{\boldsymbol{X}}_i = [u_i \quad u_{i+\tau} \quad \dots \quad u_{i+(m-1)\tau}] \tag{2}$$

where m is the embedding dimension and τ is the time-delay parameter (here τ is the index-based time delay, whereas the real time delay is $\tau\Delta t$). The preservation of the original trajectory is guaranteed if $m \geq 2d + 1$ where d is the dimension of the attractor (Takens, 1981).

The embedding parameters, the time-delay and the embedding dimension need to be chosen appropriately. Random errors and low measurement precision lead to linear dependence between the columns of $\widehat{\boldsymbol{X}}_i$ (Eq.2). The delay parameter should be chosen such that the dependence approaches zero. A method to accomplish this task is to determine the time-delay parameter using the *mutual information* (Fraser & Swinney, 1986). Intuitively, the mutual information is the average amount of information that a value carries with itself to the next value. The mutual information is given by the formula:

$$I(\tau) = -\sum_{\varphi,\psi} p_{\varphi,\psi}(\tau)\, log \frac{p_{\varphi,\psi}(\tau)}{p_\varphi p_\psi} \tag{3}$$

Here $I(\tau)$ represents the mutual information at time τ. p_φ and p_ψ are the probabilities that $u_i = \varphi$ and $u_{i+\tau} = \psi$ respectively and $p_{\varphi,\psi}(\tau)$ is the joint probability that $u_i = \varphi$ and $u_{i+\tau} = \psi$. Therefore, the mutual information is not a function of the variables φ and ψ but of the joint probability $p_{\varphi,\psi}(\tau)$. The best choice of the time-delay parameter is where $I(\tau)$ has its first local minimum. The advantage of the mutual information over other methods for determining the time-delay parameter (e.g. autocovariance function) is that it is sensitive to nonlinear interrelations; however, experiment has shown that the mutual information method sometimes overestimates the time delay.

Among various methods for finding the optimal embedding dimension, in this study, we used the *false nearest neighbors* (FNNs) method (Kantz & Schreiber, 2004; Kennel, Brown, & Abarbanel, 1992). The intuition behind the FNN method is that as the dimension increases, some points in the neighborhood of an arbitrary phase space point are projected onto locations that are not in the neighborhood of that point. These points are called the false nearest neighbors. In other words, as the dimension increases, the number of false nearest neighbors decrease. The simplest method that uses false nearest neighbors, uses the amount of FNNs as a function of embedding dimension to find the minimal embedding dimension; that is where the FNNs vanish (Kantz & Schreiber, 2004). In the plot of FNNs vs. dimension, the plot shows a clear inflection point or an elbow (Figure 2) at the appropriate minimum dimension (Chelidze, 2017).

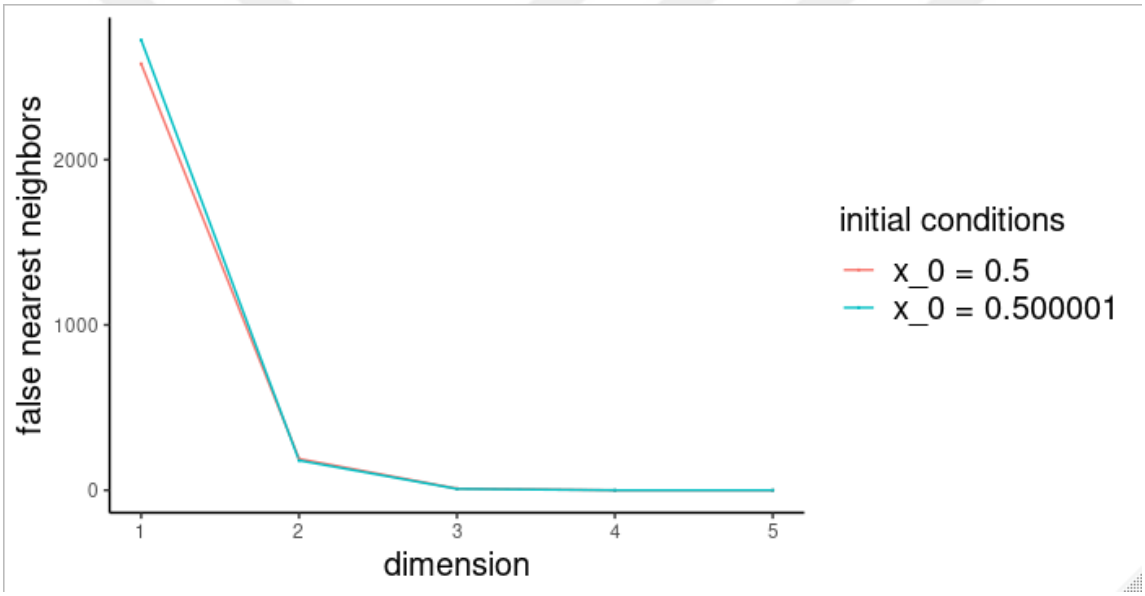


Figure 2 This plot shows the number of false nearest neighbors in the time series acquired from a logistic map with two slightly different initial points. The elbowing occurs when the dimension is 2 in both time series.

In this section we provided the theory of methods for embedding a single time series into a multi-dimensional phase space. In the next section we will introduce the basics of recurrence quantification analysis. One may or may not embed the data prior to recurrence analysis depending on the specifics of the problem.

2.4. Theory of Recurrence Quantification Analysis

2.4.1 Preliminaries

As mentioned in section 2.2, one of the fundamental properties of deterministic dynamical systems is the recurrence of the states of the system. Recurrence of states in the

sense that they get arbitrarily close to each other (cf. section 2.3) is typical for nonlinear and chaotic systems (Argyris, Faust, & Haase, 1994; Ott, 2002; Poincaré, 1890).

Eckmann et al. 1987 introduced a method of visualizing recurrences in a dynamical system called the *recurrence plot*. Briefly, a recurrence plot represents the times at which states $\widehat{\boldsymbol{X}}_i$ (section 2.3) recur in the corresponding phase space. The original intent behind the invention of recurrence plots was to provide insight into high-dimensional dynamical systems, the phase spaces of which are difficult to visualize otherwise (Eckmann et al., 1987; N. Marwan, 2008). A recurrence plot (RP) is a way of visualizing the *m-dimensional* phase space trajectory by representing a recurrence of a state at time *i* at a different time *j* within a two-dimensional square matrix $\boldsymbol{R}$ where both axes are time axes. The matrix is in Boolean form, meaning that the recurrent states are depicted as values of 1 and others as zero; however, if the matrix is depicted as a grid of pixels, the recurrences are represented as dots (black dots in the case of a black and white representation). The formal definition of the recurrence matrix is presented below (Webber Jr & Marwan, 2015):

$$R_{i,j}^{m,\varepsilon_i} = \Theta\left(\mathcal{E}_i - \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|\right), \quad \boldsymbol{x}_i \in \mathbb{R}^m, \quad i,j = 1,\dots,N \tag{4}$$

where N is the number of considered states $\boldsymbol{x}_i$; $\mathcal{E}_i$ is a threshold distance; $\|.\|$ a norm and $\Theta(.)$ the Heaviside function[1]. In the recurrence plot, because $R_{i,i} = 1$ by definition, the RP has a black main diagonal line called the *line of identity* (LOI). The line of identity should be ignored in quantifying the RP.

[1] A Heaviside function is a discontinuous function whose value is zero for negative argument and one for positive argument.

$$\Theta(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

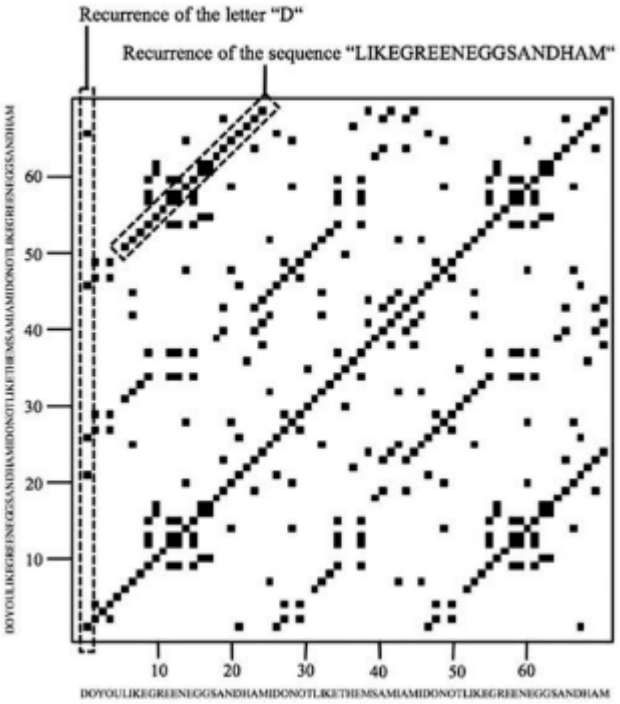


Figure 3 This plot shows the RP of a categorical data (namely the string "DOYOULIKEGREENEGGSANDHAMIDONOTLIKETHEMSAMIAMIDONOTLIKEGREENEGGSANDHAM"). Both of the axes contain the same string. The cells in the plot which correspond to the characters on the x and the y axes that are the same, are black. All the other cells are white. The diagonal line structures and the LOI are visible in the plot. (Wallot, 2017, p. 1)

A bivariate extension of the recurrence plot, called the *cross-recurrence plot* (CRP) was proposed to explore the simultaneous evolution of two different phase space trajectories. The dependencies and dynamics of the two trajectories relative to each other can be investigated using the CRP (Norbert Marwan & Kurths, 2002; Zbilut, Giuliani, & Webber, 1998).

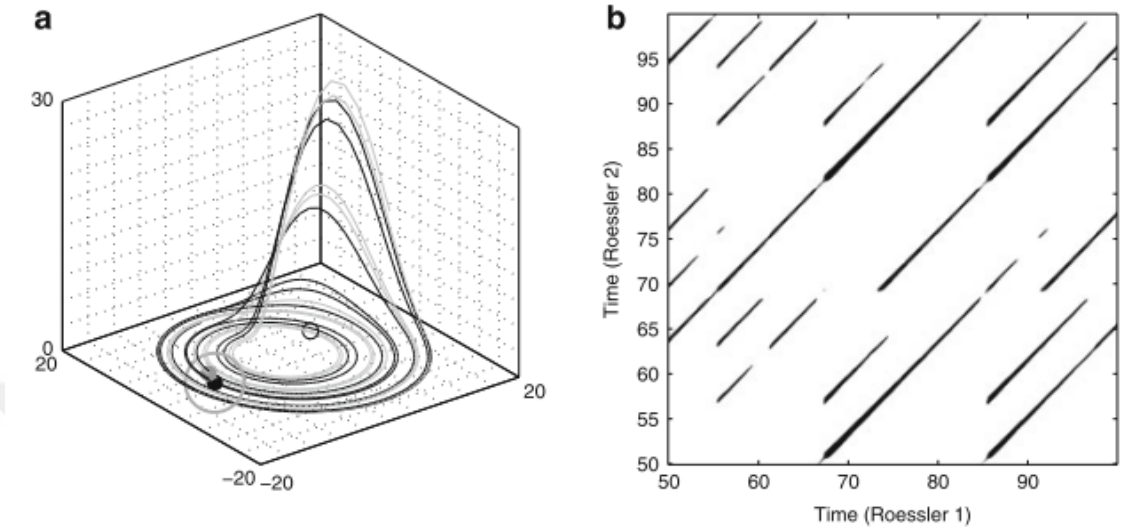


Figure 4 a) the embedded trajectories of two coupled Rossler systems in the phase space. b) the corresponding CRP. If a phase space vector of the second Rössler system at j (grey point on the grey line) falls into the neighbourhood (grey circle) of a phase space vector of the first Rössler system at i , in the CRP (b) at the location (i,j) a black point will occur. (Webber et al., 2015, p. 25)

Taking $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ to represent two trajectories of two dynamical systems in m-dimensional phase space the cross-recurrence plot can be constructed by computing the pairwise Euclidean distance between the different phase vectors of the two systems.

$$CR_{i,j}^{\boldsymbol{x},\boldsymbol{y}}(\varepsilon) = \Theta\left(\varepsilon - \left\|\boldsymbol{x}_i - \boldsymbol{y}_j\right\|\right), \quad i = 1, \dots, N, \quad j = 1, \dots, M \tag{5}$$

A $\boldsymbol{CR}$ matrix is not necessarily a square. This is because the length of the $\boldsymbol{x}$ and $\boldsymbol{y}$ trajectories do not need to be the same. It should be emphasized that both the systems are represented in the same phase space. This is because a cross-recurrence plot designates the times when the state of one system recurs to the state of the other system. If the embedding parameters are estimated for the time series (as mentioned in section 2.3) and are not equal for both of the them, the higher embedding dimension should be chosen. Furthermore, the data for the time series should be obtained from the same or similar processes and should represent the same observable (Webber Jr & Marwan, 2015).

The objects of interest in all variations of the recurrence plot are the macroscopic and microscopic structures; a single recurrence point at (i, j) does not bear any information about the states at the time i and j, but from the totality of the recurrence points the phase space trajectory can be reconstructed (Hirata, Horai, & Aihara, 2008; Norbert Marwan, Carmen Romano, Thiel, & Kurths, 2007b; Robinson & Thiel, 2009).

Exact recurrences of two states $\boldsymbol{x}_i = \boldsymbol{y}_j$ in (4) and $\boldsymbol{x}_i = \boldsymbol{y}_j$ in (5) are impractical and useless to find. This is because, as mentioned in the Poincaré's recurrence theorem in

section 2.2, the state of a chaotic system will not recur exactly to the initial state but will approach arbitrarily close to the initial state. Therefore, a recurrence is said to happen whenever a state gets closer than a certain threshold in an m-dimensional space to another state. This is expressed in (4) and (5) by the Heaviside function, its argument ε and the norm operator (Norbert Marwan, 2010; Norbert Marwan et al., 2007b).

In the current study, because we are investigating concurrent time series from different participants, the CRP fits better to our purposes. An advantage of the CRPs is that they are sensitive to the local differences of the dynamical evolution of trajectory segments near each other. A time dilatation or compression of one of the trajectories relative to another causes a distortion in the diagonal lines, usually in the form of bowed lines. In the remainder of this thesis we will focus mainly on the CRP and leave other versions of the recurrence plots in future work.

2.4.2. CRQA measures

The most commonly used norms for calculating the distance between two states in the phase space is the *L_2-norm* and the *L_∞-norm*. The L_∞-norm is often used because it is relatively easy to compute and is independent of the phase space dimension; however, it is also more prone to outliers and noise (Faure & Korn, 1998; M Thiel, Romano, & Kurths, 2003; Marco Thiel et al., 2002). In the current study, the eye tracking data used, is a highly noisy data because of the rapid and jumpy nature of eye movements. For this reason, the L_2-norm (Euclidean norm) was used in the calculations and will be the norm used in the remainder of the text.

The parameter ε, the recurrence threshold, is the third parameter after the time-delay parameter and the embedding dimension (if embedding is done), that should be determined in order to produce a CRP. Several studies have proposed methods for determining the recurrence threshold; however, a general and systematic study on determining the recurrence threshold remains an open task (Webber Jr & Marwan, 2015). The recurrence threshold selection procedure should be implemented such that it yields the minimum threshold possible but at the same time a threshold that reveals sufficient recurrences and recurrence structures. Essentially, the choice of the optimal recurrence threshold depends on the application and the experimental conditions (Norbert Marwan et al., 2007b; Matassini, Kantz, Hołyst, & Hegger, 2002; Marco Thiel et al., 2002; Webber Jr & Marwan, 2015).

There are various guidelines in the literature for choosing the optimal recurrence threshold, some of them are listed below:

- ε should be a few percent (not larger than 10%) of the maximum phase space diameter (Koebbe & Mayer-Kress, 1992; Mindlin & Gilmore, 1992; Zbilut & Webber, 1992)

- ε should be between 20% and 40% of the signal's standard deviation (for signal detection and classification) (Schinkel, Dimigen, & Marwan, 2008)

- ε is determine based on the diagonal structures within the RP. The threshold is chosen such that it minimizes the fragmentation and thickness of the diagonal lines (which change inversely to each other) (Matassini et al., 2002)

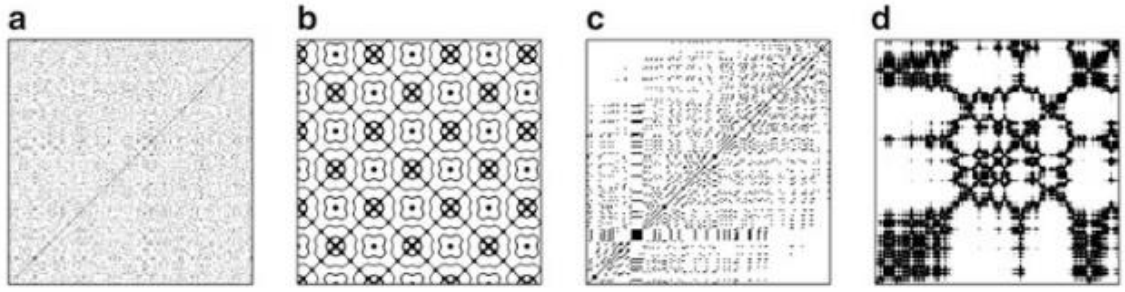
- ε should be five times larger than the standard deviation of the observational noise (Marco Thiel et al., 2002)

- ε is determined based on the recurrence point density of the RP. The recurrence point density is suggested to be 1% (Zbilut, Zaldivar-Comenges, & Strozzi, 2002)

In the current study, the methods of which are implemented in the `generalRQA` package (section 3.2) it is possible to either enter the recurrence threshold manually or enter a range for the recurrence point density so that the function chooses the recurrence threshold such that the recurrence point density falls in the range. In the analysis section (section 4.2) the threshold was entered manually because it was possible to offer an appropriate recurrence threshold based on the experiment settings (the procedure for choosing the threshold is explained in section 4.2).

Inspecting the CRP visually can provide useful insights in the dynamics of dynamical systems. For example, if there are many long diagonal lines, it means that the two systems are evolving in synchrony. More formally, if a state $\boldsymbol{x}_i$ in one system at time i recurs to another state $\boldsymbol{y}_j$ in the other system at time j, it can be expected that the two systems evolve together for a length of time with a time lag of $|i - j|$. If the CRP is homogeneous, it means that the two systems are hardly synchronized.

Figure 5 These plots show characteristic macroscopic manifestations of RPs corresponding to a) homogeneous (uniformly distributed noise), b) periodic (super-positioned harmonic oscillations), c) drift



(logistic map with a linearly increasing parameter) and d) disrupted (Brownian motion). (Webber et al., 2015, p. 10)

However, to gain insight from the visuals of the CRP is a subjective task. Furthermore, the insufficient resolution of the CRP can exacerbate the situation (even when the resolution of the CRP is higher than that of the display, visual artifacts can be seen and cause incorrect conclusions). To overcome the subjectivity of this methodology, Zbilut and Webber introduced definitions and procedures to quantify the RP (and its various

extensions, e.g. CRP) structures (Grassberger & Procaccia, 1983; Mindlin & Gilmore, 1992; Zbilut & Webber, 2007). The *recurrence quantification analysis (RQA)* term was coined by Zbilut and Webber and they defined five recurrence variables as complexity measures as part of the RQA. These five measures are based on diagonal line structures. The diagonally oriented lines in the CRP are of major interest because they represent segments on the trajectories of the two time series which evolve in parallel for some time. The length and frequency of these lines are connected to a certain similarity between the dynamics of the two systems and a measure based on them can be employed to detect nonlinear interrelations between the two dynamical systems which are impossible to find using common cross-correlation techniques (Norbert Marwan & Kurths, 2002).

The first measure is called the *percent recurrence (REC)* or *recurrence rate (RR)*:

$$RR(\varepsilon, N) = \frac{1}{N} \sum_{i=1,j=1}^{N} CR_{i,j}^{x,y}(\varepsilon) \tag{6}$$

where CR is the cross-recurrence matrix, ε is the recurrence threshold, N is the number of cells in the recurrence matrix and i and j are the indices of the matrix. The x and y superscripts on the CR matrix signify that the matrix is constructed from two different time series. The RR merely counts the black dots (or the cells with value 1) in the CRP. RR is a measure of the relative density of the recurrence points in the recurrence matrix and is related to the *correlation sum* (Grassberger & Procaccia, 1983); however, large segments of data is required to use RR as an estimator of the correlation sum.

The remaining four measures are based on the line structures in the CRP. To define these measures, we first need to construct a histogram of the lengths of the diagonal lines in the CRP:

$$H_D(l) = \sum_{i=1,j=1}^{N} \left(1 - CR_{i-1,j-1}\right)\left(1 - CR_{i+1,j+1}\right) \prod_{k=0}^{l-1} CR_{i+k,j+k} \tag{7}$$

Where l is the length of the diagonal line under consideration, CR is the cross-recurrence matrix, N is the number of cells in the CR matrix and i and j are the indices of the matrix.

The second measure is the *percent determinism (DET)*, defined as the ratio of the recurrence points that make diagonal lines to all the recurrence points:

$$DET = \frac{\sum_{l=d_{min}}^{N} l\, H_D(l)}{\sum_{i=1,j=1}^{N} CR_{i,j}} \tag{8}$$

where l is the length of a diagonal line, CR is the cross-recurrence matrix, N is the number of cells in the CR matrix and i and j are the indices of the matrix. DET can be interpreted as the predictability of the system. With a higher DET value we should expect longer times two systems spend synchronized. $d_{\min}$ sets the lower bound on the definition of a line. Typically, $d_{\min}$ is set to 2. If $d_{\min}$ is set to 1, DET and RR are identical. An extra derived variable called the *RATIO* can be defined as the ratio of DET to RR. The RATIO

can be computed directly from the frequency distributions of the lengths of the diagonal lines:

$$RATIO = N^2 \frac{\sum_{l=d_{min}}^{N} l\, H_D(l)}{\left(\sum_{l=1}^{N} l\, H_D(l)\right)^2} \quad (9)$$

where l is the length of a diagonal line, CR is the cross-recurrence matrix, N is the number of cells in the CR matrix, i and j are the indices of the matrix and $d_{\min}$ sets the lower bound on the definition of a line. During certain types of transitions on the dynamics of a dynamical system, the RR decreases while DET remains constant, hence the RATIO increases. Because of this, a study shows, the RATIO can detect dynamical transitions hence RATIO has proven to be useful in studying some physiological systems (C. L. Webber & Zbilut, 1994).

The third measure is the *maximal line length in the diagonal direction (D_{max})*:

$$D_{max} = \arg\max_{l} H_D(l) \quad (10)$$

which is the length of the longest diagonal line within the CRP. Since the diagonal lines designate segments of the trajectories of the two systems that run in parallel, this measure gives a hint about the divergence of the two systems' trajectories. This means that the smaller the D_{max}, the more divergent the trajectories. This definition suggests that there is a connection between the largest positive Lyapunov exponent[2] and D_{max}. The relation

---

[2] The Lyapunov exponents measures how quickly two states very close together diverge over time. In the equation below:

$$|F^t(x_0 + \varepsilon) - F^t(x_0)| \approx \varepsilon e^{\lambda t}$$

the exponent λ when t approaches infinity ($t \to \infty$) is the Lyapunov exponent. If $\lambda > 0$, initially close states diverge and if $\lambda < 0$, initially close states do not diverge. To gain more insight into Lyapunov exponent, we can show that:

$$\begin{aligned}
e^{\lambda t} &\approx \frac{|F^t(x_0 + \varepsilon) - F^t(x_0)|}{\varepsilon} \\
\lambda &= \lim_{t\to\infty,\varepsilon\to 0} \frac{1}{t} \log\left|\frac{F^t(x_0 + \varepsilon) - F^t(x_0)}{\varepsilon}\right| \\
&= \lim_{t\to\infty} \frac{1}{t} \log\left|\frac{dF^t}{dx}|_{x=x_0}\right| \\
&= \lim_{t\to\infty} \frac{1}{t} \log\left|\frac{dF}{dx}|_{x=F^{t-1}(x_0)=x_{t-1}} . \frac{dF}{dx}|_{x=F^{t-2}(x_0)=x_{t-2}} . \ldots . \frac{dF}{dx}|_{x=x_0}\right| \\
&= \lim_{t\to\infty} \frac{1}{t} \sum_{i=0}^{t-1} \log\left|\frac{dF}{dx}|_{x=x_i}\right|
\end{aligned}$$

can be explained using the frequency distribution of the lengths of the diagonal lines and the lower limit of the sum of the positive Lyapunov exponents (Norbert Marwan et al., 2007b). The *average diagonal line length (AvD)* can be derived from the D_{max}:

$$\langle D \rangle = \frac{\sum_{l=d_{min}}^{N} l\, H_D(l)}{\sum_{l=d_{min}}^{N} H_D(l)} \tag{11}$$

where l is the length of a diagonal line and $d_{\min}$ sets the lower bound on the definition of a line. The average diagonal length designates the mean time that the two segments of the two trajectories remain close to each other.

The fourth measure is the *Shannon entropy of the frequency distribution of the diagonal line lengths (ENT)*:

$$ENT = -\sum_{l=d_{min}}^{N} p(l) \ln p(l) \quad where \quad p(l) = \frac{H_D(l)}{\sum_{l=d_{min}}^{N} H_D(l)} \tag{12}$$

where N is the number of cells in the CR matrix, l is the length of a diagonal line, $d_{\min}$ sets the lower bound on the definition of a line and $p(l)$ is the probability of a diagonal line with length l occurring in the cross-recurrence plot. The ENT is a measure of the complexity of the deterministic structure in the interaction between the two systems. The ENT is measured in $\frac{bits}{bin}$. The higher the ENT the more complex the dynamics. For instance, the value of ENT for uncorrelated noise or oscillations is low indicating their low complexity. The ENT is sensitive to the choice of different parameters like the recurrence threshold ε and l_{min}.

All the *cross-recurrence quantification analysis (CRQA)* measures aforementioned, can also be computed separately for diagonals parallel to the diagonal with the indices $i = j$. We will denote these measures computed on a single diagonal by a subscript in the form of an asterisk from now on, e.g. RR_*. The measures calculated on a single diagonal are useful for investigating the periodicity of a signal (Zbilut & Marwan, 2008), periodic orbits (Gilmore, 1998; Lathrop & Kostelich, 1989; Mindlin & Gilmore, 1992) and the interrelationship between complex systems (Norbert Marwan, Thiel, & Nowaczyk, 2002).

The fifth measure is called the *trend (TND)*. The trend is a linear regression coefficient over the recurrence point density RR_* of the diagonals of the CRP as a function of the time distance between the diagonals and the diagonal with indices $i = j$.

---

where the final result means that the Lyapunov exponents is a time average of $\log\left|\frac{dF}{dx}\right|$ at every state along the trajectory (Sayama, 2015).

$$TND = \frac{\sum_{i=l}^{\widetilde{N}} \left(\frac{i-\widetilde{N}}{2}\right)(RR_* - \langle RR_* \rangle)}{\sum_{i=l}^{\widetilde{N}} \left(\frac{i-\widetilde{N}}{2}\right)^2} \quad (13)$$

where $\widetilde{N}$ is the subset of the cells of the CR matrix which are being considered, l is the length of a diagonal line and RR_* is the recurrence rate on a specific diagonal of the CR matrix. The TND gives information about the stationarity of the process. The TND value for quasi-stationary systems will be near zero and for nonstationary systems will be far from zero. This is because nonstationary systems usually show a drift in the system dynamics, often because the system is en route between more stationary states. In computing the TND, the edges of the RP are ignored ($\widetilde{N} < N$) because the diagonals near the edges are too short to hold any significant information.

As mentioned previously and as can be seen from (5), repeated below, this equation is designed to accommodate only two time series.

$$CR_{i,j}^{\boldsymbol{x},\boldsymbol{y}}(\varepsilon) = \Theta\left(\varepsilon - \left\|\boldsymbol{x}_i - \boldsymbol{y}_j\right\|\right), \quad i = 1, \dots, N, \quad j = 1, \dots, M \quad (5)$$

However, in the current study, we want to study the relative evolution of more than two time series. For the purposes of the current study we need to extend the definition of the CRP so that it can consider a higher number of time series. To do this we need to define the cross-recurrence plot as an M dimensional matrix. We consider a state recurrent if the largest distance between the M states from the M time series is less than the recurrence threshold. The CRP is given by:

$$CR_{i_1,i_2,\dots,i_M}^{\boldsymbol{x_1},\boldsymbol{x_2},\dots,\boldsymbol{x_M}}(\varepsilon) = \Theta(\varepsilon - \max(\|x_1 - x_2\|, \|x_1 - x_3\|, \dots, \|x_{M-1} - x_M\|)),$$
$$i_1 = 1, \dots, N_1, \quad i_2 = 1, \dots, N_2, \quad \dots \quad , \quad i_M = 1, \dots, N_M \quad (14)$$

where $N_1, N_2, \dots, N_M$ are the lengths of the time series and the max function chooses the largest mutual distance between the set of the considered M number of states. Furthermore, we need to look at a single diagonal at a time (the diagonal with the indices $i_1 = i_2 = \cdots = i_M$ specifically in section 4.2). Hence all the measures will be subscripted by an asterisk. This is because we are primarily interested in investigating the segments of the M time series that evolve *concurrently* with each other or with a certain set of delays (a set of $\frac{M(M-1)}{2}$ time delays to be explicit) mutually between the M time series.

2.5.Eye Tracking

Previous research show that there is a close relation between eye gaze and attention (Duchowski, 2017b; Kuhn, Tatler, Findlay, & Cole, 2008; Vickers, 2009). Accordingly, gaze data may provide insights into the cognitive processes during the course of performing a task. This reasoning has become the incentive for the interest into eye tracking devices in cognitive sciences. Recently, due to the advances in technology, eye trackers have become more accessible and affordable compared to the previous years. Manufacturers like Tobii,

Gazepoint, The EyeTribe[3], etc. manufacture professional eye trackers both for commercial and academic use.

Eye tracking technology in research has been used in many fields of study like psychology and neuroscience, infant and child research, virtual reality, marketing and consumer research, professional performance, user experience and interaction, sports performance and research, education and clinical research (Duchowski, 2017a; Holmqvist et al., 2011; Liversedge, Gilchrist, & Everling, 2011; K Rayner, 1997; Keith Rayner, 2009).

In the context of cognitive sciences dual eye tracking was the starting point for studying the cognitive processes behind human interaction. Dual eye tracking has been used to study mutual gaze during conversation, collaborative game playing and collaborative problem solving (Acarturk, Kalkan, & Arslan Aydın, 2018; Broz, Lehmann, Nehaniv, & Dautenhahn, 2012; Jermann, Nüssli, & Li, 2010; Nüssli, 2011) among others. Group eye tracking is the extension of dual eye tracking to study group cognition. In the scope of the group eye tracking (GET) platform in METU, a first study was conducted on group eye tracking based on a whack-a-mole game to investigate gaze behavior and the effect of gaze awareness in different group conditions (Deniz, 2016).

Although there are many models and versions of eye tracker devices, the basic technology is quite similar in most of them. The physics of eye tracking involves a light ray (usually infrared) to be radiated towards the eye. The radiated ray, then, is reflected once from the cornea and once from the retina. As the eye moves, the reflection from the cornea does not change direction; however, the reflection from the retina does. By calculating the difference between these two reflections, an eye tracker can detect where the eye is looking at (Figure 6 and Figure 7). There are layers of post analysis (etc. calibration) for the location of the gaze to be detected accurately (Nyström & Holmqvist, 2010).

[3] This company was purchased and shut down by Facebook in 2017. This may be conceived as an evidence of expected closer interest in eye tracking soon by large companies.

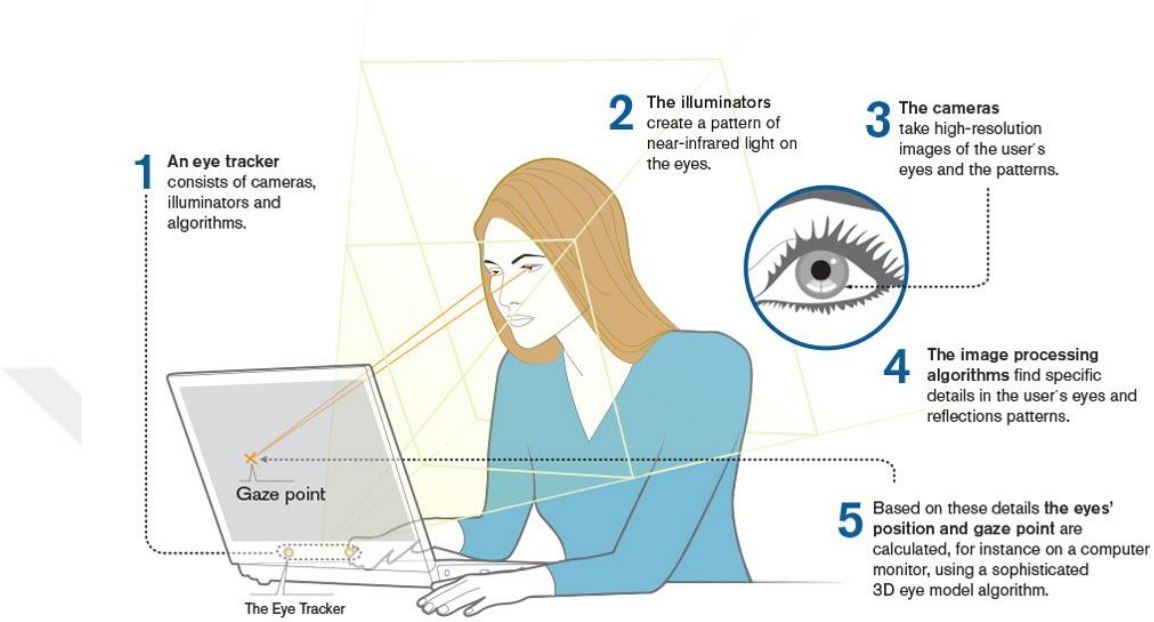


Figure 6 A subject in front of an eye tracker (https://www.tobii.com/group/about/this-is-eye-tracking)

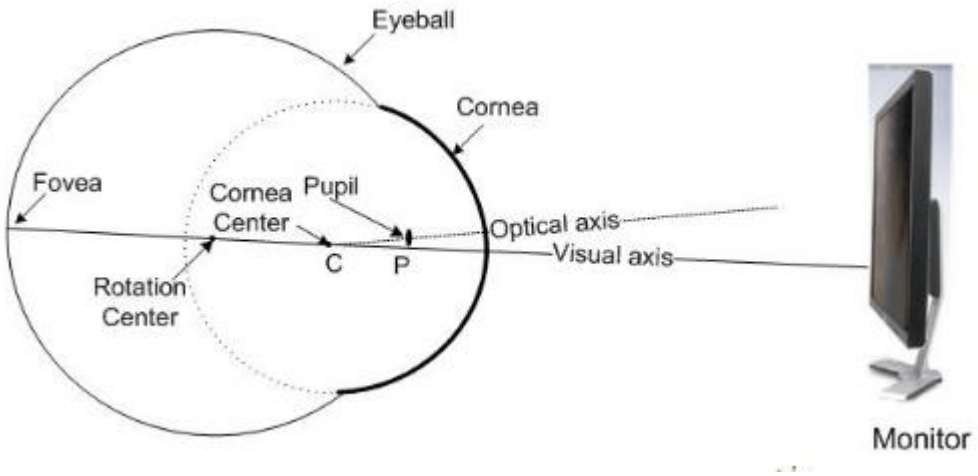


Figure 7 The difference between the infrared reflection from cornea and fovea help in calculating where the eye is gazing at (https://www.ecse.rpi.edu/~cvrl/zhiwei/gazetracking/gazetracking.html)

In this chapter, we made a summary of the relevant research literature along with an outline of the underlying theories needed for recurrence quantification analysis. In the next chapter we will see these theories implemented in a package for R programming

language along with a description of the software developed for collecting data in group eye tracking experiments.

CHAPTER 3

METHODOLOGY

In this section an overview of the structure and architecture of the Group Eye Tracking (GET) data collection framework, namely the *GETapp*, is presented. Also, the R package for cross-recurrence quantification analysis, viz. `generalRQA`, is described along with an explanation of the functions and algorithms used in it for data processing.

3.1. The Group Eye Tracking Platform (GETapp)

The *GETapp* platform is an application written by the author in the JavaScript language for data collection in group eye tracking experiments[4]. The GETapp was designed to be generic, meaning that it is not specific to the experiment in the current study. Experiment designers can feed their experiments as HTML documents to the application. The version of the application used in the current study is version 0.1.1. This is an early version of the application where experiment designers may need to tweak the source code slightly in order for their HTML-based experiment to work completely. Effort is being made to make the application more modular and user friendly in the upcoming versions so that experiment designers can feed their entire design as bundles of HTML, CSS and JavaScript codes (https://gitlab.com/GETters/GETapp.git).

3.1.1. The choice of language.

GETapp was written mainly in JavaScript. The rationale for this choice was that firstly, JavaScript is a programming language which is currently getting more and more widespread and powerful despite its potential for compatibility issues. The support and the community around it are comprehensive and the resources are plentiful recently. Secondly, the new Node.js server environment, a JavaScript run-time built on Chrome's V8 JavaScript engine, is scalable, which is important for the use case in this research and as will be pointed out subsequently, the server will have to work with many clients concurrently.

---

[4] The application has reached its functional state thanks to the efforts of Mine Cuneyitoglu Ozkul for her review, bug fixing, refactoring and checking robustness of the code.

Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. This is why it is mainly used for serving websites that have a high traffic load. The JavaScript language, contrary to many other programming languages, is a single threaded, asynchronous language, which means that it does not always execute a script from top to bottom. While an I/O operation or any process that can be carried out in a non-blocking way is being run, other functions which take much shorter time to terminate get executed. This is especially useful in the case of this research, because a stream of eye tracking data is constantly being transferred to the server, written to a file and broadcasted back to the clients while other functions must be executed.

3.1.2. The architecture.

The system is made up of three sets of physical components: The server computer (referred to as *server* from now on), the client computers (referred to as *client* from now on) and eye trackers. Each client computer has an eye tracking device connected to it which in turn is run by a server program on the client computer (Figure 8). This server program will be referred to as the eye tracker server from now on. While the platform is in use, eye tracking data are transferred from all the eye tracker servers directly to the server. The sever takes the streams of data and while recording them to a file, also broadcasts them back to all the clients at once. In this way each client receives all the data that other clients send to the server in real time with a small delay (as will be mentioned later) so that each participant can see, in real time, where the other participants are gazing on the display.

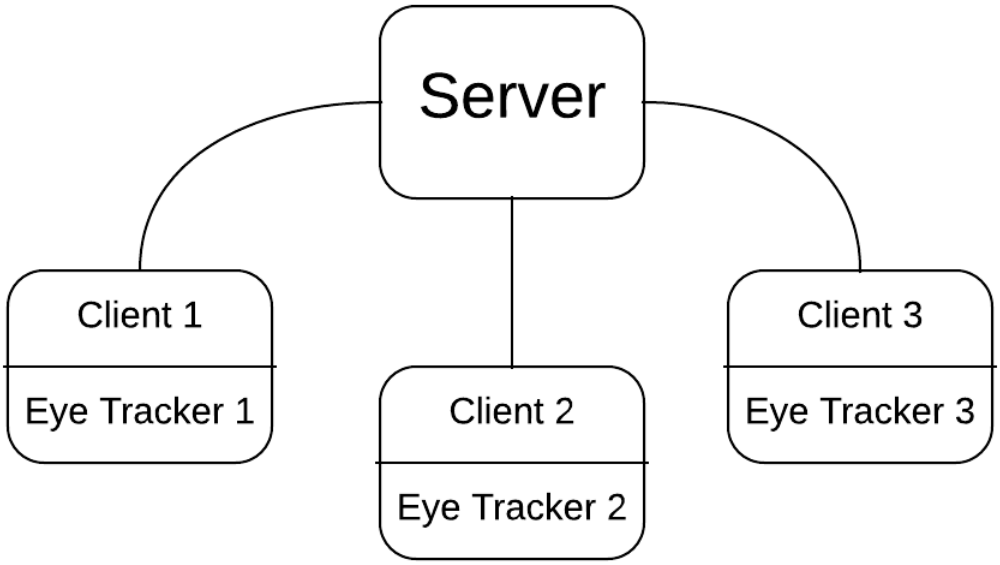


Figure 8 In this diagram 3 clients are connected to the server. One Eye Tracker is connected to each client and runs a server program on it.

3.1.3. Overview of the program.

The general case of the function envisioned for the GET platform consists of multiple screens designed by the experimenter. The particular form and function of a screen depends mostly on the experiment design. Usually there are screens where the participants perform a task, screens where participants make some choices and screens that inform the participants of some achievement, some event to come or some instruction.

Eye trackers need to be calibrated before the experiment begins. The EyeTribe eye tracker's server application has a utility to calibrate the device. This utility is used immediately before the experiment begins. During this calibration process, the participant is required to gaze at a circle until the circle moves to a new location on the screen at which point, the participant should relocate her gaze to a new location. The relocation can be configured to occur 9, 12 or 16 times (9 times in the current study). While the test taker gazes at a certain circle, the gaze positions are recorded and then compared to the position of the circle, from the deviance of which the device gets calibrated accordingly. Nevertheless, the device can lose its calibration after some time. It is not practical to stop the experiment, switch to the EyeTribe calibration utility and calibrate the device. Instead, GETapp implements a function for calibrating the device by using the EyeTribe's API. The function implements the EyeTribe's API, which takes commands in JSON format containing the start and stop commands and the position of the calibration point. This

function can be implemented anywhere in the screens or in its own screen. The GETapp's calibration function first performs a test in which the participant gazes at a target with a position predefined by the experiment designer on a screen. After a delay, the gaze positions are recorded for a certain amount of time. Subsequently, if the mean of the recorded gaze positions is closer than a threshold to the target's position and the standard deviation is under a certain threshold, the test is passed and the screen proceeds to the next state; however, if the test is not passed, a process of calibration, as mentioned above, will commence for that particular participant, while other participants are waiting. Once the device is calibrated, the experiment continues as usual. The parameters of the calibration process, namely the delay time, the recording time, the thresholds for the mean and standard deviation in the test stage and the positions of all the calibration circles in the calibration stage, are definable by the experiment designer.

Another component of the experiment in this platform is the visualization modality (previously mentioned as the "marker"). This visualization modality is often represented as a circle, a plus sign, a heat-map or any other custom-made marker. As mentioned before, the marker is to designate, for each participant, positions on the screen where other participants are looking at simultaneously [5]. These locations can be visualized individually, where one can see many markers moving on the screen, or collectively, where a marker represents the mean location of all the gaze locations. The marker can also, as in the case of the heat-map, depict the concentration of the other's gazes. One such visualization modality, which is available in GETapp, is the heat-map. This marker is in the form of a red, semi-transparent circle which is darker at the center and gets lighter and fades into the background, getting closer to the periphery. The center of the circle is at the mean of the gaze positions and its diameter corresponds to the standard deviation of the gaze positions from the mean. As the circle grows in size, its color gets lighter. The purpose of this design is to draw the participants' attention to a location where other participants are most concentrated.

[5] In the current study the visualization modality was designed and implemented by Mine Cuneyitoglu Ozkul.

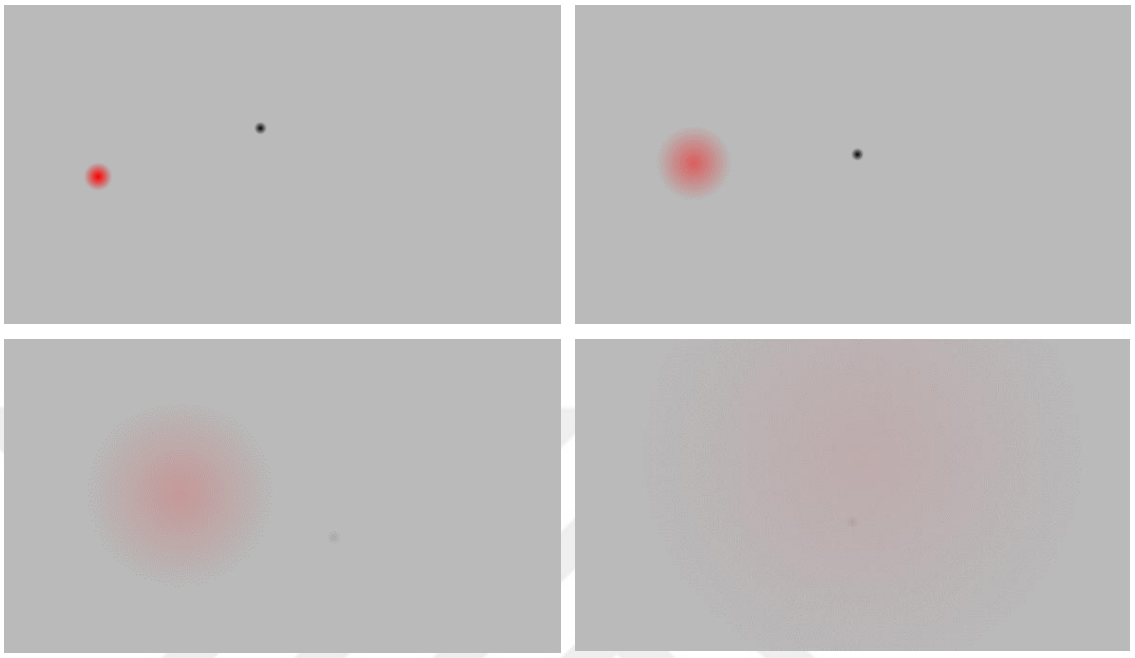


Figure 9 The *HeatMap* visualization modallity with various radii

The smoothing of visualization modalities' movement, the calibration process and the flow of screens are all controlled on the server side[6]. The server also records the data that came from the eye tracker servers (the server application that runs on client computers and manages the eye tracker device) and the clients to a file in JSON format. The experiment designer can specify multiple variables to be written in the data file. The most common ones are presented below:

- Participants' unique Ids: Each participant is assigned a unique ID to make them trackable.

- Screen number: Each experiment is made up of a set of screens that recur in some order during the experiment. For example, in the current experiment, there is a screen on which the participants should look at the symbol at the center, a screen where calibration takes place and a screen where the search

[6] This design choice was made, based on comparing performance when the mentioned operations are done on the server side vs. the client side. As the number of participants increases it may be better for some operations (e.g. smoothing the movement of visualization modality) to be done on the client side.

task is performed. The screen number is the number assigned to each of these screens.

- Question number: Each condition of the experiment is assigned a number called the question number. For example, in the current study, each search target with a specific configuration (the color and the position) is assigned a question number. This is to control how many times and in which order each condition appears.

- Group size: The number of participants performing a task.

- Visualization modality: The marker that depicts where the participants are looking at.

The client time stamp (cts), is the time stamp assigned to the data point by the client when the data leave the client, whereas the server time stamp (sts) is the time stamp that the server assigns to the data when received by the server. There may be cases where an experimenter would need cts; however, because it is difficult to synchronize client computers' clocks to match with a high enough precision for the purposes of the current study (about 10 milliseconds), it is better to use sts to avoid artifacts created by differences in client computers' clocks. Moreover, the difference between cts and sts may be the lag between the client and the server caused by the algorithms used (not by the network). Measuring the lag will be addressed in later sections.

3.2. The `generalRQA` Package for R

In this section the algorithms and functions in the `generalRQA` package for the R language will be presented. The package has been developed by the author and used for the analysis of the data in the present study. There exist other packages in the R community written for recurrence analysis, such as the `crqa` package (Coco, with contributions of James D. Dixon, & Nash, 2015); however, they have a major limitation for the purposes of the current study. The main limitation is that the available packages work with recurrence matrices of only two dimensions, whereas the goal in the present study is to analyze multiple time series concurrently. In other words, the recurrence matrices may be more than two dimensional in group eye tracking. As presented before in section 2.4, the mathematical description, from which the algorithms in this package were derived, are also defined on one (for the RQA case) and two time series (for the CRQA and JRQA cases) only (Webber Jr & Marwan, 2015). This is the reason that the mathematical descriptions had to be generalized to fit the multi-time-series situation, as well.

The information provided in this section is an integration of the individual function documentations in the package and the package vignette which provides a high-level description of how to use the package through an example. The `generalRQA` package

can be used to implement *Cross Recurrence Quantification Analysis (CRQA), Joint Recurrence Quantification Analysis (JRQA)* and *simple Recurrence Quantification Analysis (RQA).*

3.2.1. The workflow

The general workflow of CRQA involves embedding the data (section 2.3), creating distance matrix from the trajectories, creating recurrence matrix from the distance matrix, choosing which line or lines parallel to the main diagonal in the matrix to work on, creating line histograms for each one of the diagonal lines and calculating RQA measures for each one of them. In simple RQA, length histogram of the whole recurrence matrix is used for calculating recurrence measures, whereas in CRQA, one may need to investigate the length histogram of individual diagonal lines to analyze concurrent behavior among multiple time series (Norbert Marwan et al., 2007a). In this case, each diagonal line corresponds to a specific lag combination among the times series.

3.2.2. Hands on

The functions and algorithms in the package will be illustrated by going through calculating CRQA measures of two logistic maps with slightly different initial conditions. The logistic map is a difference equation which is a simple model of population growth. The model below shows exponential growth of a population, and also convergence to a certain population limit. If we define k as the carrying capacity, x_t as the current population we can write:

$$x_{t+1} = x_t + rx_t\left(1 - \frac{x_t}{k}\right) \tag{15}$$

where r is a coefficient defining how fast the population would grow were there no limitation to growth. We can confirm that the equation above satisfies the requirements of the model description; if the current population is near zero the equation will become $x_{t+1} = (1 + r)x_t$ which shows an exponential growth. If the current population size approaches the carrying capacity, the equation will become $x_{t+1} = x_t$ which stays constant (Niazi, 2016).

If we replace x with αx and simplify the equation we will have:

$$\begin{aligned}
\alpha x_{t+1} &= \alpha x_t + \alpha r x_t\left(1 - \frac{\alpha x_t}{k}\right) \\
&= x_t + r x_t\left(1 - \frac{\alpha x_t}{k}\right) \\
&= x_t\left(1 + r\left(1 - \frac{\alpha x_t}{k}\right)\right) \\
&= x_t\left(1 + r - \frac{r\alpha x_t}{k}\right) \\
&= (1 + r)x_t\left(1 - \frac{\alpha r x_t}{k(1 + r)}\right)
\end{aligned}$$

replacing α with $\frac{k(1+r)}{r}$ we will get:

$$x_{t+1} = (1 + r)x_t(1 - x_t)$$

and replacing $1 + r$ with r we will finally arrive at the below equation, called the logistic map:

$$x_{t+1} = rx_t(1 - x_t) \tag{16}$$

which is the same as (15) written in a more compact form by introducing new variables.

In this system, if for example $r = 2$, the asymptotic state of the system will converge to a certain value regardless of the initial condition. If r is set to 3.1, the asymptotic state of the system will oscillate between two values. Likewise, as r increases, in progressively smaller increments of r, the period of the oscillating asymptotic state will double. This cascade eventually leads to the divergence of the period to infinity at $r = 3.569946$. At this point the system loses any periodicity of finite length and its behavior seems random. This regime is called a chaotic regime (Mitchell, 2009).

A hallmark of chaotic regimes is their sensitivity to the initial conditions. This means that if two trajectories of the system start from two initial conditions, arbitrarily close to each other, they will diverge eventually. To showcase this phenomenon two trajectories of the logistic map with $r = 3.9$ and with two initial conditions $x_0 = 0.5$ and $x_0 = 0.500001$ are plotted in Figure 10.

The two trajectories start overlapping each other but after about 50 steps they begin to diverge; however, after diverging, the trajectories are not completely uncorrelated. They

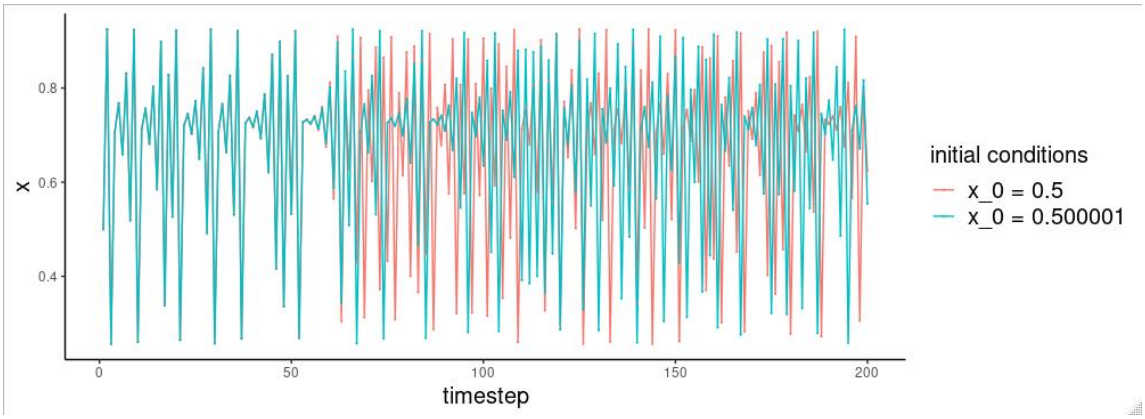


Figure 10 Two trajectories of the logistic map with slightly different initial conditions

show a pattern of recurrences. To investigate the recurrence pattern of these two trajectories we will use the CRQA method and showcase the use of the `generalCRQA` package.

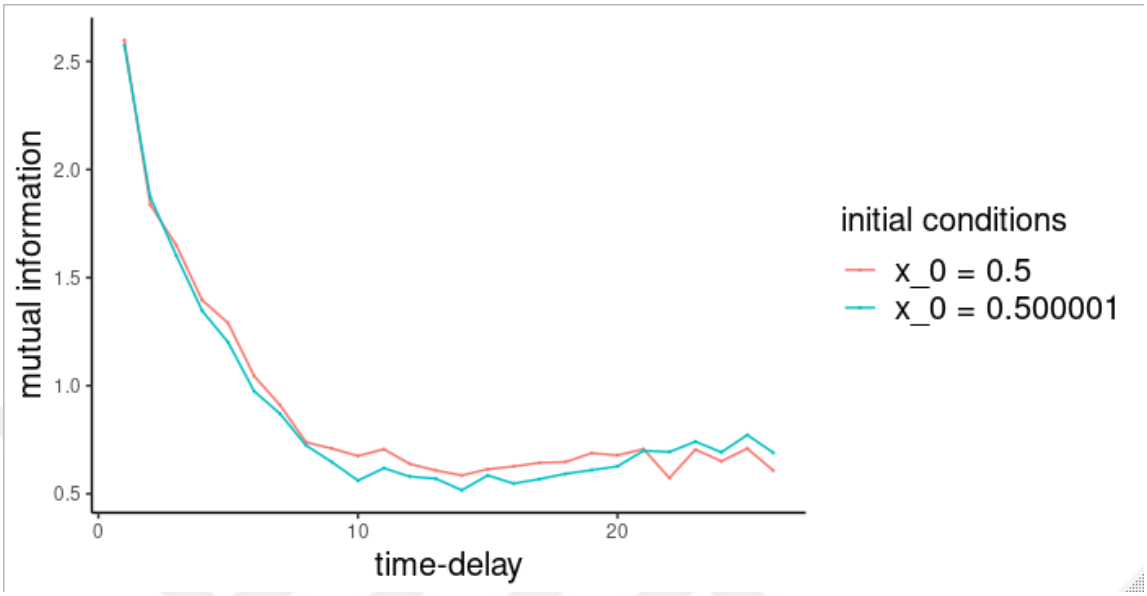


Figure 11 Mutual information diagram

The first step towards CRQA is to embed the data in a higher dimensional space (theory of embedding has been explained in 2.3 above). The two important parameters for embedding are the *time-delay* and the *embedding dimension*. The appropriate time-delay can be found using the *mutual information* formula which is implemented by the function `mutual` in the package `tseriesChaos` (Antonio & Narzo, 2013). Plotting the mutual information vs. different time-delay values (Figure 11), the optimal time-delay parameter is chosen as the first local minimum of the plot (Coco & Dale, 2016). This minimizes the dependencies between consecutive time steps in the embedded time series.

The local minimum occurs at a mutual information value of 10, in the case of the two time series from the logistic map (Figure 11). Having determined the appropriate time-delay, we can use that information to find the optimal embedding dimension. As mentioned in section 2.3, one method of accomplishing this is through calculating *false nearest neighbors*. False nearest neighbors can be calculated by the `false.nearest` function from `tseriesChaos` package. The function calculates the number of false nearest neighbors for the time-delay parameter input to the function and an array of different dimensions. Figure 12 shows the plot of the number of false nearest neighbors decreasing as the dimension increases. The optimum dimension is chosen as the dimension where the plot shows a rapid change in the number of false neighbors which can be detected as an elbowing in the plot (Chelidze, 2017).

In Figure 12 below the elbowing occurs at dimension 2. Having decided on the time-delay and the dimension parameters, the embedding can be realized by the `embed` function from the `generalRQA` package.

The `embed` function gets the time series, the time-delay and the dimension as parameters and outputs a data frame. Each column in the data frame corresponds to one dimension and each row corresponds to a state in the state space (Figure 12).

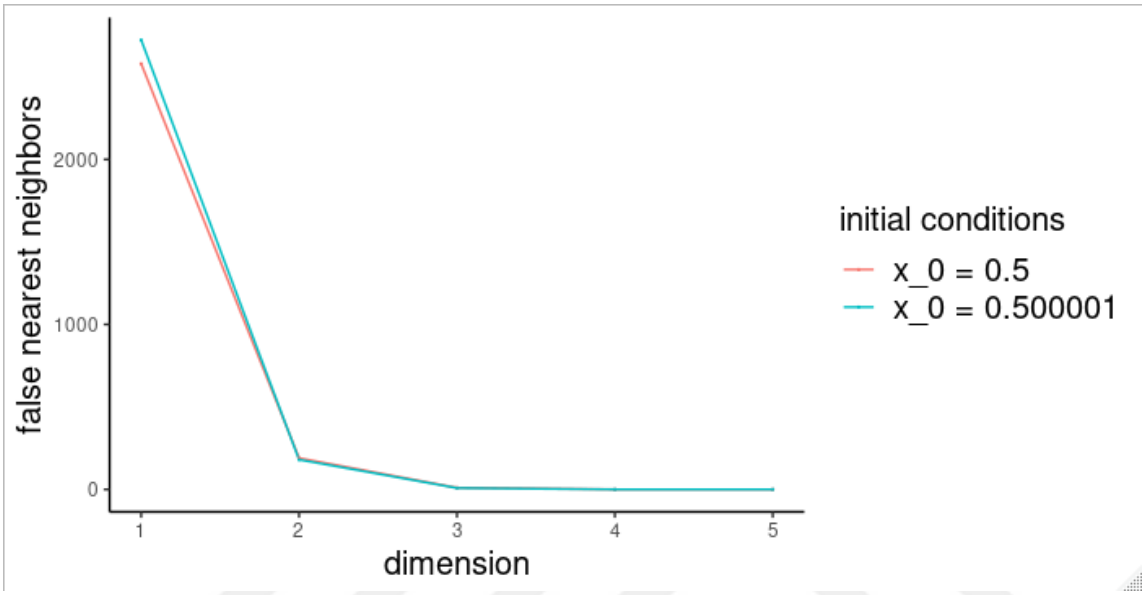


Figure 12 False nearest neighbors diagram

```
embed(timeSeries, delay, dimension){

      out ← list()

      lengthOfEmbeddedTimeSeries ← length(timeSeries) - (delay * (dimension - 1))

      strt ← 0

      for (i in 1 : dimension) {

            out[i] ← ts[(1 + (strt * delay)) : ((strt * delay) +
lengthOfEmbeddedTimeSeries)]

            strt ← strt + 1

      }

      return(columnBind(out))

}
```

Code 1 The function returns a data frame in which each column in the data frame corresponds to one dimension and each row corresponds to a state in the state space.

The steps so far included finding the optimum time-delay parameter and the optimum dimensionality of the state space. Using the time-delay and the dimension we created a time-delayed embedded version of the original time series. We will use these new time series to calculate a distance matrix. The distance matrix will contain the largest Euclidean distance between corresponding data points in the embedded time series. By enforcing a threshold on the distance matrix, it will be converted into a binary matrix called the recurrence matrix. As mentioned in section 2.4, the recurrence matrix identifies locations along the time series where a recurrence between the time series occurs by assigning value 1 to the corresponding cell in the matrix and 0 to all the cells. The recurrence matrix will be the base for calculating all the CRQA measures.

The time-delayed embedded time series can be fed into the `makeDistanceMatrix` function from `generalRQA` package to obtain a *distance matrix*. The format in which the matrices are represented in the `generalRQA` package is specific to the package and is designed to optimize speed of the calculations. The distance matrix is defined as a multi-dimensional matrix where each dimension corresponds to a time series. The value of each cell in the matrix designates the largest Euclidean distance between the data points in the embedded time series corresponding to the indices of that cell. For example, the cell designated by indices [2, 4, 6] in a three-dimensional matrix is the largest Euclidean distance between the 2^{nd} point in the first embedded time series, the 4^{th} point in the second embedded time series and the 6^{th} point in the third embedded time series. However, the matrix in the package is defined as a data frame where each column contains a diagonal of the multi-dimensional matrix (A diagonal here means a line in the matrix grid parallel to the main diagonal). The names of the columns correspond to the starting indices of that diagonal. For example, the diagonal that starts from index [1, 1, 1] in a three-dimensional matrix (which is the main diagonal) corresponds to the column with the name *1.1.1* and the diagonal that starts from index [1, 3, 2] corresponds to the column with the name *1.3.2*. Because the length of the diagonals in a matrix are different, to make a data frame out of the diagonals, zeroes are appended to the beginning of the columns of the data frame to make their lengths even. Figure 13 top, shows an ordinary two-dimensional matrix with values 1 to 25 arranged column-wise. Under it we can see the representation of the same matrix in the specific format of the `generalRQA` package.

	1	2	3	4	5
1	1	6	11	16	21
2	2	7	12	17	22
3	3	8	13	18	23
4	4	9	14	19	24
5	5	10	15	20	25

	5.1	4.1	3.1	2.1	1.1	1.2	1.3	1.4	1.5
1	0	0	0	0	1	0	0	0	0
2	0	0	0	2	7	6	0	0	0
3	0	0	3	8	13	12	11	0	0
4	0	4	9	14	19	18	17	16	0
5	5	10	15	20	25	24	23	22	21

Figure 13 Top: an ordinary two-dimensional matrix with values 1 to 25 arranged column-wise. Bottom: the same matrix in the specific format of the `generalRQA` package.

In the `generalRQA` package, the `makeDistanceMatrix` function takes a list of time series and a parameter *k* as input and returns a recurrence matrix (Code 3). *k* specifies the distance from the main diagonal in which the diagonals will be considered. For example, if the matrix is two dimensional (two time series) and $k = 2$, only the *1.1*, *1.2* and *2.1* diagonals will be calculated. For a matrix of dimension three and $k = 3$, diagonals *1.1.1*, *1.1.2*, *1.2.2*, *1.2.1*, *2.2.1*, *2.1.1*, *2.1.2*, *1.1.3*, *1.2.3*, *1.3.3*, *1.3.2*, *1.3.1*, *2.3.1*, *3.3.1*, *3.2.1*, *3.1.1*, *3.1.2*, *3.1.3*, *2.1.3* are the diagonals that will be calculated. In general, the number of diagonals considered given the dimension and k is $k^{dimension} - (k - 1)^{dimension}$.

The `makeDistanceMatrix` calls `prepareIndex` function which takes dimension, k and a vector containing the lengths of each dimension of the matrix (dimensionLengths) as input. The function returns a matrix of indices (Code 2). Using the indecies matrix the `makeDistanceMatrix` function will choose corresponding data points from the input time series to calculate the Euclidian distance.

```
prepareIndex(dimension, k, dimensionLengths){
     tmp ← list()
     for(i in dimensionLengths){
            append(tmp, 1 : i)
     }
     indeces ← all the element vise combinations of vectors in tmp
     indeces ← keep only the elements in indeces which have at least one element 1
     indeces ← a data frame where each column is one element of indeces
     while(maximum(last row of indeces) < min(dimensionLengths)){
            append(indeces, last row of indeces + 1)
     }
     return(indeces)
}
```

Code 2 The function prepares a matrix of indices from which the makeDistanceMatrix will choose corresponding data points from the input time series to calculate the Euclidian distance.

```
makeDistanceMatrix(listOfTimeSeries, k){
     index ← prepareIndex(length(listOfTimeSeries), k, lengthsOfTimeSeries)
     tmp ← vector()
     for(i in index){
            tmp ← vector()
            for(j in rows(i)){
                  positions ← vector()
                  for(p in 1 : length(j)){
                        append(positions, listOfTimeSeries[p][j[i]]
                  }
                  append(tmp, maximum(dist(p)))
            }
            i ← tmp
     }
     return(index)
}
```

Code 3 The function takes a list of time series and a parameter *k* as input and returns a recurrence matrix. The `dist` function above calculates the Euclidean distances between all elements of its input.

The `makeDistanceMatrix` function, thus, returns a matrix in the specific format of the `generalRQA` package where each value is the largest distance between data points from the input time series corresponding to the indices of that value.

The next step is to convert the distances in the previous matrix into binary values. This task is accomplished by the `makeRecurrenceMatrix` function (see Code 7). The critical parameter in this function, as pointed out in section 2.4, is the threshold. There are two ways to define the threshold parameter. The first is to directly input the threshold by setting the *radius* parameter in the function. The second way is to specify a range for the recurrence rate (section 2.4), which is the ratio of recurrent points to all the points in the matrix, so that the function calculates a threshold with which the recurrence rate falls into

that range. The `makeRecurrenceMatrix` takes the *distanceMatrix, dimension, maxR, range* (see the next paragraph for a description of maxR and range) and *radius* as inputs and outputs a recurrence rate, a radius (the threshold) and a recurrence matrix. The returned recurrence matrix is in the same format as the distance matrix except that binary values replace distances in the matrix.

If *maxR* and *range* are specified in the parameters, the function will do a binary search in the domain 0 to maxR to find a threshold so that the recurrence rate falls into the *range*. However, if the *radius* is specified in the parameters, it will be directly regarded as the threshold.

The `makeRecurrenceMatrix` incorporate three other functions. The function `recurrenceFromDistance`, given a threshold value and a distance matrix, simply converts the distance matrix to a binary format (Code 4). It replaces distance values less than the threshold with 1 and distance values more than the threshold with 0. The function `radiusFinder` takes the distanceMatrix and two parameters *maxR* and *range* and as mentioned in the previous paragraph, searches for a *radius* upon which the recurrence rate falls into the *range* (Code 5). `radiusFinder` returns the recurrence rate (so that there is no need to calculate it again), the found radius (the threshold) and the recurrence matrix as output. The function `findRR` returns the recurrence rate given the recurrence matrix as input by dividing the number of 1's in the recurrence matrix with the size of the matrix (Code 6).

```
recurrenceFromDistance(distanceMatrix, radius){
      for(i in distanceMatrix){
            if(i$distance < radius){
                  i$recurrence ← 1
            } else {
                  i$recurrence ← 0
            }
      }
 }
```

Code 4 The function converts a distance matrix to a binary matrix by replacing the distance values with ones and zeroes based on the threshold value.

```
radiusFinder(distanceMatrix, maxR, range){
     rad = maxR
     maximumR = maxR
     minimumR = 0
     rMtrx = recurrenceFromDistance(distanceMatrix, rad)
     RR = findRR(rMtrx)
     while ((RR < range[1]) | (RR > range[2])) {
            if((((maximumR + minimumR)/2)-rad) < 0.001){
                        break
            }
            rad = (maximumR + minimumR)/2
            rMtrx = recurrenceFromDistance2(distanceMatrix, rad)
            RR = findRR2(rMtrx)
            if(RR > range[2]){
                        maximumR = rad
            }else if(RR < range[1]){
                        minimumR = rad
            }
     }
     return(list(rad, RR, rMtrx))
}
```

Code 5 The function searches for a radius (threshold) between 0 and maxR upon which the recurrence rate falls into the range.

```
findRR(recurrenceMatrix) {

     RR <- (number of 1's in the recurrenceMatrix)/(number of cells in the
recurrenceMatrix)

     return(RR)

}
```

Code 6 The function calculates the recurrence rate given the recurrence matrix.

```
makeRecurrenceMatrix(distanceMatrix, dimension, maxR, range, radius){
     if(radius){
            tmp = recurrenceFromDistance(distanceMatrix, radius)
            RR = findRR(tmp)
            return(list(tmp, RR, radius))
     } else {
            tmp = radiusFinder(distanceMatrix, maxR, range)
            return(list(tmp$recurrenceMatrix, tmp$RR, tmp$radius))
     }
}
```

Code 7 The function returns the recurrence matrix, the recurrence rate and the radius (threshold).

At this point one CRQA measure, the recurrence rate (RR) is available. To calculate the other measures, a *line histogram* is needed on the diagonal that the measures are to be calculated from. A line histogram in this case is a data structure specifying the number of lines with different lengths that can be found on the particular diagonal. A *line* here means any succession of 1's in the recurrence matrix in the direction of the main diagonal and with a minimum length which is specified by the `minLength` parameter.

The line histogram can be computed by the `findLengthHistogram` function (Code 8). The function takes a diagonal from the *recurrenceMatrix* and a minimum line length (which is 2 by default) as input. The output of the function is a data frame in which the first column specifies the length of the line and the second column specifies the frequency of that line occurring.

```
findLengthHistogram(diagonalOfrecurrenceMatrix, minLength ← 2){
     appendAtBeggining(diagonalOfRecurrenceMatrix, 0)
     appendAtEnd(diagonalOfRecurrenceMatrix, 0)
     tmp ← indeces(which(diagonalOfRecurrenceMatrix = 0))
     tmp ← (subtract each element in tmp from the next element) – 1
     tmp ← tmp < minLength
     return(count(tmp))
}
```

Code 8 The function returns a histogram of line lengths given a diagonal of the recurrence matrix. A line here means any succession of 1's in the direction of the main diagonal of the recurrence matrix with the minimum length given by the `minLength` parameter.

The measure *DET* is calculated by the function `findDET` (Code 9). The function takes recurrence rate, length histogram and recurrence matrix as input and outputs a number as *DET*.

```
findDET(RR, lengthHistogram, recurrenceMatrix){
     nrows ← vector()
     for(i in recurrenceMatrix){
            append(rowNumber(i))
     }
     nrows ← sum(nrows)
     DET ← (sum(lengthHistogram$lengths * lengthHistogram$frequency))/RR
     return(DET)
}
```

Code 9 The function calculates the DET measure as mentioned in section 2.4

RATIO is the ratio of DET over RR. It is calculated by the `findRATIO` function (Code 10).

```
findRATIO(RR, DET){
    return(DET/RR)
}
```

Code 10 The function calculates the RATIO measure by dividing DET with RR.

Dmax is the length of the longest line in the length histogram. It is found by the `findDmax` function (Code 11).

```
findDmax(lengthHistogram){
    return(maximum(lengthHistogram$lengths))
}
```

Code 11 The function finds the longest line in the length histogram.

AvD is the average length of lines in the length histogram calculated by the `findAvD` function (Code 12).

```
findAvD(lengthHistogram){
    return(sum(lengthHistogram$lengths *
lengthHistogram$frequency)/sum(lengthHistogram$frequency))
}
```

Code 12 The function finds the average length of all the lines in the length histogram.

The last measure, the *Shannon entropy* is calculated by the `findShannonEntropy` function (Code 14). The mathematics of this measure is explained in section 2.4. The `findShannonEntropy` function incorporates the function `probL` (Code 13). The `probL` function takes the length histogram and a length value (as parameter *l*) as input and simply returns the frequency value corresponding to that line length.

```
probL(lengthHistogram, l){
    return(lengthHistogram[which(lengths = l)]$frequency)
}
```

Code 13 The function returns the corresponding frequency value for the input line length.

```
findShannonEntropy(lengthHistogram){
    tmp = 0
    for(i in 1 : numberOfRows(lengthHistogram)){
          tmp ← tmp + probL(lengthHistogram, lengthHistogram.row(i)$lengths) *
log(probL(lengthHistogram, lengthHistogram.row(i)$lengths)
    }
    return(tmp)
}
```

Code 14 The function calculates the Shannon Entropy based on the mathematical description of the measure in section 2.4.

Returning to the logistic map example, because there are two time series and hence the dimension of the recurrence matrix is 2, it is possible to visualize the matrix (Figure

14). In this case, after embedding, *k* in `makeDistanceMatrix` has been chosen to cover the whole matrix (all the diagonals). In the `makeRecurrenceMatrix` the parameters are maxR = 5 and range = [0.03, 0.05].

Measures are as follows:

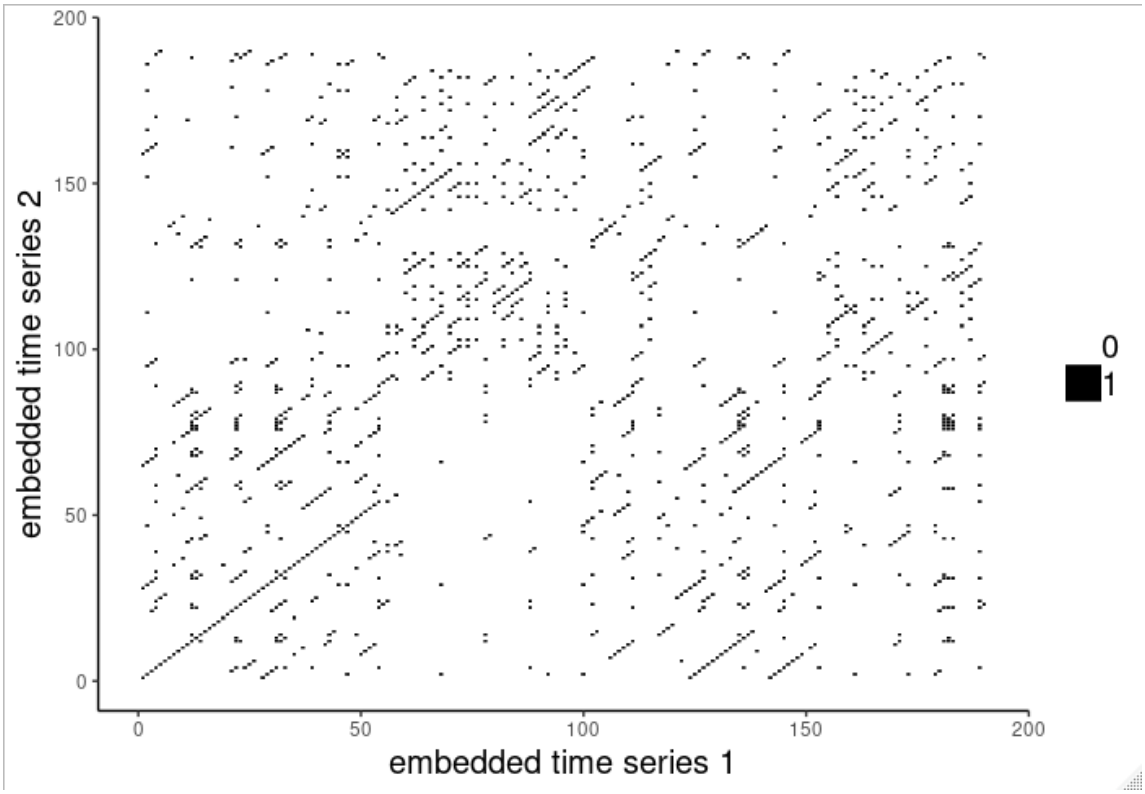


Figure 14 cross recurrence plot of the two trajectories of the logistic map with r = 3.7 and initial conditions 0.5 and 0.50001

RR = 0.03601108

DET = 0.5730769

RATIO = 15.91391

Dmax = 54

AvD = 3.497653

ENT = -1.505847

In this chapter we presented a description of the software tool, the GETapp, developed for data collection in group eye tracking. We gave explanation of the structure of the software system and various design choices in the development of the software. We also

provided an explanation of the `generalRQA` R package. We illustrated how we have implemented the methods described in Chapter 2 in program and showed the logic of the functions by providing pseudo-codes. We also used the `generalRQA` package to calculate CRQA measures on an example data set. In the next chapter we will go through the same process to calculate CRQA measures for the real data from the experiment.

CHAPTER 4

EXPERIMENT

4.1.Experiment Setup

The experiment was made in two sessions. In the first session, six groups of three participants and then in the second session two groups of five participants (19 male 9 female). All the participants were students at METU. The participants were admitted into the room in groups and each participant was seated in front of a separate computer. After calibrating the eye tracking devices and after the instructions, the participants were asked not to talk to each other during the experiment. When all the players passed the calibration phase they proceeded with the next screen. Otherwise, the ones who failed the calibration was presented the calibration screen again and the others waited until all the participants finish calibration.

In the next screen, participants were *to look for a small circle* (i.e., to conduct visual search). The color of the circle was an experimental factor with three conditions. It was either *black* (thus relatively easy to find), *gray* (which was close to the background color thus relatively difficult to find), or there was *no circle* on the screen (but participants did not know that and they were asked to actively search for a circle on the screen).

The other experimental factor was *visualization modality* with two conditions. There was either *no visualization* of the other participants on the screen, or there was a *heat-map*, which was basically a visual representation of the other participants during the course of their visual search, as described in the previous sections.

The participants were instructed to click the left mouse button (it did not matter where they clicked) once they found the circle. When all the participants clicked, they proceeded with the next screen, which was again the calibration test screen. The session was composed of 23 trials as such. The initial three trials were warm ups and the rest displayed the three experimental conditions with equal chance in random order.

4.2. Analysis Without Embedding

The data were recorded by *GETapp* in JSON format and imported as data-frame into R. The function `importAndClean(file)` in the `generalRQA` package automates this process. It reads the data from the file, makes it into a data-frame format and checks for data integrity where it deletes rows having NA, Nan, Inf or -Inf values.

The analysis was first conducted on the x coordinates of the eye tracking data. The radius in this case was chosen to be 30 pixels, which corresponds to approximately 4 degrees of visual angle given the current experiment setting. After creating distance matrices from the time series and acquiring recurrence matrices from the distance matrices, the recurrence rates were obtained. After calculating the line histograms from the recurrence matrices, the DET and ENT measures were calculated (these steps were outlined in section 3.2.2). The plots below visualize the RR (Figure 15), DET (Figure 16) and ENT (Figure 17) measures for three conditions (*Black*, *Gray* and *None* circle) and the two conditions *of heat-map modality* and *no heat-map modality* for *groups of 3*. It should be pointed out that in all our analyses, the *group* factor has not been considered. This is to say that each group has performed multiple trials of various target point and visualization modality conditions, but each one of these trials is considered independently. In the plots

below the error bars correspond to standard error.

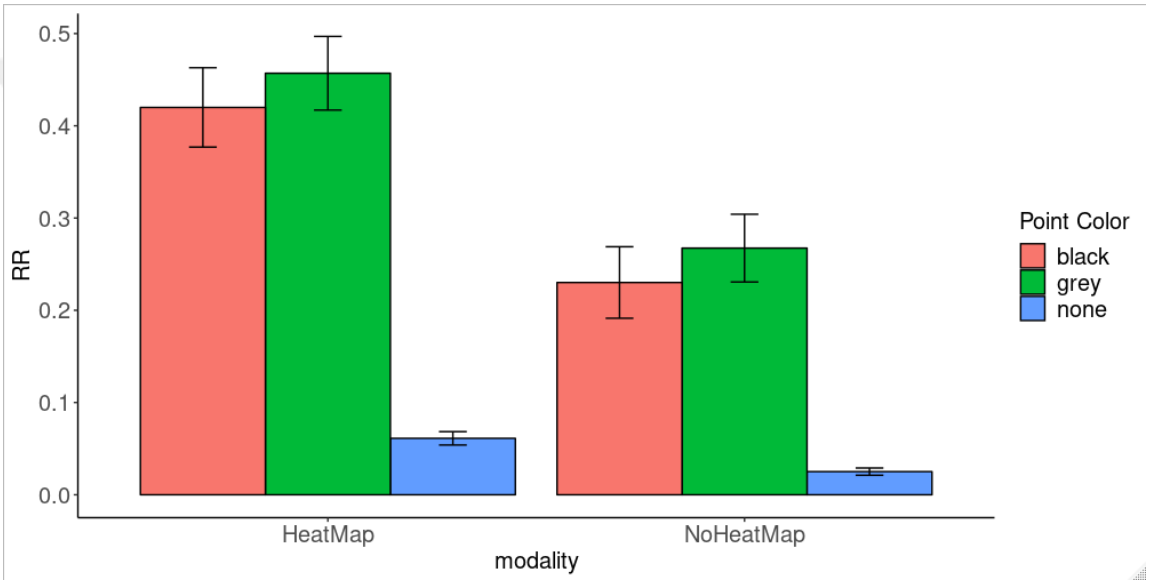


Figure 15 RR for HeatMap vs NoHeatMap conditions and three point colors

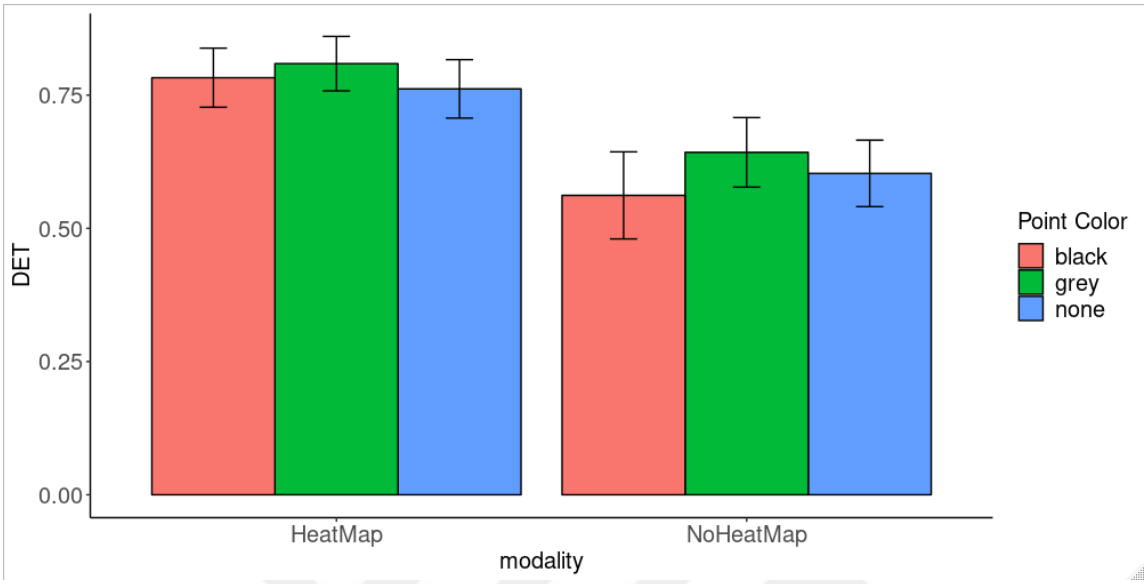


Figure 16 DET for HeatMap vs NoHeatMap conditions and three point colors

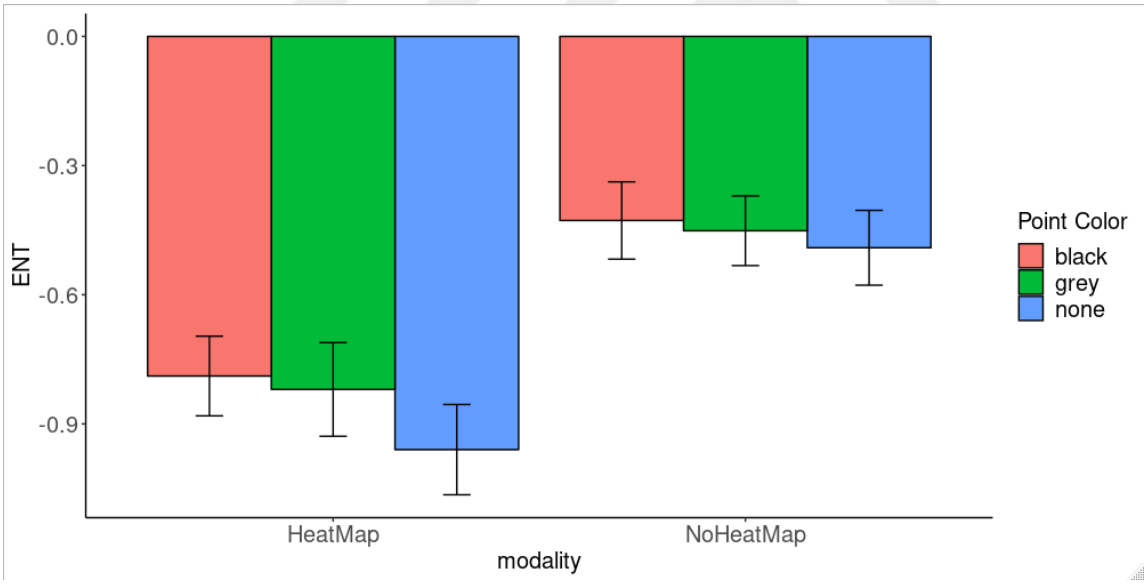


Figure 17 ENT for HeatMap vs NoHeatMap conditions and three point colors

4.3. Statistical Tests

4.3.1. Groups of 3, x axis

A two-way ANOVA test was conducted on the data for groups of 3 for RR (recurrence rate), DET (percent determinism) and ENT (Shannon entropy) measures. The design was a 2*3 unbalanced design (Table 1). The inputs of the ANOVA test were the x-axis values of the gaze location in three conditions of the point to be found (*black*, *grey* and *none*) and two conditions of the visualization modality (*HeatMap* and *NoHeatMap*). We assumed that the x values of the gaze locations were enough to study the recurrences in group eye tracking. The experiment data for 6 groups of three participants consisted of 41 trials of *black point with heat-map modality*, 36 trials of *grey point with heat-map modality*, 36 trials of *no point with heat-maomodality*, 32 trials of *black point with no heat-map modality*, 44 trials of *grey point with no heat-map modality* and 43 trials of *no point with no heat-map modality* (see Table 1).

Table 1 The ANOVA unbalanced design for groups of 3

	black	grey	none
HeatMap	41	36	36
NoHeatMap	32	44	43

For the RR measure the ANOVA test results are shown in Table 2.

Table 2 ANOVA results for RR for groups of 3, x axis

```
Anova Table (Type II tests)

Response: RR
                    Sum Sq  Df F value    Pr(>F)
pointColor          4.6282   2 56.9528 < 2.2e-16 ***
modality            1.0802   1 26.5846 5.504e-07 ***
pointColor:modality 0.3044   2  3.7458    0.0251 *
Residuals           9.1829 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 2 shows that the *pointColor* are associated with *significantly* different *recurrence rate*s (p-value << 0.001) and *levels of modality* are also associated with *significant* different *recurrence rates* (p-value < 0.001). The p-value for the interaction between *pointColor* and *modality* is 0.02 (significant), which indicates that the relationship between *modality* and *recurrence rate* depends on *pointColor*, too. This means that the participants exhibited different gaze distribution under the three conditions (*black*, *gray*, *none*) and they were also influenced by the type of visualization modality

(*HeatMap* vs. *NoHeatMap*) in groups of three. Follow-up pairwise t-test was conducted on the RR data. The results are shown in Table 3.

Table 3 Pairwise t-test for three point colors

```
Pairwise comparisons using t tests with pooled SD

data:  tmp$RR and tmp$pointColor

     black   grey
grey 0.65    -
none 4.6e-15 < 2e-16

P value adjustment method: BH
```

Table 3 shows that the *significant* difference *in recurrence rates* between *point colors* is mainly between *Black* and *None* and between *Gray* and *None* conditions but *not* between *Black* and *Gray* conditions.

In summary, the three conditions of the target point (*black*, *grey* and *none*) and two conditions of the visualization modality (*HeatMap* and *NoHeatMap*) had significant effect on the number of recurrent states of the three time series. In the case of the target point conditions, *black* vs. *none* and *black* vs. *none* were significantly different in RR; however, *black* vs. *grey* conditions showed no significant difference in recurrence rate.

For the DET measure the ANOVA test results are shown in Table 4.

Table 4 ANOVA results for DET fir groups of 3, x axis

```
Anova Table (Type II tests)

Response: DET
                    Sum Sq  Df F value    Pr(>F)
pointColor           0.120   2  0.4002 0.6706804
modality             1.876   1 12.5299 0.0004862 ***
pointColor:modality  0.043   2  0.1422 0.8675675
Residuals           33.831 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 4 suggests that *pointColor* has no significant effect on the *DET values* in the data; however, *modality* is associated with *significantly* different *DET values* between the *HeatMap* and *NoHeatMap* conditions (p-value < 0.05). The effect of interaction between *pointColor* and *modality* is not significant which means that the relation between *DET* and *modality* is independent of *pointColor*. This means that the pattern of synchronization between the eye gazes of participants (pattern of group formations) showed different

levels of predictability (determinism) between *HeatMap* and *NoHeatMap* conditions of the visualization modality but not between target point colors.

In summary, visualization modality had a significant effect on the DET measure but the effect of target point color on the DET measure was insignificant. In other words, if one were to use some method of prediction of the group formations of group eye tracking in a search task, the values of the prediction error probability would be significantly different for the two conditions of *HeatMap* vs. *NoHeatMap.*

For the ENT measure the ANOVA test results are shown in Table 5.

Table 5 ANOVA results for ENT for groups of 3, x axis

```
Anova Table (Type II tests)

Response: ENT
                   Sum Sq  Df F value    Pr(>F)
pointColor          0.557   2  0.8198    0.4418
modality            9.202   1 27.0772 4.383e-07 ***
pointColor:modality 0.140   2  0.2053    0.8145
Residuals          76.802 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 5 suggests that *pointColor* has no significant effect on the *ENT values* in the data; however, *modality* is associated with *significantly* different *ENT values* between the *HeatMap* and *NoHeatMap* conditions (p-value < 0.05). The interaction between *pointColor* and *modality* is not significant meaning that the relation between *modality* and *ENT* is independent from *pointColor*. This means that the pattern of synchronization between the eye gazes of participants (pattern of group formations) showed different levels of complexity between *HeatMap* and *NoHeatMap* conditions of the visualization modality but not between target point colors.

In summary, the target point color (*black*, *grey* and *none*) had no significant effect on the ENT measure but the ENT measure for different visualization modality conditions (*HeatMap* and *NoHeatMap*) were significantly different.

For Dmax measure, the ANOVA results are shown in Table 6 below:

Table 6 ANOVA results for Dmax for groups of 3, x axis

```
Anova Table (Type II tests)

Response: Dmax
                     Sum Sq  Df F value    Pr(>F)
pointColor             1410   2  4.3062   0.01461 *
modality               5602   1 34.2140 1.718e-08 ***
pointColor:modality     442   2  1.3495   0.26145
Residuals             37002 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 6 shows that the *pointColor* are associated with *significantly* different *Dmax* (p-value < 0.05) and *levels of modality* are also associated with *significant* different *Dmax* (p-value < 0.001). This means that the trajectories of participants' eye movements exhibited different divergence rates from each other under the three conditions (*black*, *gray*, *none*) and also different divergence rates from each other under two conditions of the visualization modality (*HeatMap* vs. *NoHeatMap*) in groups of three. Follow-up pairwise t-test was conducted on the Dmax data. The results are shown in Table 7.

Table 7 Pair-wise t-test results for three conditions of the target point

```
Pairwise comparisons using t tests with pooled SD

data:  tmp$Dmax and tmp$pointColor

     black grey
grey 0.10  -
none 0.16  0.60

P value adjustment method: BH
```

Table 7 suggests that the divergence rate between trajectories of the participants' eye movements were significantly different between *black* vs. *none* and *black* vs. *grey* but not between *grey* vs. *none*.

For the AvD measure the ANNOVA test results are shown in Table 8 below:

Table 8 ANOVA results for AvD for groups of 3, x axis

```
Anova Table (Type II tests)

Response: AvD
                     Sum Sq  Df F value    Pr(>F)
pointColor            184.0   2  1.7029    0.1845
modality              991.1   1 18.3492 2.721e-05 ***
pointColor:modality    17.6   2  0.1625    0.8501
Residuals           12207.5 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 8 suggests that *pointColor* has no significant effect on the *AvD values* in the data; however, *modality* is associated with *significantly* different *AvD values* between the *HeatMap* and *NoHeatMap* conditions (p-value < 0.001). The interaction between *pointColor* and *modality* is not significant meaning that the relation between *modality* and *ENT* is independent from *pointColor*. The *AvD* is another parameter for measuring the divergence of the trajectories. This means that the trajectories of participants' eye movements showed different rates of divergence from each other under two conditions of the visualization modality (*HeatMap* and *NoHeatMap*) but not between target point colors.

4.3.2. Groups of 5, x axis

Two group of five participants took place in the experiment. A two-way ANOVA test was conducted on the data for groups of 5 for RR, DET and ENT measures, similar to the analyses reported above for group of three participants. The design was a 2*3 unbalanced design (Table 9). The experiment data for two groups of 5 participants consisted of 13 trials of *black point with heat-map modality*, 10 trials of *grey point with heat-map modality*, 13 trials of *no point with heat-map modality*, 14 trials of *black point with no heat-map modality*, 12 trials of *grey point with no heat-map modality* and 11 trials of *no point with no heat-map modality* (see Table 9).

Table 9 The ANOVA unbalanced design for groups of 5

	black	grey	none
HeatMap	13	10	13
NoHeatMap	14	12	11

For the RR measure the ANOVA test results are shown in Table 10 below:

Table 10 ANOVA results for RR for groups of 5, x axis

```
Anova Table (Type II tests)

Response: RR
                     Sum Sq Df F value Pr(>F)
pointColor          0.001057  2  0.6711 0.5145
modality            0.002004  1  2.5442 0.1154
pointColor:modality 0.001235  2  0.7839 0.4608
Residuals           0.052774 67
```

It can be concluded from Table 10 that neither the *pointColor*, nor the *levels of modality* have significant effect on recurrence rates.

For the DET measure the ANOVA test results are shown in Table 11 below:

Table 11 ANOVA results for DET for groups of 5, x axis

```
Anova Table (Type II tests)

Response: DET
                    Sum Sq Df F value Pr(>F)
pointColor          0.0698  2  0.5007 0.6084
modality            0.1289  1  1.8499 0.1784
pointColor:modality 0.1583  2  1.1352 0.3275
Residuals           4.6701 67
```

Table 11 suggests that neither the *pointColor*, nor the *levels of modality* have significant effect on recurrence rates.

For the ENT measure the ANOVA test results are shown in Table 12 below:

Table 12 ANOVA results for ENT for groups of 5, x axis

```
Anova Table (Type II tests)

Response: ENT
                     Sum Sq Df F value Pr(>F)
pointColor          0.03415  2  0.5928 0.5556
modality            0.07500  1  2.6036 0.1113
pointColor:modality 0.03681  2  0.6390 0.5310
Residuals           1.93000 67
```

It can be concluded from Table 12 that neither the *pointColor*, nor the *levels of modality* had significant effect on the *ENT* measure.

For the Dmax measure, the ANOVA test results are shown in Table 13 below:

Table 13 ANOVA results for Dmax for groups of 5, x axis

```
Anova Table (Type II tests)

Response: Dmax
                    Sum Sq Df F value Pr(>F)
pointColor           0.657  2  0.2648 0.7681
modality             1.326  1  1.0682 0.3051
pointColor:modality  4.260  2  1.7160 0.1876
Residuals           83.164 67
```

It can be concluded from Table 13 that neither the *pointColor*, nor the *levels of modality* had significant effect on the *Dmax* measure.

For the AvD measure, the ANOVA test results are shown in Table 14 below:

Table 14 ANOVA results for AvD fir groups of 5, x axis

```
Anova Table (Type II tests)

Response: AvD
                    Sum Sq Df F value Pr(>F)
pointColor           0.506  2  0.3561 0.7017
modality             0.449  1  0.6325 0.4293
pointColor:modality  2.840  2  1.9991 0.1435
Residuals           47.587 67
```

It can be concluded from Table 14 that neither the *pointColor*, nor the *levels of modality* had significant effect on the *AvD* measure

In summary, in groups of 5 participants, no condition had any significant effect on the *recurrence rate (RR)*, the *percent determinism (DET)*, the *Shannon entropy (ENT)*, the *length of the longest diagonal line (Dmax)* or the *average length of diagonal lines (AvD)* measures.

4.3.3. Groups of 3, y axis

The Anova analysis results for the y axis data were the same as for the x axis data for both groups of three and 5 participants, as shown below.

Two-way Anova on the y axis data for groups of three participants on the *recurrence rate (RR)*,

Table 15 ANOVA results for RR for groups of 3, y axis

```
Anova Table (Type II tests)

Response: RR
                    Sum Sq  Df F value    Pr(>F)
pointColor          1.1378   2 25.2687 1.248e-10 ***
modality            0.1073   1  4.7662   0.03005 *
pointColor:modality 0.0027   2  0.0589   0.94284
Residuals           5.0884 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 15 shows that the *pointColor* are associated with *significantly* different *recurrence rate*s (p-value << 0.001) and *levels of modality* are also associated with *significant* different *recurrence rates* (p-value < 0.05). This means that the participants exhibited different gaze distribution under the three conditions (*black*, *gray*, *none*) and they were also influenced by the type of visualization modality (*HeatMap* vs. *NoHeatMap*) in groups of three. Follow-up pairwise t-test was conducted on the RR data. The results are shown in Table 16.

Table 16 Pair-wise t-test for three point colors

```
Pairwise comparisons using t tests with pooled SD

data:  tmp$RR and tmp$pointColor

     black   grey
grey 0.59    -
none 2.5e-09 1.1e-08

P value adjustment method: BH
```

For the *percent determinism (DET)*,

Table 17 ANOVA results for DET for groups of 3, y axis

```
Anova Table (Type II tests)

Response: DET
                     Sum Sq  Df F value   Pr(>F)
pointColor           0.3879   2  1.4394 0.239233
modality             1.3358   1  9.9143 0.001861 **
pointColor:modality  0.5773   2  2.1426 0.119728
Residuals           30.4493 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 17 suggests that *pointColor* has no significant effect on the *DET values* in the data; however, *modality* is associated with *significantly* different *DET values* between the *HeatMap* and *NoHeatMap* conditions (p-value < 0.05). The effect of interaction between *pointColor* and *modality* is not significant which means that the relation between *DET* and

modality is independent of *pointColor*. This means that the pattern of synchronization between the eye gazes of participants (pattern of group formations) showed different levels of predictability (determinism) between *HeatMap* and *NoHeatMap* conditions of the visualization modality but not between target point colors.

For the *Shannon entropy (ENT)*,

Table 18 ANOVA results for ENT for groups of 3, y axis

```
Anova Table (Type II tests)

Response: ENT
                     Sum Sq  Df F value    Pr(>F)
pointColor            3.629   2  5.8195  0.003432 **
modality              4.965   1 15.9225 8.919e-05 ***
pointColor:modality   1.292   2  2.0716  0.128364
Residuals            70.467 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 18 shows that the *pointColor* are associated with *significantly* different *ENTs* (p-value < 0.05) and *levels of modality* are also associated with *significant* different *ENTs* (p-value < 0.001). This means that the pattern of synchronization between the eye gazes of participants (pattern of group formations) showed different levels of complexity between *HeatMap* and *NoHeatMap* conditions of the visualization modality and between different target point colors (black, grey and none). Follow-up pairwise t-test was conducted on the RR data. The results are shown in Table 19 below.

Table 19 Pair-wise t-test results for three conditions of the target point

```
Pairwise comparisons using t tests with pooled SD

data:  tmp$ENT and tmp$pointColor

     black grey
grey 0.748 -
none 0.018 0.019

P value adjustment method: BH
```

Table 19 suggests that the complexity of the synchronization patterns between participants trajectories are significantly different for target point color *black* vs. *none* and *grey* vs. *none* but not for *black* vs. *grey*.

For the *length of the longest diagonal line (Dmax)*,

Table 20 ANOVA results for Dmax for groups of 3, y axis

```
Anova Table (Type II tests)

Response: Dmax
                    Sum Sq  Df F value   Pr(>F)
pointColor           340.0   2  5.3011 0.005625 **
modality             259.0   1  8.0747 0.004899 **
pointColor:modality  188.0   2  2.9316 0.055340 .
Residuals           7247.8 226
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 20 shows that the *pointColor* are associated with *significantly* different *Dmax* (p-value < 0.05) and *levels of modality* are also associated with *significant* different *Dmax* (p-value < 0.05). This means that the trajectories of participants' eye movements exhibited different divergence rates from each other under the three conditions (*black*, *gray*, *none*) and also different divergence rates from each other under two conditions of the visualization modality (*HeatMap* vs. *NoHeatMap*) in groups of three. Follow-up pairwise t-test was conducted on the Dmax data. The results are shown in Table 21 below.

Table 21 Pair-wise t-test results for the target point conditions

```
Pairwise comparisons using t tests with pooled SD

data:  tmp$Dmax and tmp$pointColor

     black grey
grey 0.516 -
none 0.038 0.010

P value adjustment method: BH
```

Table 21 suggests that the divergence rate for trajectories of the participants' eye movements were different for the target point colors *black* vs. *none* and *grey* vs. *none* but not for *black* vs. *grey*.

For the *average length of diagonal lines (AvD)*,

Table 22 ANOVA results for AvD for groups of 3, y axis

```
Anova Table (Type II tests)

Response: AvD
                     Sum Sq  Df F value Pr(>F)
pointColor            16.68   2  0.7265 0.4847
modality              11.48   1  1.0005 0.3183
pointColor:modality   11.92   2  0.5191 0.5958
Residuals           2594.14 226
```

Table 22 shows that the AvD measure was not significantly different either for the target point colors nor for visualization modality conditions.

4.3.4. Groups of 5, y axis

Two-way Anova on the y axis data for groups of 5 participants on the *recurrence rate (RR)* (Table 23),

Table 23 ANOVA results for RR for groups of 5, y axis

```
Anova Table (Type II tests)

Response: RR
                       Sum Sq Df F value  Pr(>F)
pointColor          0.0009976  2  2.5674 0.08426 .
modality            0.0000476  1  0.2449 0.62230
pointColor:modality 0.0005185  2  1.3345 0.27019
Residuals           0.0130164 67
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

shows that target point color has a significant effect ($p < 0.1$) on the RR measure, but RR is not significantly different for the two visualization modalities.

For the *percent determinism (DET)*,

Table 24 ANOVA results for DET for groups of 5, y axis

```
Anova Table (Type II tests)

Response: DET
                     Sum Sq Df F value Pr(>F)
pointColor          0.04344  2  0.8774 0.4206
modality            0.02257  1  0.9119 0.3430
pointColor:modality 0.01874  2  0.3786 0.6863
Residuals           1.65849 67
```

Table 24 shows that there is no significant effect of either the target point color or the visualization modality on the DET measure for groups of 5.

For the *Shannon Entropy (ENT)*, the data set was not large enough for the ANOVA test to yield meaningful results.

For the *longest diagonal line (Dmax)*,

Table 25 ANOVA results for Dmax for groups of 5, y axis

```
Anova Table (Type II tests)

Response: Dmax
                    Sum Sq Df F value Pr(>F)
pointColor          0.2397  2  1.4764 0.2358
modality            0.0660  1  0.8131 0.3704
pointColor:modality 0.0435  2  0.2678 0.7659
Residuals           5.4397 67
```

Table 25 shows that there was not significant different in *Dmax* for either the target point colors of the visualization modalities.

For the *average diagonal line length (AvD),*

Table 26 ANOVA results for AvD for groups of 5, y axis

```
Anova Table (Type II tests)

Response: AvD
                    Sum Sq Df F value Pr(>F)
pointColor          0.2397  2  1.4764 0.2358
modality            0.0660  1  0.8131 0.3704
pointColor:modality 0.0435  2  0.2678 0.7659
Residuals           5.4397 67
```

Table 26 shows that neither the target point color nor the visualization modality had a significant effect on the *AvD* measure.

The results of analysis for x axis data and y axis data being almost the same for both group sizes suggests that there is a correlation between the recurrences and hence the group formations in the x axis and the recurrences and group formations in the y axis.

4.3.5. Concatenating the data

At this point we can see that the time series of the grey and black conditions being short causes some problems like the impracticality of embedding and the analysis results being inconclusive for the 5 participant groups. One way to overcome this is to design the experiment so that the time series become longer and also collect data from more groups of 5 participants; however, with the data in hand, another way that may produce useful results is to concatenate the data of trials for various conditions to construct one long time-series for each of the six conditions (*black-HeatMap*, *black-NoHeatMap*, *grey-HeatMap*, *grey-NoHeatMap*, *none-HeatMap* and *none-NoHeatMap*). The results for the concatenated data without embedding are shown in plots Figure 18 to Figure 22 for x and y data for groups of three participants.

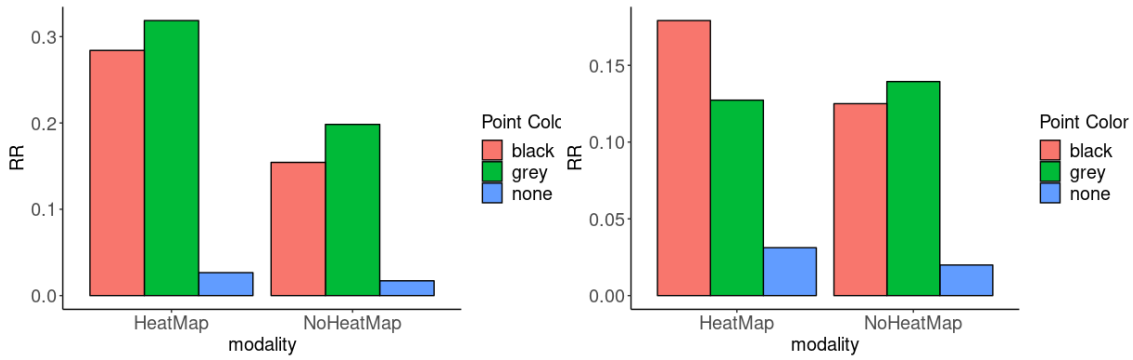


Figure 18 The RR plot for groups of 3. *left:* x-axis data *right:* y-axis data

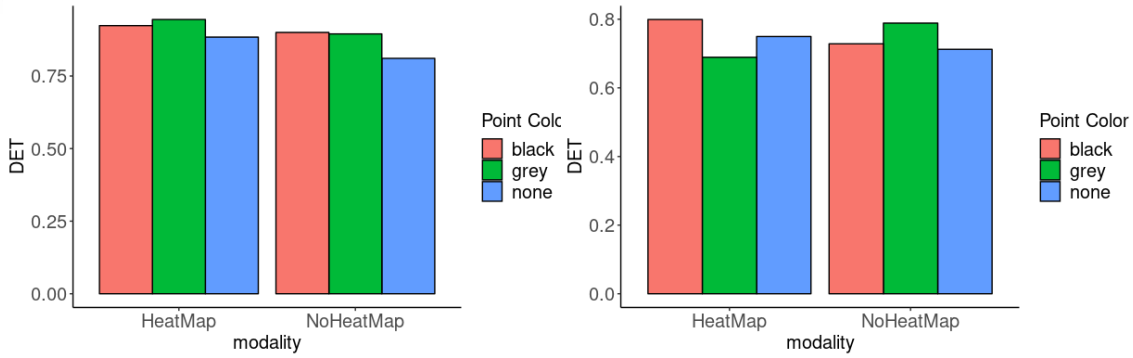


Figure 19 the DET plot for groups of 3. *left:* x-axis data *right:* y-axis data

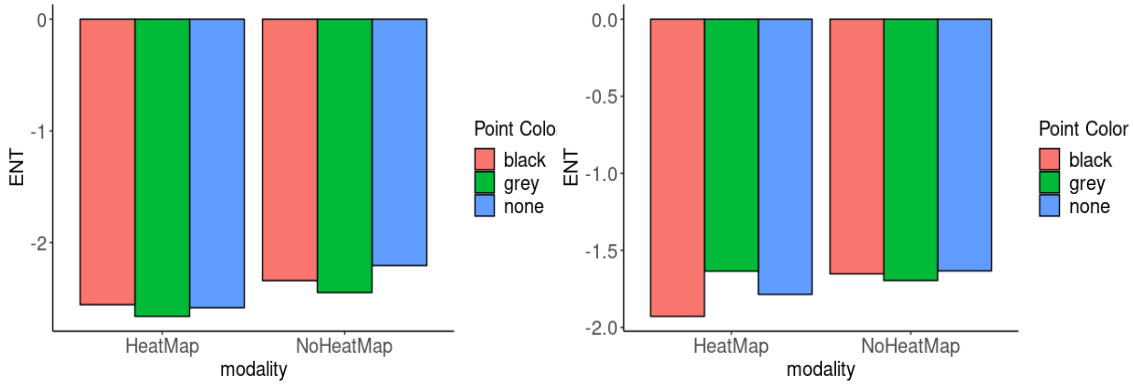


Figure 20 The ENT plot for groups of 3. *left:* x-axis data *right:* y-axis data

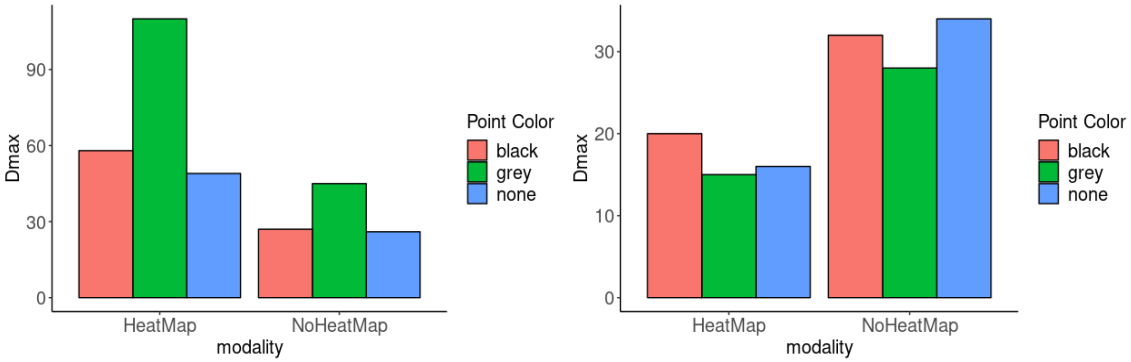


Figure 21 The Dmax plot for groups of 3. *left:* x-axis data *right:* y-axis data

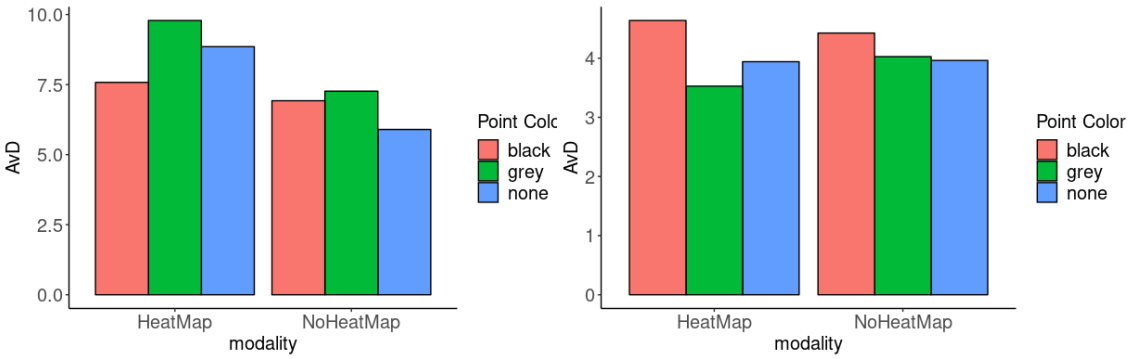


Figure 22 The AvD plot for groups of 3. *left:* x-axis data *right:* y-axis data

After concatenating the data sets, we have a data set long enough for embedding. To embed the time-series we need to decide the time-delay and the dimension parameters (see Section 2.3). To decide the time-delay parameter we use the mutual information formula. The plot for time-delay values from 1 to 50 are shown in Figure 23 for black, grey and none and for the HeatMap condition for x-axis data.

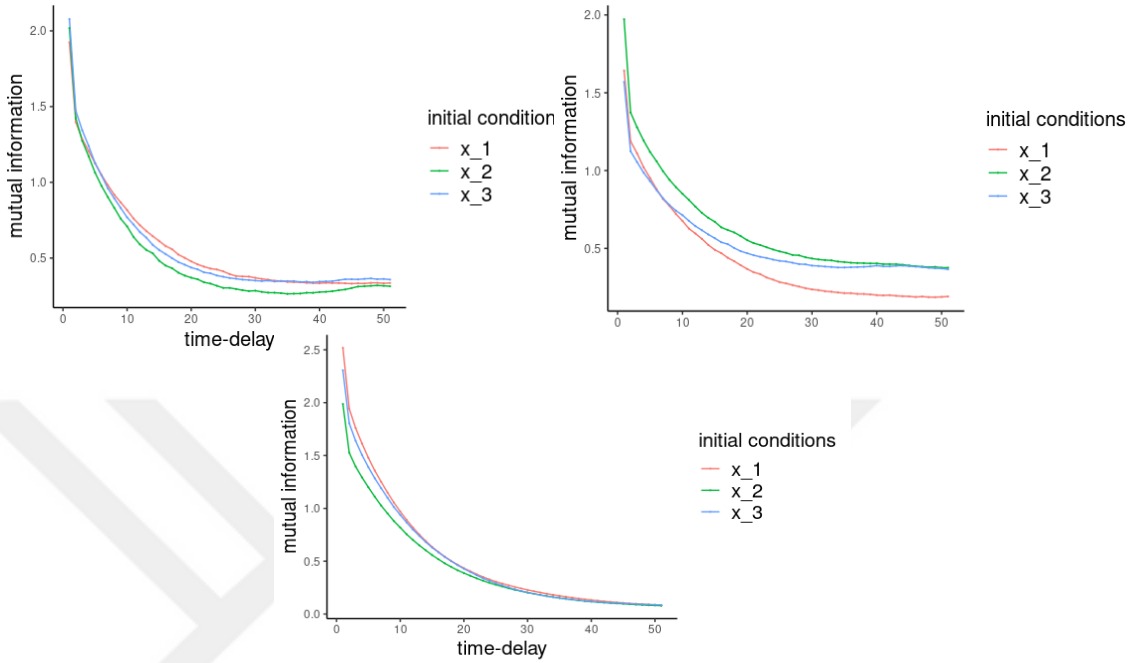


Figure 23 Mutual information plot for groups of 3 for x-axis data. *top-left:* Black-HeatMap, *top-right:* Grey-HeatMap, *bottom:* None-HeatMap

The local minimum seems to be somewhere between 30 and 40, so we set it to 35. Using this time-delay value and the false nearest neighbor method we can find the appropriate dimensions for embedding. The false nearest neighbor plots are shown in Figure 24 for black, grey and none conditions and for the HeatMap condition.

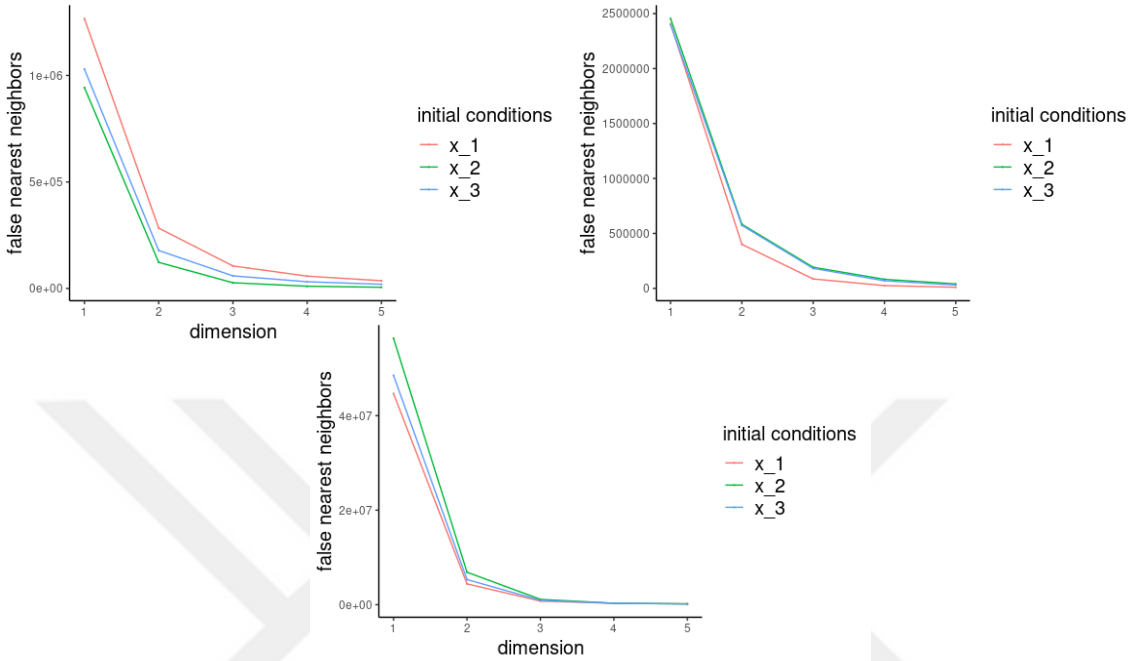


Figure 24 False nearest neighbor plot for groups of 3 for x-axis data. top-left: Black-HeatMap, top-right: Grey-HeatMap, bottom: None-HeatMap

The plot for time-delay values from 1 to 50 are shown in Figure 25 for black, grey and none and for the NoHeatMap condition for x-axis data.

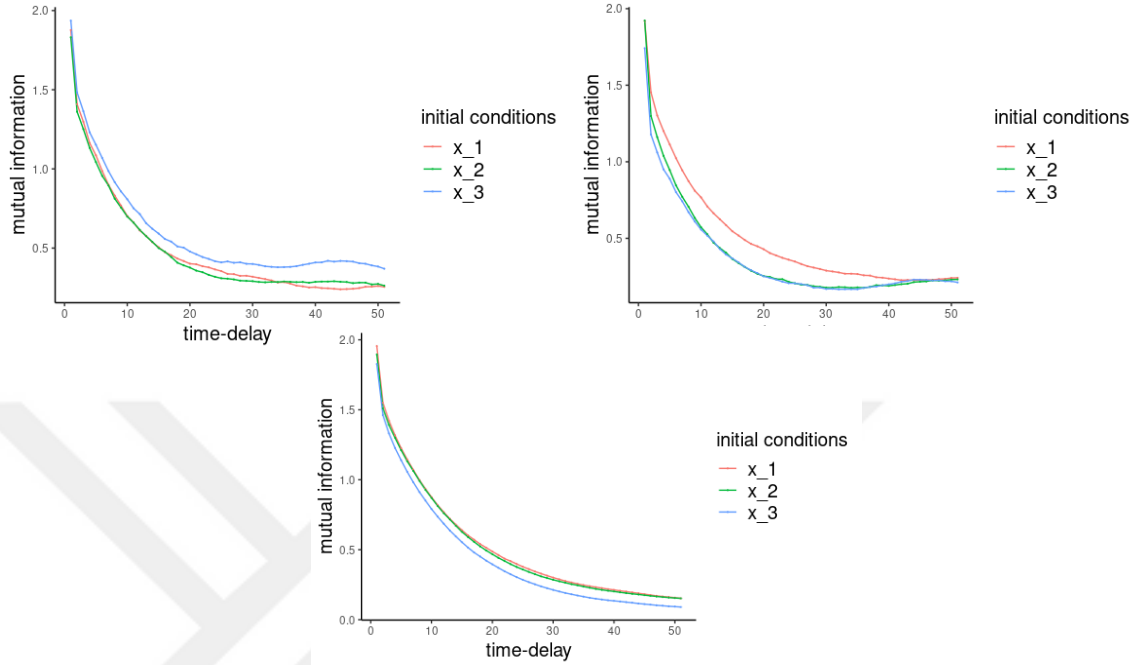


Figure 25 Mutual information plot for groups of 3 for x-axis data. top-left: Black-NoHeatMap, top-right: Grey-NoHeatMap, bottom: None-NoHeatMap

Again, we set the time-delay value to 35 time-delay. The false nearest neighbor plots are shown in Figure 26 for black, grey and none conditions and for the NoHeatMap condition.

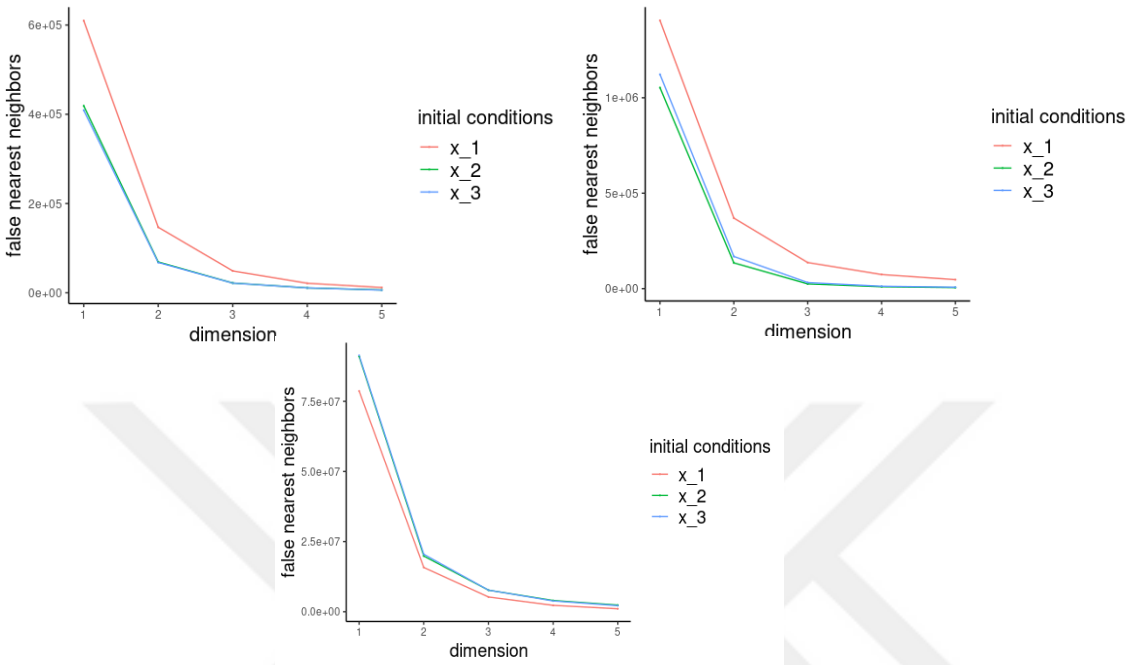


Figure 26 False nearest neighbor plot for groups of 3 x-axis data. top-left: Black-NoHeatMap, top-right: Grey-NoHeatMap, bottom: None-NoHeatMap

The bar plots for RR, DET, ENT, Dmax and AvD are shown in Figure 27 to Figure 31 along with the same plots for the non-embedded data for the sake of comparison.

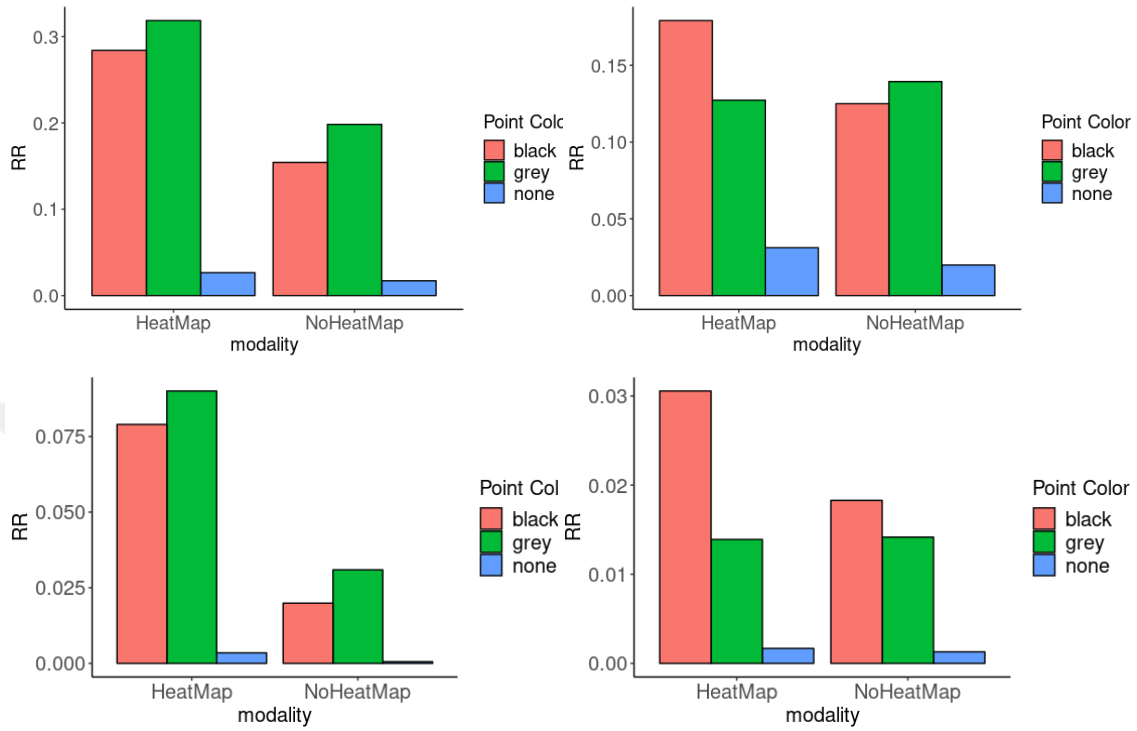


Figure 27 The RR plot for groups of 3. *top-left*: x-axis data non-embedded, *top-right*: y-axis data non-embedded, *bottom-left:* x-axis data embedded, *bottom-right*: y-axis embedded

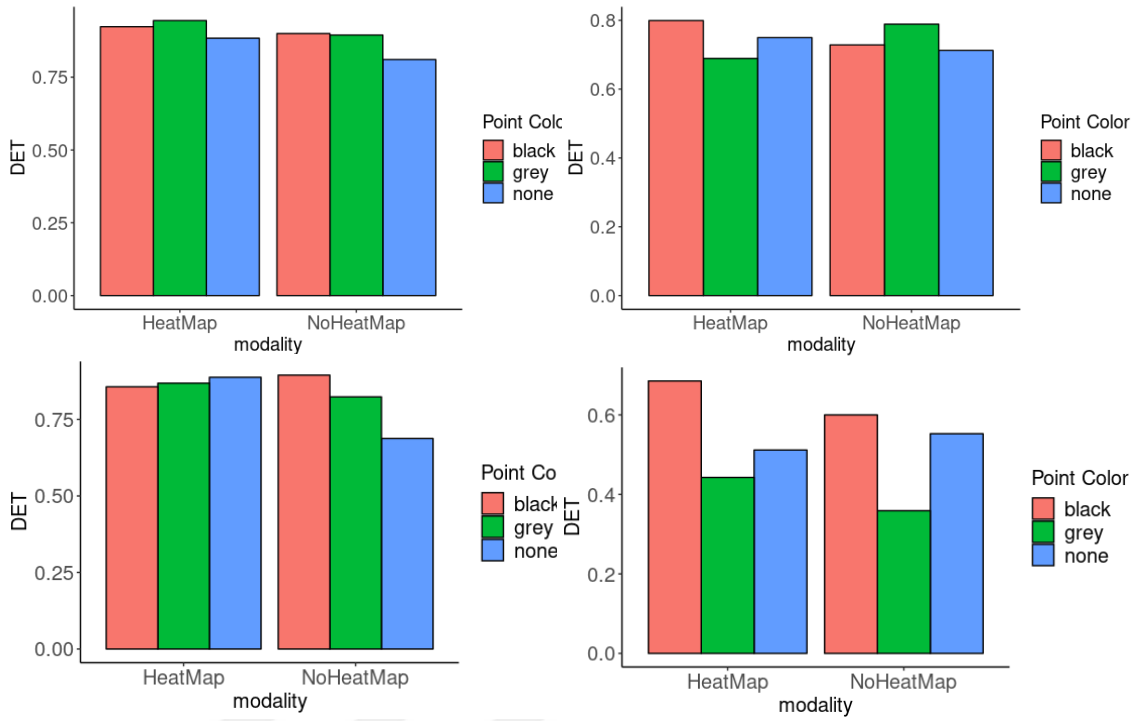


Figure 28 The DET plot for groups of 3. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

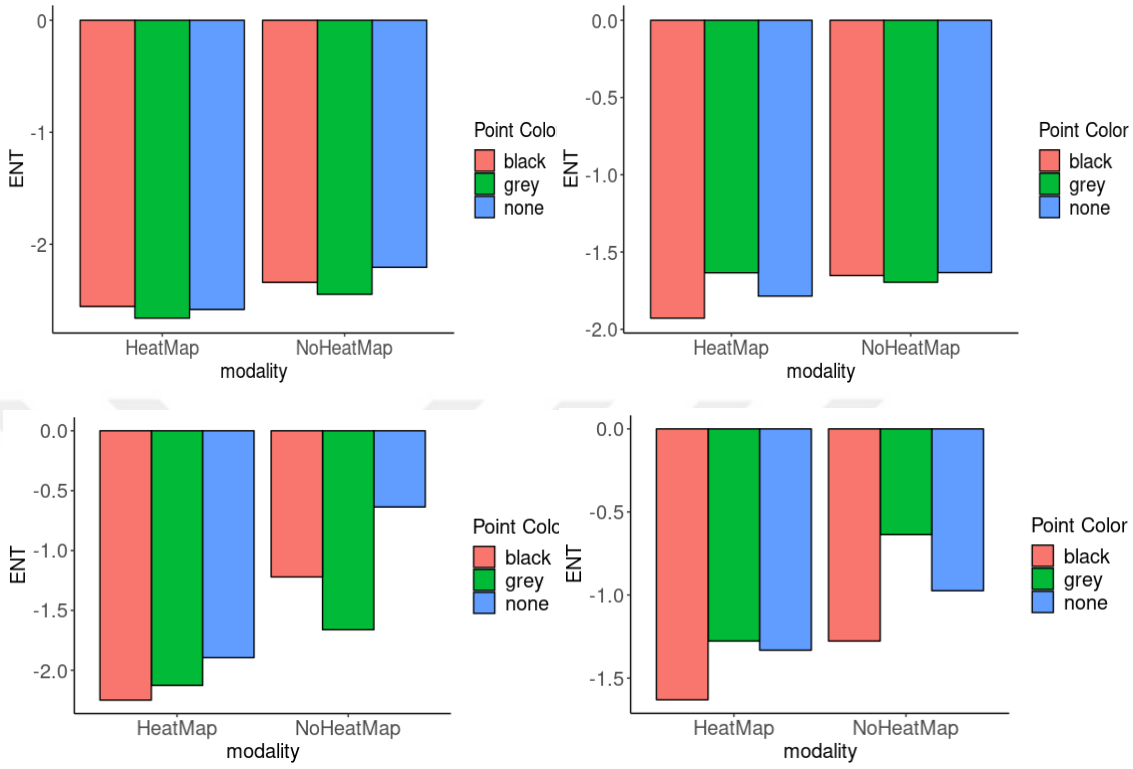


Figure 29 The ENT plot for groups of 3. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

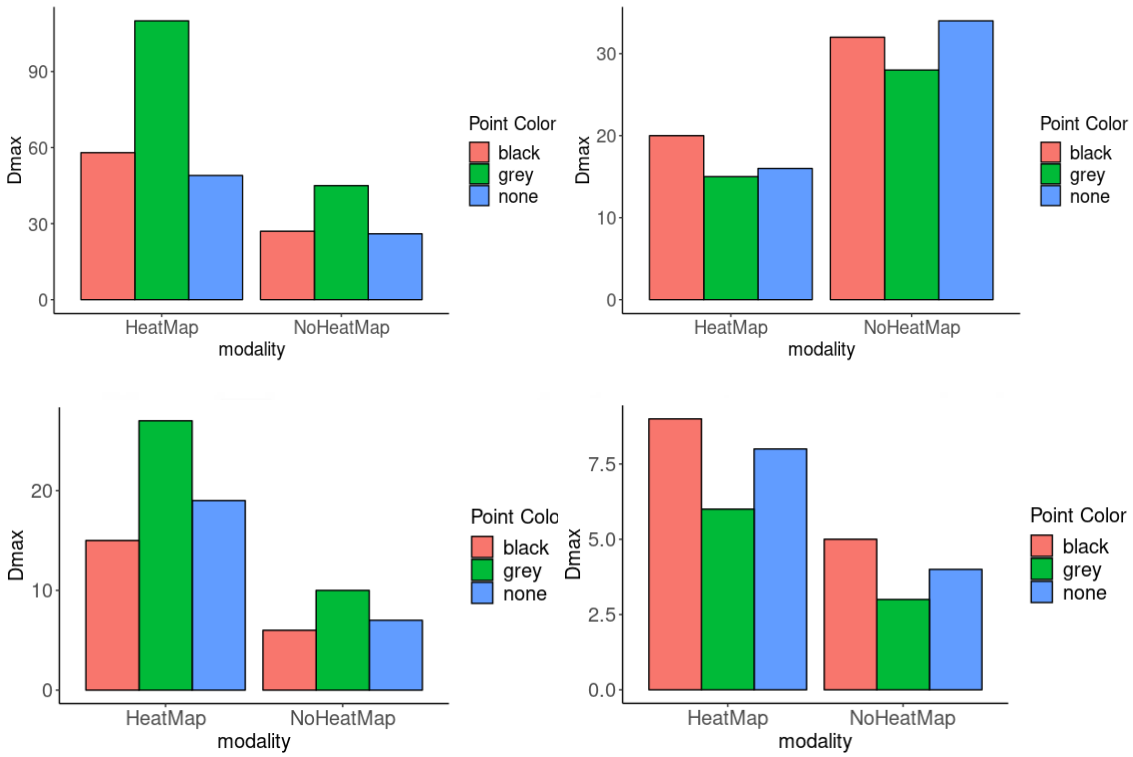


Figure 30 The Dmax plot for groups of 3. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

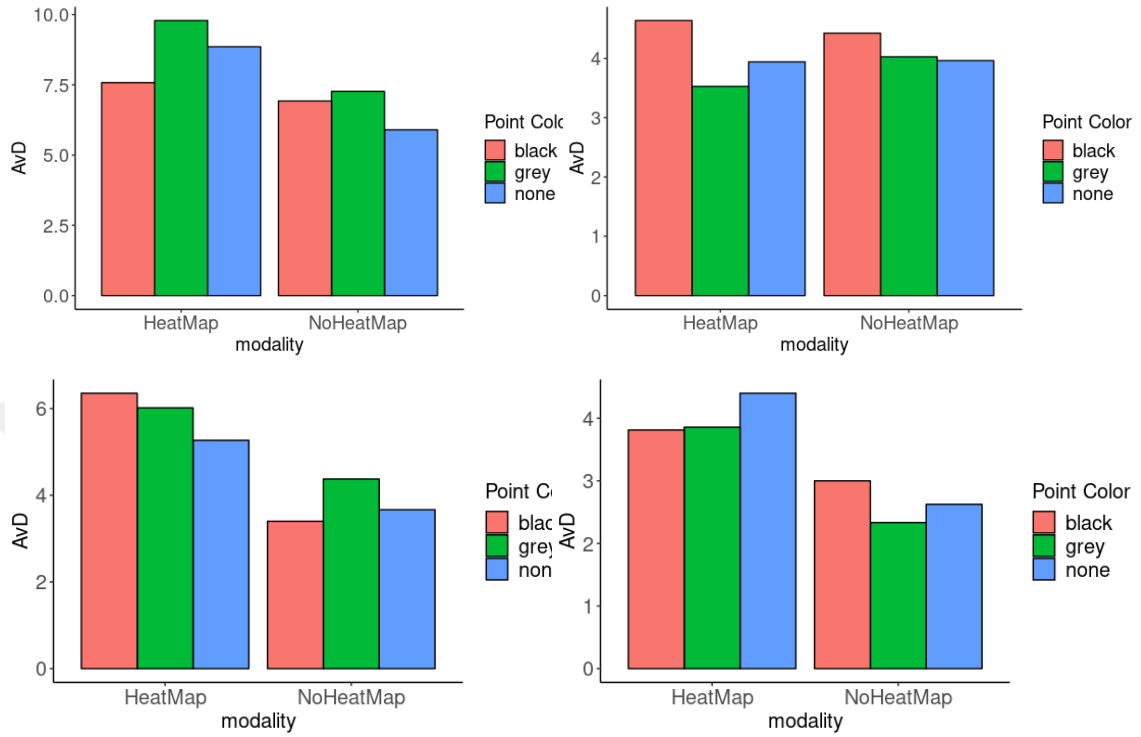


Figure 31 The AvD plot for groups of 3. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

As for the groups of 5 participants, we had to set the threshold at a higher value than 30 pixels to get meaningful results for the CRQA measures. To embed the data for the groups of 5 participants the mutual information plots and the false nearest neighbor plots for the three conditions of black, grey and none and two conditions of HeatMap and NoHeatMap are shown in Figure 32 to Figure 35 for x-axis data.

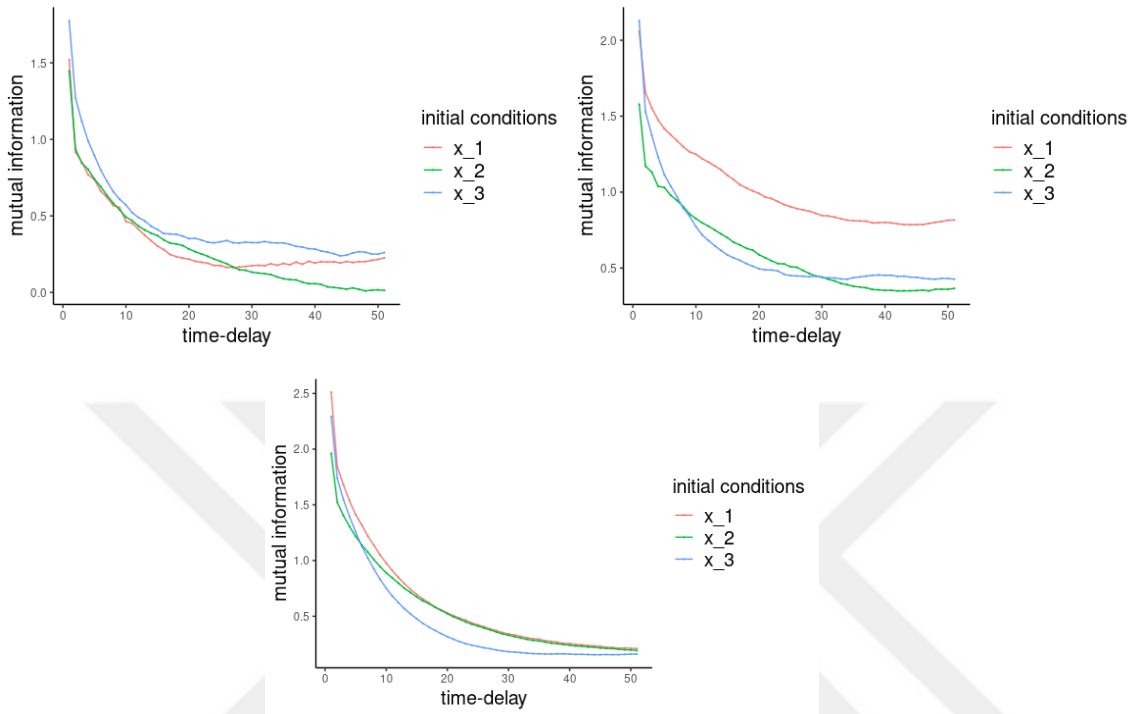


Figure 32 Mutual information plot for groups of 5 for x-axis data. top-left: Black-HeatMap, top-right: Grey-HeatMap, bottom: None-HeatMap

The local minimum is set to 35.

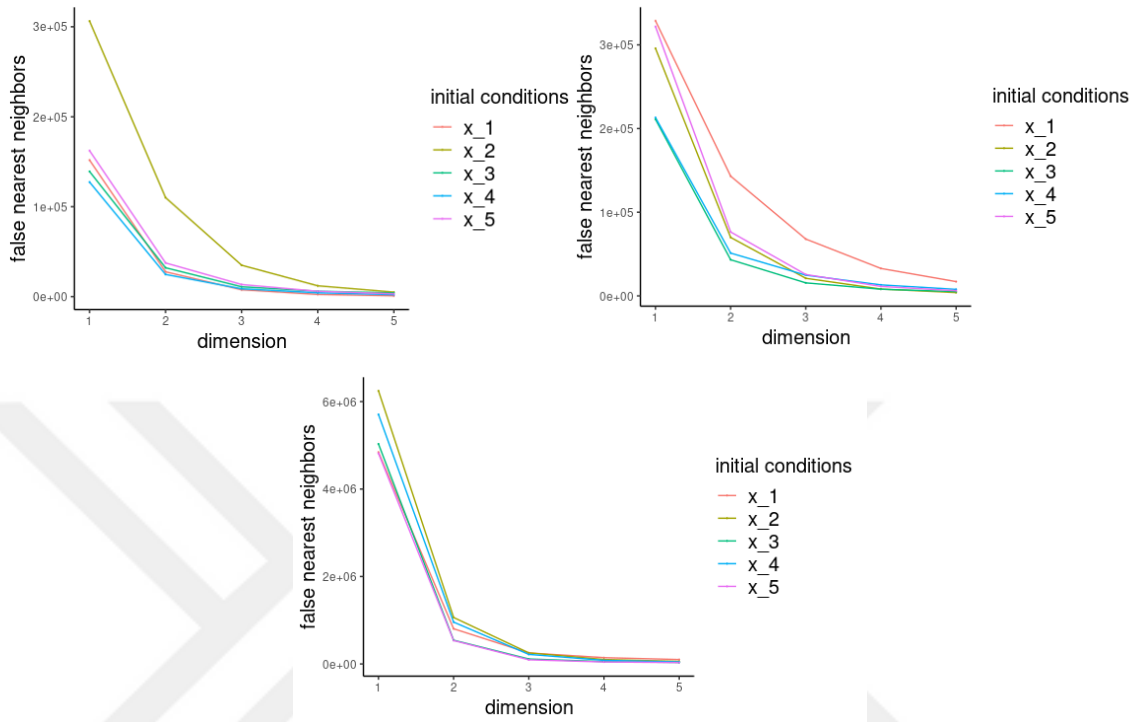


Figure 33 False nearest neighbor plot for groups of 5 x-axis data. top-left: Black-NoHeatMap, top-right: Grey-NoHeatMap, bottom: None-NoHeatMap

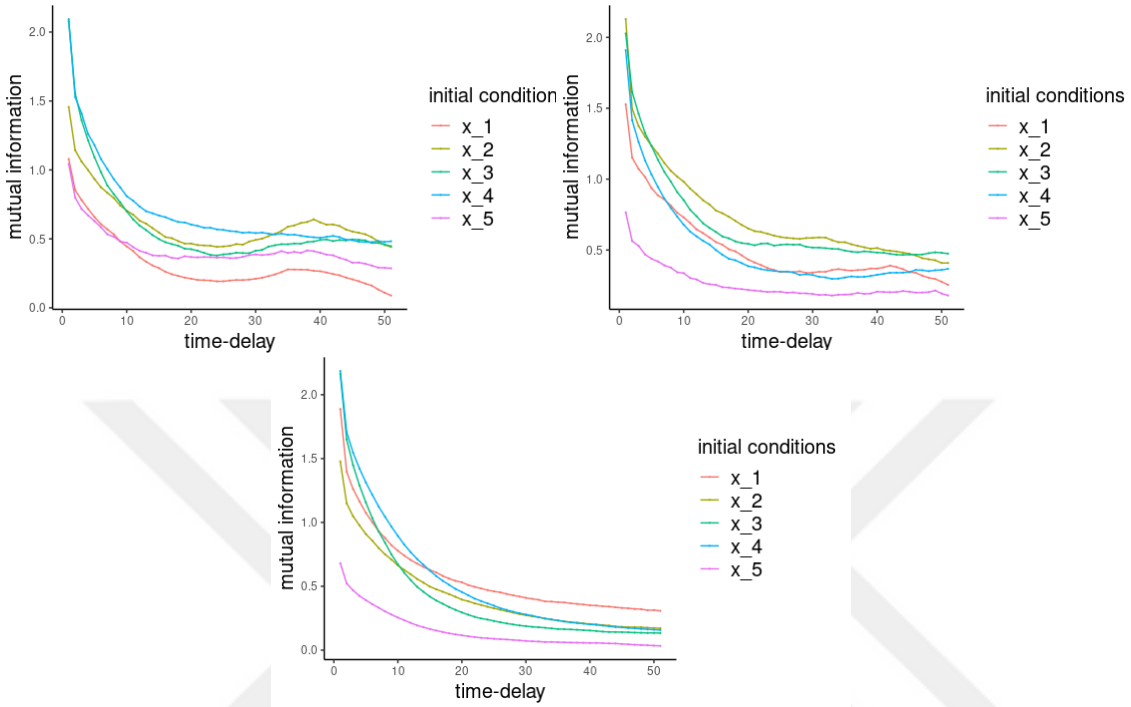


Figure 34 Mutual information plot for groups of 5 for x-axis data. top-left: Black-NoHeatMap, top-right: Grey-NoHeatMap, bottom: None-NoHeatMap

The time-delay value is set to 25.

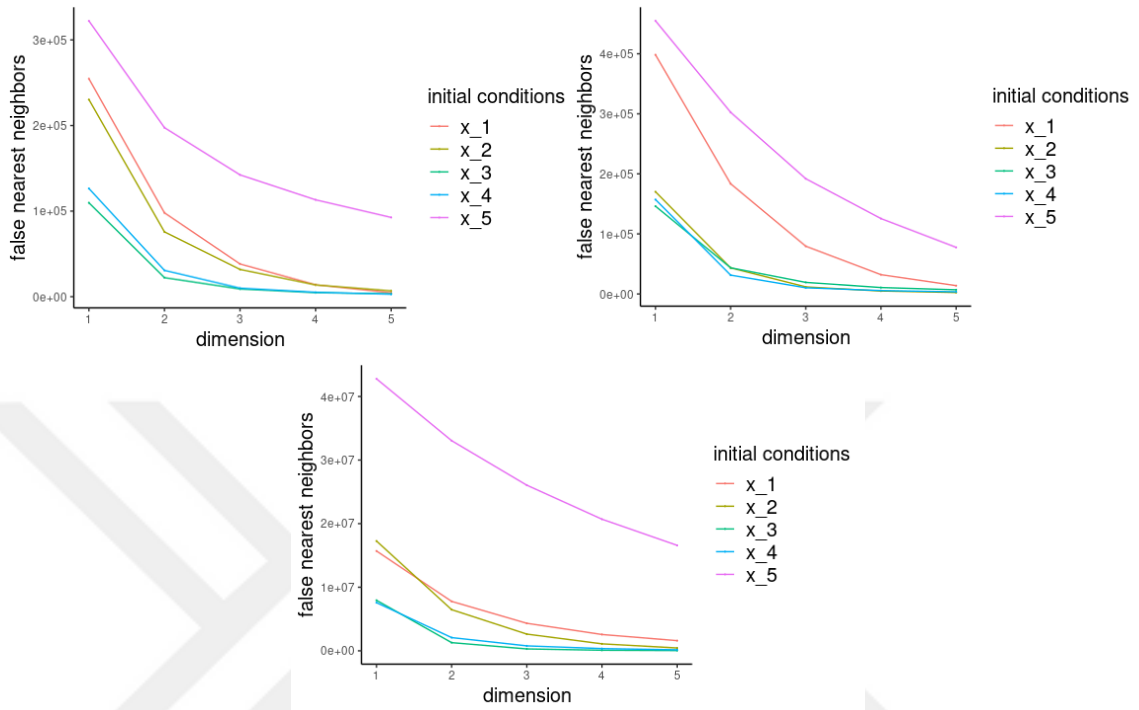


Figure 35 False nearest neighbor plot for groups of 5 x-axis data. top-left: Black-NoHeatMap, top-right: Grey-NoHeatMap, bottom: None-NoHeatMap

The bar plots for RR, DET, ENT, Dmax and AvD are shown in Figure 36 to Figure 40 along with the same plots for the non-embedded data for the sake of comparison.

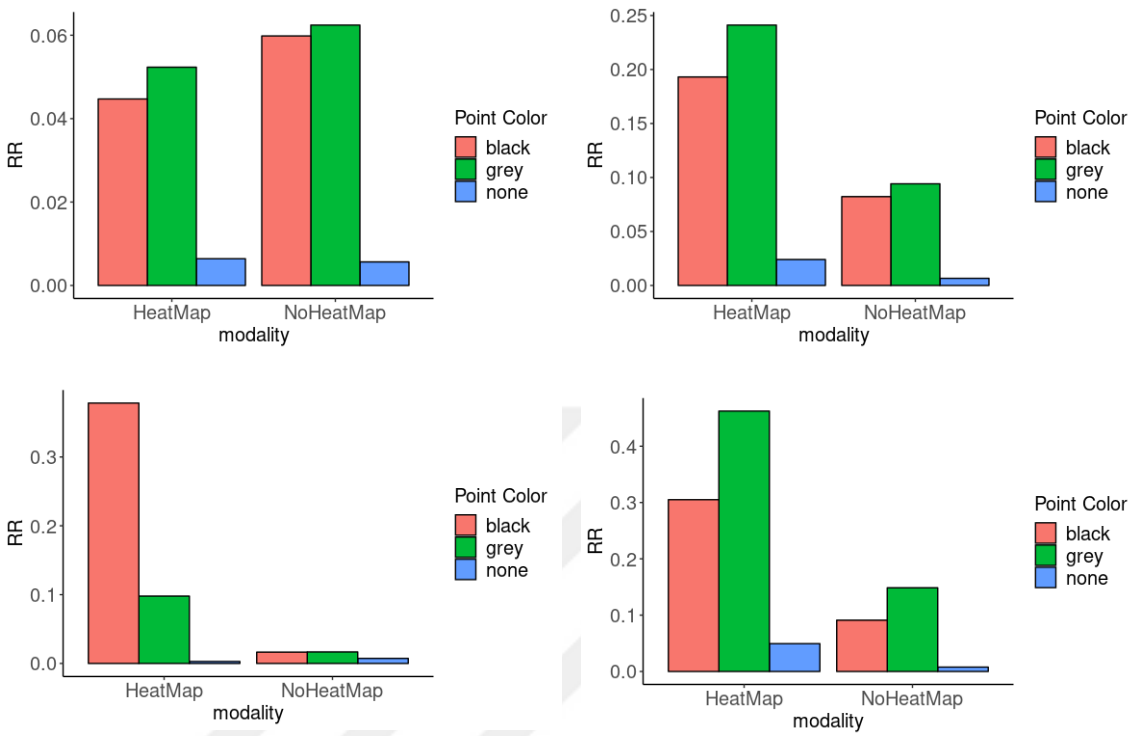


Figure 36 The RR plot for groups of 5. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

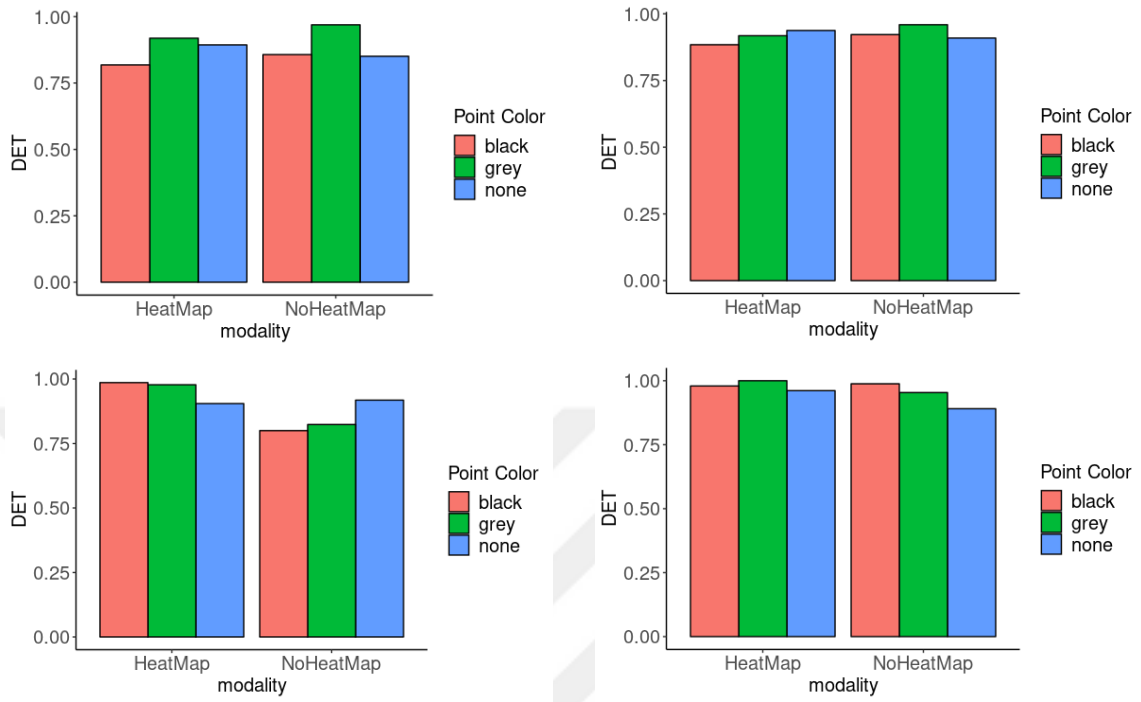


Figure 37 The DET plot for groups of 5. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

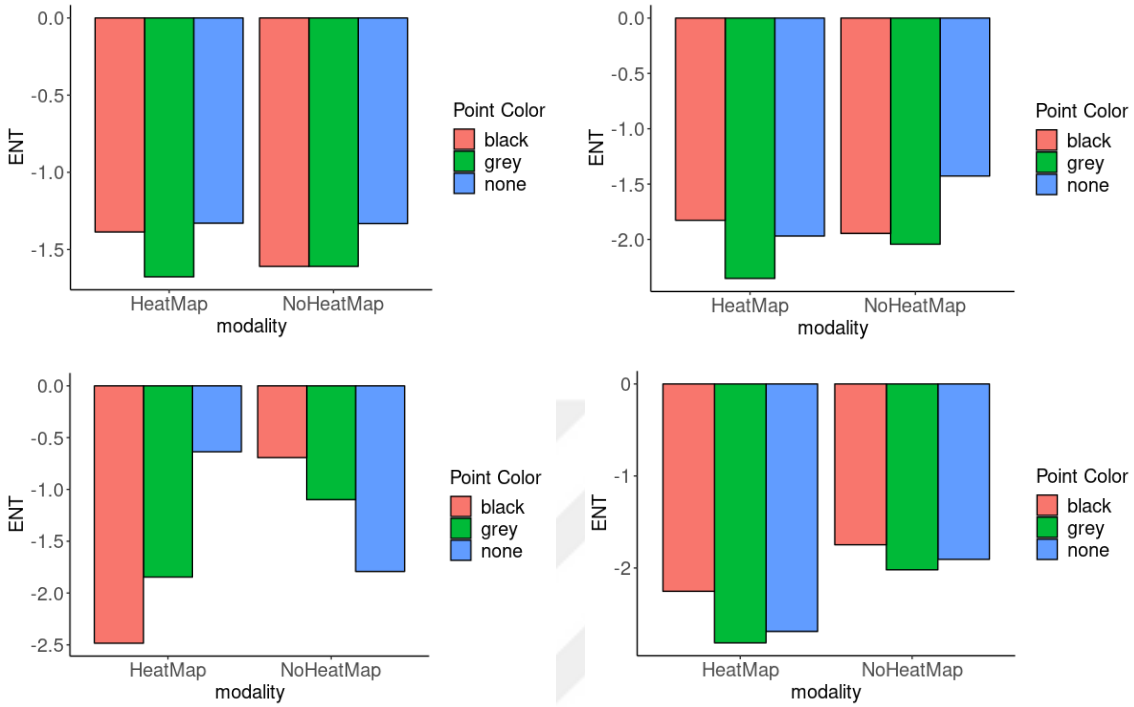


Figure 38 The ENT plot for groups of 5. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

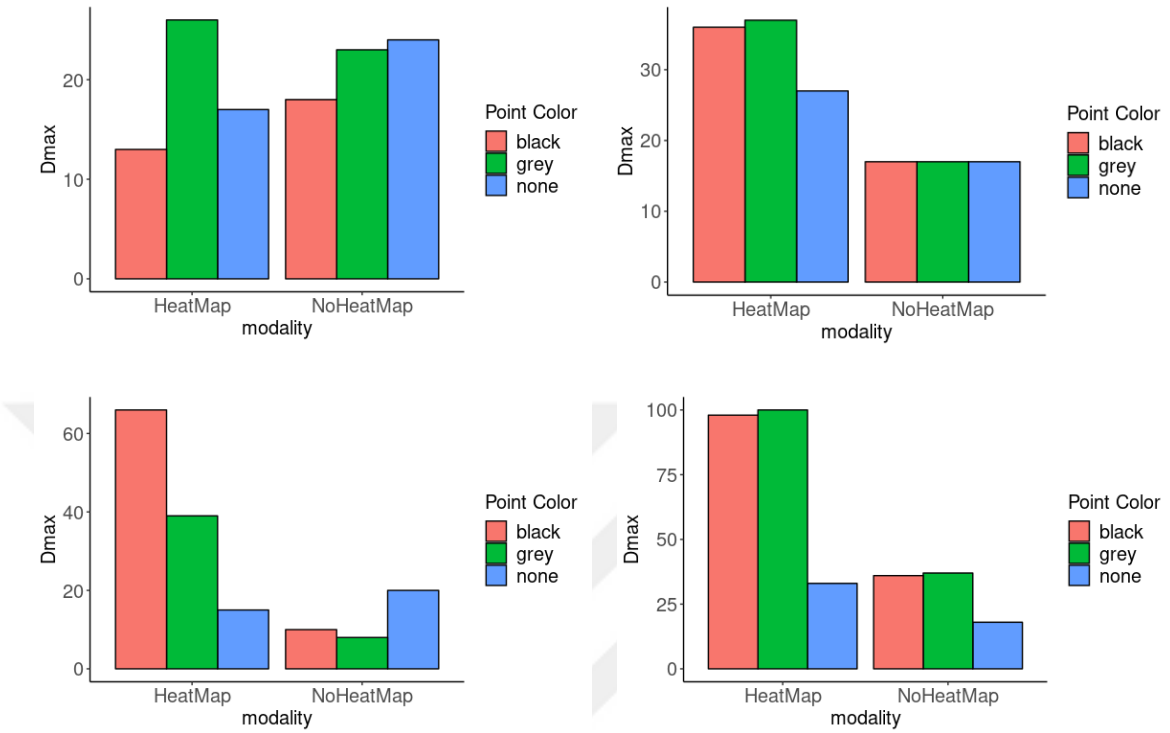


Figure 39 The Dmax plot for groups of 5. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

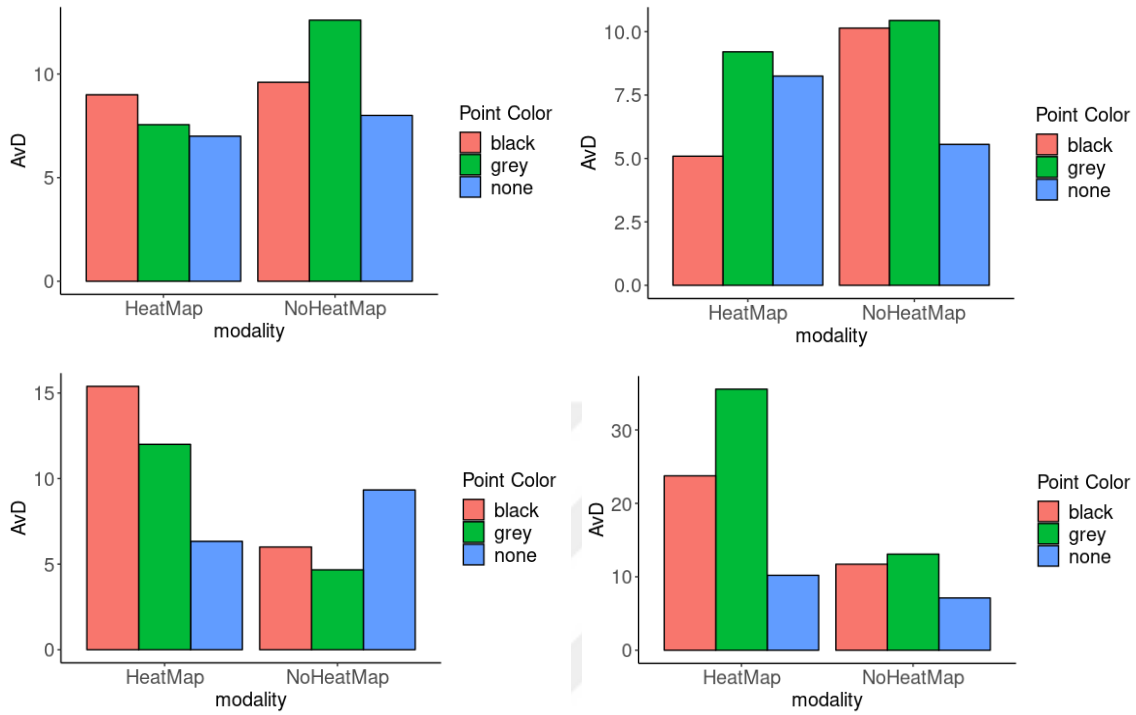


Figure 40 The AvD plot for groups of 5. top-left: x-axis data non-embedded, top-right: y-axis data non-embedded, bottom-left: x-axis data embedded, bottom-right: y-axis embedded

By examining Figure 18 to Figure 40, we conclude that the results for the *embedded* data are more cohesive than that of not embedded data. The plots for the various measures of CRQA for embedded data suggest that the measures generally have lower values in the case of *NoHeatMap* condition than *HeatMap* condition except for the DET measure.

CHAPTER 5

CONCLUSION

5.1.Discussion

An individual in a group is affected, constrained and guided by the group action. Accordingly, it is necessary to understand how actions, processes and changes, occur inside a group and between groups. This is called the *group dynamics* or the study of how groups change over time. Eye tracking is a fruitful means for studying group dynamics because the eye movement give insights about the cognitive processes behind them.

In this thesis we evaluated alternative measures for describing the group formation patterns (synchronization patters) of eye movements in a visual search task through a group eye tracking experiment. To do this we introduced a software tool for the group eye tracking paradigm (GET) called the *GETapp* (see section 3.1). Using this software, we collected data from groups of participants who performed a search task collectively. The target of the search task was a point which appeared in three conditions of *black*, *grey* and *none (not visible)* in random. While searching for the target point, the participants were either provided with a visualization modality in the form of a *heat-map* or were not provided with such a modality.

The methods for analyzing the data were part of a nonlinear time-series analysis theory called the *recurrence quantification analysis (RQA)*. We introduced a specific part of the RQA, namely a version of it called the *cross-recurrence quantification analysis (CRQA)* for analyzing two different time-series. We provided the basics of its theory and an explanation of how to expand it to be able to analyze an arbitrary number of time-series. For the analysis of the data, we introduced a software tool, the `generalRQA` package for the R programming language which implements methods for calculating CRQA measures based on the CRQA theory. We calculated five CRQA measures for 6 groups of three participants and two groups of 5 participants with the target point in three conditions of target point color (*black*, *grey* and *none*) and the visualization modality in two conditions of visualization modality (*HeatMap* and *NoHeatMap*).

The CRQA method works on the concept of recurrence. In the current study a recurrence is defined as when the time-series of the participants' eye gazes become focused in a neighborhood of a certain radius. The measures were the *recurrence rate (RR)* which is related to the definition of the *correlation sum* and designates the ratio of recurrences of the participants time-series, the *percent determinism (DET)* which is a

measure of the *predictability* of the group formation patterns of the participants' time-series, the *Shannon entropy (ENT)* which is a measure of the complexity of the group formation patterns of the participants' time-series, the *lengths of the longest diagonal line (Dmax)* which is a measure of the divergence rate of the trajectories of participants' eye movements and the *average length of diagonal lines (AvD)* which is another measure for the divergence rate of the trajectories of participants' eye movements.

The radius can either be set directly or by setting a range for the *recurrence rate,* so that the radius is calculated so that the recurrence rate falls into the predefined range. The choice of a method for setting the *radius* depends on the experiments. In the current study we set the radius to be 30 pixels corresponding to 4 degrees of visual angle. This is chosen thus, because 4 degrees of visual angle subsumes the focus area of the human eye.

Among the aforementioned CRQA measures, in the case of groups of three participants, only for the *HeatMap* vs. *NoHeatMap* conditions of the visualization modality **all** the aforementioned CRQA measures exhibited significant differences both for the x axis and y axis data (except the *AvD* measure which showed no significant difference for any condition). In the case of groups of 5 participants, none of the measures were significantly different for any of the conditions, neither for the x axis or the y axis data. However, when the data are concatenated CRQA analysis showed similar results for groups of three and groups of 5 which is that the values of CRQA measures are generally lower in the case of *NoHeatMap* than *HeatMap.* This makes the results promising but difficult to choose an appropriate CRQA measure for differentiating between group formation patterns; however, it may be that by collecting data from more groups of 5 participants, some of the CRQA measures show statistically significant difference for different conditions of visualization modality.

5.2.Limitations and Future Work

The goal of this thesis was to find a measure that can differentiate between different group formation patterns in group eye tracking search tasks. The results were promising for groups of three participants; however, for groups of 5 participants, none of the proposed CRQA measures showed significant difference for different target point color and visualization modality conditions. This may be resolved by collecting data from more groups of 5. The number of groups that we collected data from in this study may have been too low to reach a convergence.

In proposing ways to find appropriate measures for group formation patterns, we have only considered 4 of the five classic RQA measures (*Dmax* and *AvD* considered the same measure) and not experimented with all the measures available in the recurrence quantification analysis. The other classic measure of RQA is the *TREND (TND)* (see section 1). Other than these classic measures, five extended measures and other more modern measures have been proposed (Webber Jr & Marwan, 2015). A modern approach

is to consider the recurrence matrix (section 2.4) as the adjacency matrix of a network. This allows to use measures from complex networks statistics like *clustering coefficient*, *betweenness coefficient* and *average shortest path length* as measures for differentiating patterns of group formation (R. V. Donner et al., 2011; Reik V. Donner et al., 2010; Reik V Donner, Zou, Donges, Marwan, & Kurths, 2010; Norbert Marwan, Donges, Zou, Donner, & Kurths, 2009). Another measure for characterizing group formation patterns can be extracted from the *recurrence time*. The recurrence time is the time it takes for a state to recur in a phase space. Measures based on recurrence time, like *recurrence period density entropy* (Little, McSharry, Roberts, Costello, & Moroz, 2007), *mean recurrence time* and *number of the most probable recurrence time* (Ngamga et al., 2008), can be considered measures of complexity, like Shannon entropy (ENT).

In section 2.3, we introduced Takens' embedding theorem and in section 2.4 we embedded our example data set before doing cross-recurrence quantification analysis; however, in the case of the real data we did not embed out 1-dimensional time-series into a higher dimensional phase space. This was due to the length of our time-series where the target point color was either *black* or *grey*. The lengths of these time-series ranged from 30 to 50 time steps approximately. This is due to the fact that when the target point has high saliency, it takes a relatively short time to detect it. We can infer from the embedding theorem and its implementation in the `generalRQA` package that if the length of a time series is n time steps, the length of the embedded time-series would be $n - (\tau(d-1))$ where τ is the delay-parameter and d is the embedding dimension. Thus, the lengths of the time-series for the *black* and *grey* conditions of the target point were too short to allow embedding. This being said, analyzing recurrences with embedded time-series may reveal insights into the group formation patters not available otherwise. The time-series for the *none* condition of the target point can be embedded as their lengths range from 200 to 800 approximately.

In the future endeavor, we are hoping that we could make the GET platform available online. The eye tracking devices are getting more accessible and they may soon be implemented on personal computers (as some of them have). This would open a range of possibilities for data collection on group eye tracking. Having this perspective in mind, we have chosen JavaScript as the programming language for the GETapp. the fact that JavaScript and its related technologies are primarily web-based technologies, is in accordance with the outlook for the GET platform which is to further develop it to use over the web with a large number of participants.

Furthermore, the design of the experiment was so that all the participants started from the same location of the screen which was a way of simplifying he experiment; however, were this limitation to be lifted, it may result in additional insights into group formation patterns. As another simplification, we used two-way ANOVA statistical test to assess the CRQA analysis results; however, using a *Hierarchical Linear Model* may better suite the structure of the data.

Another aspect that can be considered in future work, is the use of different visualization modalities. In the current study we used the heat map as a visualization modality; however, the complex dynamics of the heat map visualization modality could have had a negative effect on the results of the experiment. Using a simpler visualization modality in the form of a Donut, may result in better outcomes.

REFERENCES

Acarturk, C., Kalkan, S., & Arslan Aydın, Ü. (2018). A Multimodal Framework for Analysing Gaze in Dyadic Communication (MAGiC v1.0).

Anderson, N. C., Bischof, W. F., Laidlaw, E. W. K., Risko, F. E., & Kingstone, A. (2013). Recurrence quantification analysis of eye movements. *Behavior Research Methods*, *45*(3), 842–856. https://doi.org/10.3758/s13428-012-0299-5

Antonio, & Narzo, F. Di. (2013). tseriesChaos: Analysis of nonlinear time series. Retrieved from https://cran.r-project.org/package=tseriesChaos

Broz, F., Lehmann, H., Nehaniv, C. L., & Dautenhahn, K. (2012). Mutual gaze, personality, and familiarity: Dual eye-tracking during conversation. In *RO-MAN, 2012 IEEE* (pp. 858–864).

Chelidze, D. (2017). Reliable Estimation of Minimum Embedding Dimension Through Statistical Analysis of Nearest Neighbors. *Journal of Computational and Nonlinear Dynamics*, *12*(5). https://doi.org/10.1115/1.4036814

Coco, M. I., & Dale, R. (2016). Cross-Recurrence Quantification Analysis of Categorical and Continuous Time Series: an R package. *Frontiers in Psychology*, *5*(355), 1–31. https://doi.org/10.3389/fpsyg.2014.00510

Coco, M. I., with contributions of James D. Dixon, R. D., & Nash, J. (2015). crqa: Cross-Recurrence Quantification Analysis for Categorical and Continuous Time-Series. Retrieved from https://cran.r-project.org/package=crqa

Conradt, L., & List, C. (2009, March 27). Group decisions in humans and animals: A survey. *Philosophical Transactions of the Royal Society B: Biological Sciences*. The Royal Society. https://doi.org/10.1098/rstb.2008.0276

Deniz, O. (2016). *Group Eye Tracking*. Retrieved from http://etd.lib.metu.edu.tr/upload/12620362/index.pdf

Donner, R. V., Heitzig, J., Donges, J. F., Zou, Y., Marwan, N., & Kurths, J. (2011). The geometry of chaotic dynamics - A complex network perspective. In *European*

Physical Journal B (Vol. 84, pp. 653–672). https://doi.org/10.1140/epjb/e2011-10899-1

Donner, R. V., Small, M., Donges, J. F., Marwan, N., Zou, Y., Xiang, R., & Kurths, J. (2010). Recurrence-based time series analysis by means of complex network methods. *International Journal of Bifurcation and Chaos*, *21*(04), 1019–1046. https://doi.org/10.1142/S0218127411029021

Donner, R. V, Zou, Y., Donges, J. F., Marwan, N., & Kurths, J. (2010). Recurrence networks-a novel paradigm for nonlinear time series analysis. *New Journal of Physics*, *12*(3), 033025. https://doi.org/10.1088/1367-2630/12/3/033025

Duchowski, A. T. (2017a). *Eye tracking methodology: Theory and practice: Third edition*. *Eye Tracking Methodology: Theory and Practice: Third Edition*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-57883-5

Duchowski, A. T. (2017b). Visual Attention. In *Eye Tracking Methodology* (pp. 3–13). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-57883-5_1

Eckmann, J.-P., Kamphorst, S. O., & Ruelle, D. (1987). Recurrence Plots of Dynamical Systems. *Europhysics Letters (EPL)*, *4*(9), 973–977. https://doi.org/10.1209/0295-5075/4/9/004

Faure, P., & Korn, H. (1998). A new method to estimate the Kolmogorov entropy from recurrence plots: its. *Physica D: Nonlinear Phenomena*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.8245&rep=rep1&type=pdf

Forsyth, D. R. (2014). *Group Dynamics*. *Annu. Rev. Psychol.* (Vol. 15). https://doi.org/10.1146/annurev.ps.15.020164.002225

Fraser, A. M., & Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information. *Physical Review A*, *33*(2), 1134–1140. https://doi.org/10.1103/PhysRevA.33.1134

Furstenberg, H. (1981). Poincaré recurrence and number theory. *Bulletin of the American Mathematical Society*. https://doi.org/10.1090/S0273-0979-1981-14932-6

Fusaroli, R., & Tylén, K. (2016). Investigating Conversational Dynamics: Interactive Alignment, Interpersonal Synergy, and Collective Task Performance. *Cognitive Science*, *40*(1), 145–171. https://doi.org/10.1111/cogs.12251

Gilmore, R. (1998). Topological analysis of chaotic dynamical systems. *Reviews of Modern Physics*, *70*(4), 1455–1529. https://doi.org/10.1103/RevModPhys.70.1455

Grassberger, P., & Procaccia, I. (1983). Characterization of strange attractors. *Physical Review Letters*, *50*(5), 346–349. https://doi.org/10.1103/PhysRevLett.50.346

Hirata, Y., Horai, S., & Aihara, K. (2008). Reproduction of distance matrices and original time series from recurrence plots and their applications. *European Physical Journal: Special Topics*, *164*(1), 13–22. https://doi.org/10.1140/epjst/e2008-00830-8

Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Weijer, J. Van De. (2011). Eye Tracking: A comprehensive guide to methods and measures. Retrieved from https://www.google.com/books?hl=en&lr=&id=5rIDPV1EoLUC&oi=fnd&pg=PR12&dq=Eye+Tracking:+A+Comprehensive+Guide+to+Methods+and+Measures&ots=_w1zQUsNsP&sig=x347fMl0bvtN-m-1IeszlmSojuU

Jermann, P., Nüssli, M.-A., & Li, W. (2010). Using dual eye-tracking to unveil coordination and expertise in collaborative Tetris. In *Proceedings of the 24th BCS Interaction Specialist Group Conference* (pp. 36–44).

Kantz, H., & Schreiber, T. (2004). *Nonlinear Time Series Analysis*. *Nonlinear Time Series Analysis* (Vol. 47). https://doi.org/10.1198/tech.2005.s306

Kennel, M. B., Brown, R., & Abarbanel, H. D. I. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A*, *45*(6), 3403–3411. https://doi.org/10.1103/PhysRevA.45.3403

Koebbe, M., & Mayer-Kress, G. (1992). Use of Recurrence Plots in the Analysis of Time-Series Data. In *Nonlinear modeling and forecasting* (Vol. XXI, pp. 361–378). Retrieved from https://pdfs.semanticscholar.org/37ae/58bae759a7d719fdcf1cb44e7d5b6752d22d.pdf

Kuhn, G., Tatler, B. W., Findlay, J. M., & Cole, G. G. (2008). Misdirection in magic: Implications for the relationship between eye gaze and attention. *Visual Cognition*, *16*(2–3), 391–405. https://doi.org/10.1080/13506280701479750

Lathrop, D. P., & Kostelich, E. J. (1989). Characterization of an experimental strange attractor by periodic orbits. *Physical Review A*, *40*(7), 4028–4031. https://doi.org/10.1103/PhysRevA.40.4028

Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A. E., & Moroz, I. M. (2007).

Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering Online*, *6*(1), 23. https://doi.org/10.1186/1475-925X-6-23

Liversedge, S. P., Gilchrist, I. D., & Everling, S. (2011). *The Oxford handbook of eye movements*. *Oxford library of psychology*. https://doi.org/10.1093/oxfordhb/9780199539789.001.0001

Ludvigsen, S., Law, N., Rose, C. P., & Stahl, G. (2017, June 10). Frameworks for mass collaboration, adaptable scripts, complex systems theory, and collaborative writing. *International Journal of Computer-Supported Collaborative Learning*, *12*(2), 127–131. https://doi.org/10.1007/s11412-017-9257-7

Marwan, N. (2008, October 26). A historical review of recurrence plots. *European Physical Journal: Special Topics*. Springer-Verlag. https://doi.org/10.1140/epjst/e2008-00829-1

Marwan, N. (2010). How to avoid potential pitfalls in recurrence plot based data analysis. *World Scientific*. https://doi.org/10.1142/S0218127411029008

Marwan, N., Carmen Romano, M., Thiel, M., & Kurths, J. (2007a). Recurrence plots for the analysis of complex systems. *Physics Reports*. https://doi.org/10.1016/j.physrep.2006.11.001

Marwan, N., Carmen Romano, M., Thiel, M., & Kurths, J. (2007b, January 1). Recurrence plots for the analysis of complex systems. *Physics Reports*. North-Holland. https://doi.org/10.1016/j.physrep.2006.11.001

Marwan, N., Donges, J. F., Zou, Y., Donner, R. V., & Kurths, J. (2009). Complex network approach for recurrence analysis of time series. *Physics Letters A*, *373*(46), 4246–4254. https://doi.org/10.1016/j.physleta.2009.09.042

Marwan, N., & Kurths, J. (2002). Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters, Section A: General, Atomic and Solid State Physics*, *302*(5–6), 299–307. https://doi.org/10.1016/S0375-9601(02)01170-2

Marwan, N., Thiel, M., & Nowaczyk, N. R. (2002). Cross recurrence plot based synchronization of time series. *ArXiv Preprint Physics/0201062*.

Matassini, L., Kantz, H., Hołyst, J., & Hegger, R. (2002). Optimizing of recurrence plots for noise reduction. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, *65*(2). https://doi.org/10.1103/PhysRevE.65.021102

Mindlin, G. B., & Gilmore, R. (1992). Topological analysis and synthesis of chaotic time series. *Physica D*, *58*, 229–242. Retrieved from https://www.sciencedirect.com/science/article/pii/016727899290111Y

Mitchell, M. (2009). *Complexity: A Guided Tour*. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. https://doi.org/10.1063/1.3326990

Ngamga, E. J., Buscarino, A., Frasca, M., Fortuna, L., Prasad, A., & Kurths, J. (2008). Recurrence analysis of strange nonchaotic dynamics in driven excitable systems. *Chaos*, *18*(1), 036222. https://doi.org/10.1063/1.2897312

Niazi, M. A. (2016). Introduction to the modeling and analysis of complex systems: a review. *Complex Adaptive Systems Modeling*, *4*(1), 3. https://doi.org/10.1186/s40294-016-0015-x

Nüssli, M.-A. (2011). Dual eye-tracking methods for the study of remote collaborative problem solving.

Nyström, M., & Holmqvist, K. (2010). An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data. *Behavior Research Methods*, *42*(1), 188–204. https://doi.org/10.3758/BRM.42.1.188

Orsucci, F., Giuliani, A., Webber, C., Zbilut, J., Fonagy, P., & Mazza, M. (2006). Combinatorics and synchronization in natural semiotics. *Physica A: Statistical Mechanics and Its Applications*, *361*(2), 665–676. https://doi.org/10.1016/j.physa.2005.06.044

Packard, N. H., Crutchfield, J. P., Farmer, J. D., & Shaw, R. S. (1980). Geometry from a time series. *Physical Review Letters*, *45*(9), 712–716. https://doi.org/10.1103/PhysRevLett.45.712

Poincaré, H. (1890). Sur le probléme des trois corps et les équations de la dynamique. *Acta Mathematica*, *13*(1), 5–7. https://doi.org/10.1007/BF02392506

Rayner, K. (1997). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, *124*(3), 372–422. https://doi.org/10.1037/0033-2909.124.3.372

Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *The Quarterly Journal of Experimental Psychology*, *62*(8), 1457–1506. https://doi.org/10.1080/17470210902816461

Robinson, G., & Thiel, M. (2009). Recurrences determine the dynamics. *Chaos*, *19*(2), 023104. https://doi.org/10.1063/1.3117151

Sauer, T., Yorke, J. a., & Casdagli, M. (1991). Embedology. *Journal of Statistical Physics*, *65*(3), 579–616. https://doi.org/10.1007/BF01053745

Sayama, H. (2015). *Introduction to the modeling and analysis of complex systems*. Retrieved from http://lib.hpu.edu.vn/handle/123456789/21480

Schinkel, S., Dimigen, O., & Marwan, N. (2008). Selection of recurrence threshold for signal detection. *European Physical Journal: Special Topics*, *164*(1), 45–53. https://doi.org/10.1140/epjst/e2008-00833-5

Stahl, G. (2006). *GROUP COGNITION Computer Support for Building Collaborative Knowledge*.

Takens, F. (1981). Detecting strange attractors in turbulence (pp. 366–381). https://doi.org/10.1007/BFb0091924

Thiel, M., Romano, M. C., & Kurths, J. (2003). Analytical description of Recurrence Plots of white noise and chaotic processes. *Izvestija Vyssich Ucebnych Zavedenij Prikladnaja Nelinejnaja Dinamika Applied Nonlinear Dynamics*, *11*(3), 14. Retrieved from https://arxiv.org/abs/nlin/0301027

Thiel, M., Romano, M. C., Kurths, J., Meucci, R., Allaria, E., & Arecchi, F. T. (2002). Influence of observational noise on the recurrence quantification analysis. *Physica D: Nonlinear Phenomena*, *171*(3), 138–152. https://doi.org/10.1016/S0167-2789(02)00586-9

Vaidyanathan, P., Pelz, J., Alm, C., Shi, P., & Haake, A. (2014). Recurrence quantification analysis reveals eye-movement behavior differences between experts and novices. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14* (pp. 303–306). https://doi.org/10.1145/2578153.2578207

Vickers, J. N. (2009). Advances in coupling perception and action: the quiet eye as a bidirectional link between gaze, attention, and action. *Progress in Brain Research*, *174*, 279–288. https://doi.org/10.1016/S0079-6123(09)01322-3

Webber, C. L., Ioana, C., & Marwan, N. (Eds.). (2016). *Recurrence Plots and Their Quantifications: Expanding Horizons* (Vol. 180). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-29922-8

Webber, C. L., & Zbilut, J. P. (1994). Dynamical assessment of physiological systems and

states using recurrence plot strategies. *Journal of Applied Physiology*, *76*(2), 965–973. https://doi.org/10.1152/jappl.1994.76.2.965

Webber, C. L., & Zbilut, J. P. (2005). Recurrence Quantification Analysis of Nonlinear Dynamical Systems. In *Tutorials in contemporary nonlinear methods for the behavioral sciences* (pp. 26–94). Retrieved from http://www.nsf.gov/sbe/bcs/pac/nmbs/nmbs.pdf

Webber Jr, C. L., & Marwan, N. (2015). Recurrence quantification analysis. *Theory and Best Practices*.

Whitney, H. (1936). Differentiable Manifolds. *The Annals of Mathematics*, *37*(3), 645. https://doi.org/10.2307/1968482

Wikipedia contributors. (2018). Poincare recurrence theorem --- {Wikipedia}{,} The Free Encyclopedia.

Zbilut, J. P., Giuliani, A., & Webber, C. L. (1998). Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, *246*(1–2), 122–128. https://doi.org/10.1016/S0375-9601(98)00457-5

Zbilut, J. P., & Marwan, N. (2008). The Wiener-Khinchin theorem and recurrence quantification. *Physics Letters, Section A: General, Atomic and Solid State Physics*, *372*(44), 6622–6626. https://doi.org/10.1016/j.physleta.2008.09.027

Zbilut, J. P., & Webber, C. L. (1992). Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A*, *171*(3–4), 199–203. https://doi.org/10.1016/0375-9601(92)90426-M

Zbilut, J. P., & Webber, C. L. (2007). Recurrence Quantification Analysis: Introduction and Historical Context. *International Journal of Bifurcation and Chaos*, *17*(10), 3477–3481. https://doi.org/10.1142/S0218127407019238

Zbilut, J. P., Zaldivar-Comenges, J. M., & Strozzi, F. (2002). Recurrence quantification based Liapunov exponents for monitoring divergence in experimental data. *Physics Letters, Section A: General, Atomic and Solid State Physics*, *297*(3–4), 173–181. https://doi.org/10.1016/S0375-9601(02)00436-X