

UNIVERZITA PALACKÉHO V OLOMOUCI  
PŘÍRODOVĚDECKÁ FAKULTA

**BAKALÁŘSKÁ PRÁCE**

Rekurentní kvantifikační analýza eye-tracking dat



**Katedra matematické analýzy a aplikací matematiky**

Vedoucí bakalářské práce: **Mgr. Kamila Fačevicová, Ph.D.**

Vypracovala: **Veronika Kalabusová**

Studijní program: B1103 Aplikovaná matematika

Studijní obor: Matematika–ekonomie se zaměřením na bankovníctví/pojišťovnictví

Forma studia: prezenční

Rok odevzdání: 2020

## BIBLIOGRAFICKÁ IDENTIFIKACE

**Autor:** Veronika Kalabusová

**Název práce:** Rekurentní kvantifikační analýza eye-tracking dat

**Typ práce:** Bakalářská práce

**Pracoviště:** Katedra matematické analýzy a aplikací matematiky

**Vedoucí práce:** Mgr. Kamila Fačevicová, Ph.D.

**Rok obhajoby práce:** 2020

**Abstrakt:** Rekurentní kvantifikační analýza je metoda, pomocí které můžeme jednoduše analyzovat dynamický systém. Cílem této bakalářské práce je popsat a interpretovat metody, jež rekurentní kvantifikační analýza využívá v případě eye-tracking dat a pomocí vytvořených funkcí ve statistickém softwaru R zpracovat reálnou datovou sadu.

**Klíčová slova:** rekurentní kvantifikační analýza, RQA, eye-tracking, rekurentní graf, míry rekurence, software R

**Počet stran:** 49

**Počet příloh:** 2

**Jazyk:** český

## BIBLIOGRAPHICAL IDENTIFICATION

**Author:** Veronika Kalabusová

**Title:** Recurrence quantification analysis of eye-tracking data

**Type of thesis:** Bachelor's

**Department:** Department of Mathematical Analysis and Applications of Mathematics

**Supervisor:** Mgr. Kamila Fačevicová, Ph.D.

**The year of presentation:** 2020

**Abstract:** Recurrence quantification analysis is a method which can be used to analyse dynamical systems with ease. The aim of this bachelor's thesis is to describe and interpret recurrence quantification analysis methods. These methods were applied to eye-tracking data using the R statistical software.

**Key words:** Recurrence quantification analysis, RQA, eye-tracking, recurrence plot, recurrence measures, software R

**Number of pages:** 49

**Number of appendices:** 2

**Language:** Czech

### **Prohlášení**

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně pod vedením paní Mgr. Kamily Fačevicové, Ph.D. a všechny použité zdroje jsem uvedla v seznamu literatury.

V Olomouci dne .....

.....

podpis



# Obsah

Úvod	9
<b>1 Eye-tracking</b>	<b>10</b>
1.1 Pohyby očí	10
1.2 Eye-tracking a jeho metody	11
1.3 Eye-tracker	12
1.4 Využití eye-trackingu	14
<b>2 Rekurentní kvantifikační analýza</b>	<b>16</b>
2.1 Rekurence	17
2.1.1 Metoda fixní mřížky	17
2.1.2 Metoda oblastí zájmu (AOI)	18
2.1.3 Metoda fixní vzdálenosti	18
2.2 Matice rekurence	19
2.3 Graf rekurence	21
2.4 Míry rekurence	22
2.4.1 Míra rekurence	22
2.4.2 Míra determinismu	23
2.4.3 Míra laminarity	24
2.4.4 CORM	26
2.4.5 Shrnutí	27
<b>3 Praktická část</b>	<b>28</b>
3.1 Shrnutí	43
Závěr	44
Literatura	45
Přílohy	

# Seznam obrázků

1.1	Pohyby oka – fixace a sádky	11
1.2	Purkyňovy obrázky	12
1.3	Korneální odraz	12
1.4	Eye-tracker brýle	13
1.5	Printscreen programu BeGaze	14
1.6	Eye-tracking laboratoř	14
2.1	Rozdělení obrázku metodou fixní mřížky	17
2.2	Rozdělení obrázku metodou oblastí zájmu	18
2.3	Metoda fixní vzdálenosti	19
2.4	Matice vzdálenosti a matice rekurence	20
2.5	Matice vzdálenosti a matice rekurence s dobou trvání	20
2.6	Grafy rekurence	21
2.7	Míra rekurence – R	22
2.8	Pohyby očí, které vytvoří diagonální linii	23
2.9	Diagonální linie délky 5	23
2.10	Pohyby očí, které vytvoří horizontální linii	25
2.11	Pohyby očí, které vytvoří vertikální linii	25
2.12	CORM	26
3.1	Ukázka naměřených dat	28
3.2	Data	31
3.3	Rozdělení mapy pomocí metody fixní mřížky	31
3.4	Rozdělení mapy pomocí metody AOI	32
3.5	Metoda fixní vzdálenosti	33
3.6	Matice rekurence bez doby trvání	34
3.7	Grafy rekurence bez doby trvání	36
3.8	Matice rekurence s dobou trvání	39
3.9	Grafy rekurence s dobou trvání	41

# Seznam tabulek

3.1	Míry rekurence bez doby trvání . . . . .	38
3.2	Míry rekurence s dobou trvání . . . . .	42

## **Poděkování**

Na tomto místě bych ráda poděkovala vedoucí své bakalářské práce Mgr. Kamile Fačevicové, Ph.D. za odborné vedení, pomoc a rady, které mi věnovala. Ráda bych také poděkovala své rodině a přátelům, kteří mě v průběhu celého studia velmi podporovali.

# Úvod

Eye-tracking v současnosti patří mezi velmi využívané technologie, jimiž lze rozpoznat, jakým způsobem lidé pozorují svět kolem sebe, zda při shodném zadání postupují všichni stejně a jestli je možné najít v jejich chování nějaký vzor.

Cílem této bakalářské práce je seznámit se s využitím rekurentní kvantifikační analýzy při popisu eye-tracking dat. Hlavním úkolem je pomocí vlastních funkcí v programovacím prostředí R ukázat, jak rekurentní kvantifikační analýza funguje.

Celou práci lze rozdělit do tří kapitol. V první kapitole se nejprve seznámíme s tím, jaké pohyby provádí lidské oko, co je to eye-tracking a jaké metody pro sledování pohybu očí existují. Poté uvedeme, jaké zařízení se pro eye-tracking využívá a který software je možné pro následnou analýzu dat použít. Na závěr této části představíme několik oblastí, kde je eye-tracking často využíváný.

Ve druhé kapitole představíme rekurentní kvantifikační analýzu jako metodu určenou ke zkoumání dynamických systémů. Zjistíme, co je to rekurence, jakými metodami ji můžeme stanovit a na příkadech ukážeme míry rekurence, které se k charakterizování dat v této analýze používají.

V závěrečné kapitole se budeme věnovat praktické části, ve které aplikujeme poznatky z teoretické části na reálná data. Na konkrétním příkladu ukážeme, jak se dá rekurentní kvantifikační analýza využít k popisu eye-tracking dat z oblasti kartografie.

# Kapitola 1

## Eye-tracking

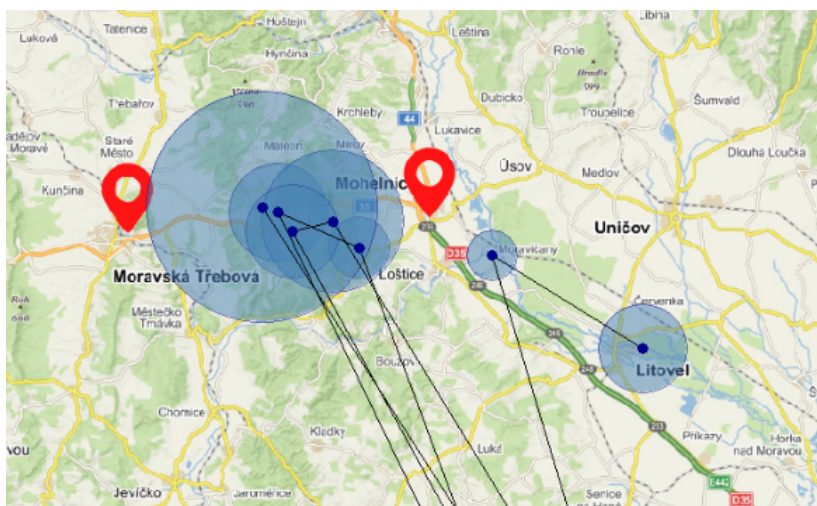
### 1.1. Pohyby očí

Jedna z nejdůležitějších schopností oka je jeho zaměření se na určité místo. Tuto schopnost označujeme jako fixaci. Délka jejího trvání může být od několika desítek milisekund až po několik sekund. Pojem „fixace“ však není naprosto přesné označení, neboť oko není nikdy plně fixováno. Během jednotlivých fixací provádí člověk tři druhy mikropohybů [24].

Prvním druhem mikropohybů je tremor (třes). Jedná se o velmi malé pohyby, jejichž přesný význam není dosud znám, ale je pravděpodobné, že jsou způsobeny nepřesnou kontrolou okohybných svalů. Druhým typem je drift, což jsou pohyby, které posunují oko mimo střed fixace. Posledním druhem je mikrosakáda, jejímž úkolem je vracení oka zpět do středu fixace. Mikropohyby, které probíhají v průběhu fixací není možné zachytit pouhým okem [11]. Můžeme je zachytit pouze s využitím vysokofrekvenčního eye-trackeru a jejich studiem se zabývá spíše neurologie.

Rychlé pohyby z jedné fixace na další jsou označovány jako sakády. Dle [11] se jedná o nejrychlejší pohyby, které může lidské tělo vyvinout, jejich běžná délka je 30 - 80 ms a mohou být znázorněny pomocí linií mezi jednotlivými fixacemi.

V této práci se budeme zabývat pouze fixacemi jako takovými (bez jednotlivých mikropohybů) a sakádami, jak můžeme vidět na obrázku 1.1, kde jsou znázorněny základní pohyby očí. Stejným způsobem probíhá naprostá většina kartografických výzkumů.



Obrázek 1.1: Znázornění pohybů oka, kterým se budeme věnovat. Fixace představují modré body na mapě, kruhy kolem jednotlivých bodů zobrazují délku fixace. Sakády jsou znázorněny liniemi mezi fixacemi.

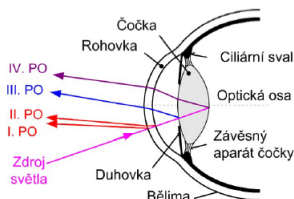
## 1.2. Eye-tracking a jeho metody

Eye-tracking je technologie, která nám umožňuje zaznamenávání pohybu očí. Dává nám informace o tom, kam, v jakém pořadí a na jak dlouhou dobu se daný člověk dívá. Podle [8] definujeme několik technik, které jsou ke sledování pohybu očí používány.

V 70. letech 20. století byla nejpoužívanější metodou elektrookulografie (je používána i v současnosti). Jedná se o techniku, kdy byly respondentovi rozmístěny elektrody kolem očí a poté se měřil rozdíl elektrického odporu kůže [39]. Tuto techniku není možné využít pro zjišťování směru pohledu, protože měříme pouze pohyby očí vzhledem k poloze hlavy. Pokud bychom současně měřili i pozici hlavy, pak by bylo možné ji využít. Jak uvádí [38], tato technika je ze všech uvedených metod nejméně přesná.

Fotookulografie nebo videookulografie jsou první z neinvazivních technik měření očního pohybu. Spočívají v měření pozice části oka, které lze rozlišit. Zejména je používáno zjišťování tvaru zornice, pozice limbu (hranice mezi rohovkou a bělmem) - nevýhodou je poměrně časté překrývání očními víčky. Dále se používá korneální (kornea = rohovka) odraz. Manuální procházení videozáznamu je velmi zdlouhavé a náchylné k chybám. Některé z těchto metod navíc vyžadují, aby byla hlava zafixovaná - pomocí opěrky hlavy nebo pomocí fixace zubů.

Pravděpodobně nejpoužívanější technikou je Pupil and Corneal Reflection Tracking. Tato metoda je založená na detekci středu zornice a korneálního odrazu paprsku infračerveného světla pomocí kamery. Dle vzájemné polohy těchto dvou bodů následně zařízení určí směr pohledu. Korneální odraz je označení pro první Purkyňův obrázek (obrázek 1.2)[8].

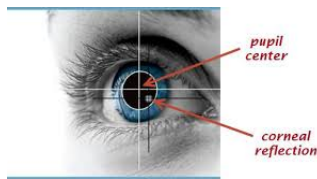


Obrázek 1.2: Čtyři Purkyňovy obrázky, ukazující odrazy světelného paprsku od různých částí oka - I.PO od vnější strany rohovky, II.PO od vnitřní strany rohovky, III.PO od přední strany čočky a IV.PO od zadní strany čočky [13].

### 1.3. Eye-tracker

Ke sledování pohybu očí se používá zařízení, které se nazývá eye-tracker. V současné době jsou eye-trackery založené na principu snímání jednoho či obou očí vůči vizuálnímu vjemu pomocí videa. Sledovanému subjektu ukážeme na zobrazovacím zařízení objekt, například obrázek, text, internetovou stránku či jiný obraz a zaznamenáváme pohyby očí respondenta.

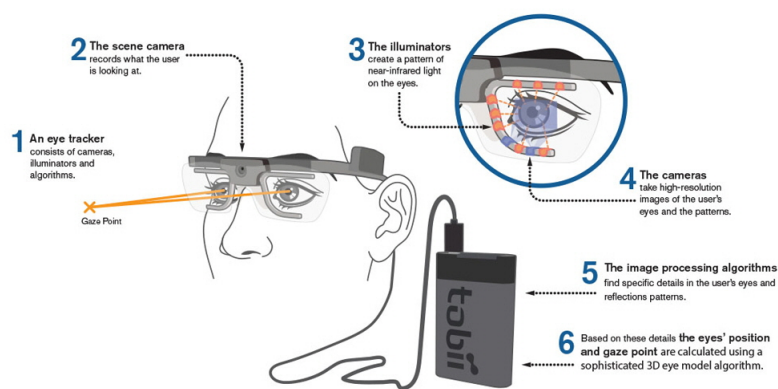
Eye-trackery se skládají z krabičky, která je umístěna pod monitorem a vysokorychlostní kamery, jež snímá oči respondenta. V krabičce jsou dvě infračervená světla, která svítí směrem k uživateli. Infračervené záření je zvoleno z důvodu, že je pro člověka neviditelné a nejlépe se odráží od sítnice lidského oka.



Obrázek 1.3: Snímek ukazující korneální odraz-označený černým křížkem a střed zornice-označený bílým křížkem [7].



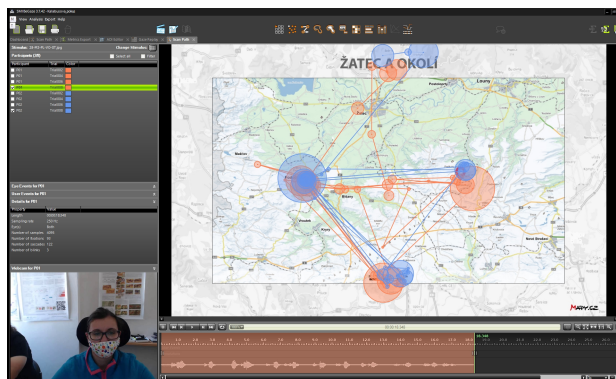
Nemusí se však vždy jednat o statické zařízení. Mnohé využití mají i mobilní eye-trackery, které umožňují sledování pohybu očí přímo ve skutečném prostředí. Eye-trackery jsou umístěny na hlavě respondenta - může se jednat například o brýle nebo eye-trackery připevněné na helmu (viz obrázek 1.4). Mobilní eye-trackery využívají, stejně jako ty statické, Pupil and Corneal Reflection tracking [4].



Obrázek 1.4: Eye-tracker brýle od společnosti Tobii [29].

V současné době jsou nejčastěji využívány eye-trackery značky Tobii a SMI (SensoMotoric Instruments).

Z naměřených hodnot získáme pomocí souřadnicového systému X,Y pouze informace o středu zornice a směru pohledu po dobu trvání testu [17]. Pro další rozbor je nezbytné data zpracovat v softwaru, který je pro tuto činnost určený. V případě eye-trackerů SMI je systémem pro zpracování dat program BeGaze. Základním výstupem tohoto programu je tzv. gaze plot - graf, který zobrazuje fixace oka a sakády (obrázek 1.5) [23]. BeGaze následně umožňuje exportování dat pro další statistickou analýzu formou textového dokumentu. Také je možné exportovat celý video záznam testu (příloha 2).



Obrázek 1.5: Printsreen softwaru BeGaze užívaného pro eye-trackery značky SMI.

Na obrázku 1.6 je možné vidět vybavení eye-tracking laboratoře na Katedře geoinformatiky UP v Olomouci. V laboratoři je doporučeno dodržovat několik základních pravidel. Například je vhodné, aby v zorném poli respondenta bylo umístěno co nejméně rušivých aspektů, aby bylo možné plně se soustředit na monitor. Dále je důležité, aby nesvítilo žádné světlo přímo do eye-trackeru, protože by docházelo ke zkreslení dat.



Obrázek 1.6: Eye-tracking laboratoř na Katedře geoinformatiky UP v Olomouci

## 1.4. Využití eye-trackingu

Jednou z mnoha oblastí, kde je možné eye-tracking využít, je sport. Eye-tracking se ve sportu používá především k lepšímu pochopení individuálního

jednání sportovce tím, že identifikuje souvislosti mezi zaměřením pozornosti, odhadem trajektorie, strategiemi vizuálního vyhledávání, koordinací rukou a očí a činnostmi jednotlivce během hry.

Jak uvádí [6], Mark Campbell a Aiden P. Moran v roce 2014 zjistili, že je velmi výrazný rozdíl mezi amatérským a profesionálním golfistou - například ve způsobu zkoumání greenu před puttem. Profesionál využívá více ekonomický přístup a to tím, že používá méně fixací, které trvají déle. Amatér využil k prohlížení greenu velký počet krátkých fixací. Toto a mnohé další zjištění mohou pomoci ke zvýšení úrovně jednotlivých sportovců.

Další z oblastí častého využití eye-trackingu je marketing a reklama. Poskytuje nám užitečné údaje o přirozeném chování spotřebitelů a odhaluje, co přitahuje jejich pozornost. Pomáhá nám zjistit, jak nakupující v obchodech postupují při prohlížení regálů s produkty, což může firmám pomoci určit, do jakých míst je nejefektivnější produkty umístit [30]. Hlavním cílem je porozumět chování spotřebitelů. Dalším z příkladů využití je Lohseho analýza reklam ve Zlatých stránkách [19]. Otázkou bylo, jestli člověk sleduje reklamy v určitém pořadí, co způsobuje, že ho určitá reklama zaujme a jaký čas dané reklamě věnuje. Výsledek analýzy říká, že barva, velikost i vzhled reklamy mají vliv na čtenáře a tím pádem i na její úspěšnost.

V dopravě je dokázáno, že většina nehod je způsobena problémy s vizuální pozorností [9]. Eye-tracking nám může pomoci při analýze chování řidiče v různých situacích - nejen při rozptýlení spolujezdcem, ale i po užití drog nebo alkoholu. Výsledky pomáhají určit, které prvky mohou řidiče nejvíce rozptylovat. Studie zaměřené na řidiče jsou velmi často prováděny na simulátorech, z důvodu bezpečnosti a možnosti kontroly prostředí. Nemusíme se omezit pouze na řízení auta. Eye-tracking se využívá také při sledování chování letových dispečerů, pilotů či řízení vlaků [18]. To má dopad na zvýšení bezpečnosti v dopravě.

Významným odvětvím, kde můžeme najít využití eye-trackingu je kartografie. Na základě výsledků výzkumu je možné neustále upravovat mapy tak, aby byly co nejlépe použitelné pro uživatele. Podle [25] řešíme při analýze map následující otázky. Jak čtenář sleduje informace v mapě, jak rychle dané informace získá, které prvky ho zaujmou nejdříve, kolikrát se pohledem vrací do legendy, které mapové prvky jsou snadno a které naopak obtížně čitelné, jakým způsobem ovlivňuje čitelnost mapy grafický design apod. Odpovědi na tyto otázky nám mohou pomoci například i v případě plánování, nebo řešení kritických situací. Další možnosti příkladů využití eye-trackingu v oblasti kartografie jsou popsány v publikaci [24].

## Kapitola 2

# Rekurentní kvantifikační analýza

Rekurentní kvantifikační analýza (RQA, Recurrence Quantification Analysis) je poměrně mladá matematická metoda analýzy dat. Základní myšlenkou této analýzy je popsat a kvantifikovat určité opakující se vzorce v časových řadách - například se může jednat o stejnou výšku přílivové vlny v analýze přílivu a odlivu, sledování stejného místa na obrázku při analýze pohybu očí či zkoumání stejné hodnoty kurzů na trhu akcií [15].

Na konci 19. století Henri Poincaré ve své práci uvedl, že jedním ze základních rysů dynamických systémů je opakování (rekurence) a lze jej využít při jejich charakteristice. Průkopová práce Poincarého a jeho objev rekurence si však musel víc než 70 let počkat na využití, které bylo podmíněno vývojem rychlejších a výkonnějších počítačů [20].

Důležitým okamžikem byl rok 1987, kdy J. P. Eckmann s kolegy představil metodu rekurentního grafu jako nástroje k vizualizaci rekurence v dynamických systémech (časových řadách).

Interpretace rekurentního grafu však závisí na schopnostech jedince a je tedy příliš subjektivní [16]. Možnost, jak eliminovat tento problém představili na začátku 90. let 20. století Joseph P. Zbilut a Charles L. Webber. Jednalo se o rekurentní kvantifikační analýzu, s jejíž pomocí je možné kvantifikovat a jednoduše popsat informace, které jsou v rekurentním grafu obsaženy.

Hlavní výhodou rekurentní kvantifikační analýzy je její jednoduchost a použitelnost na krátké časové řady. To umožňuje využití v mnoha různých disciplínách, jako je fyzika, chemie, ekonomie a inženýrství. Například ve sportu byla RQA použita při zkoumání serií přihrávek hráčů jednotlivých týmů na Mistrovství světa ve fotbale v roce 2006, pro nastínění taktiky a organizace hry jednotlivých týmů [5].

V publikaci [1] Nicola C. Anderson popisuje a ukazuje, jak rekurentní kvantifikační analýza a její míry mohou být použity při charakteristice chování pohybu očí. RQA v tomto případě určuje, jak často se respondenti vrací

pohledem do již dříve prohlédnuté oblasti obrázku a poskytuje údaje o předvídatelnosti následného očního pohybu. Především z Andersonovy práce budeme v této kapitole vycházet, dále jsme vycházeli z publikací [21], [32], [33], [34].

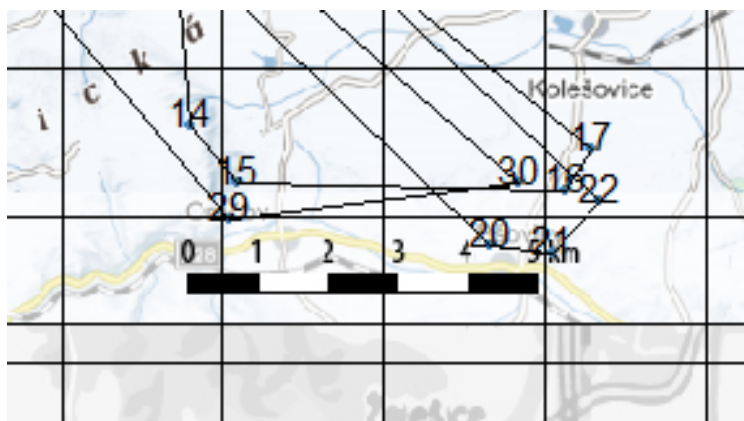
## 2.1. Rekurence

Uvažujme posloupnost fixací  $f_i, i = 1, \dots, N$ , kde každá fixace má svoji x-ovou a y-ovou souřadnici, tj.  $f_i = \langle x_i, y_i \rangle$  a svoji dobu trvání  $t_i, i = 1, \dots, N$ . Dvě fixace  $f_i$  a  $f_j$  považujeme za rekurentní (opakující se), pokud jsou obě dostatečně blízko u sebe. Existuje však několik metod, pomocí kterých můžeme určit, zda jsou dvě fixace poblíž sebe.

### 2.1.1. Metoda fixní mřížky

První možností, jak určit, zda se dvě fixace opakují, je využití metody fixní mřížky. Spočívá v tom, že obrázek nejprve rozdělíme pomocí mřížky na stejně velké čtverce. Fixace  $f_i$  a  $f_j$  považujeme za rekurentní, pokud obě fixace leží ve stejném čtverci. Jak je vidět na obrázku 2.1, fixace 16 je rekurentní s fixacemi 17 a 22, jelikož leží ve stejném čtverci.

Nevýhodou této metody je to, že mřížku definujeme bez ohledu na obsah obrázku, což může způsobit, že v některých místech zájmu je rozdělení příliš hrubé, nebo naopak až příliš jemné a také to, že 2 fixace nemusí být rekurentní, i když jsou téměř u sebe - příklad můžeme vidět na obrázku 2.1, kdy fixace 20 a 21 jsou blízko u sebe, a přesto nejsou považovány za rekurentní.

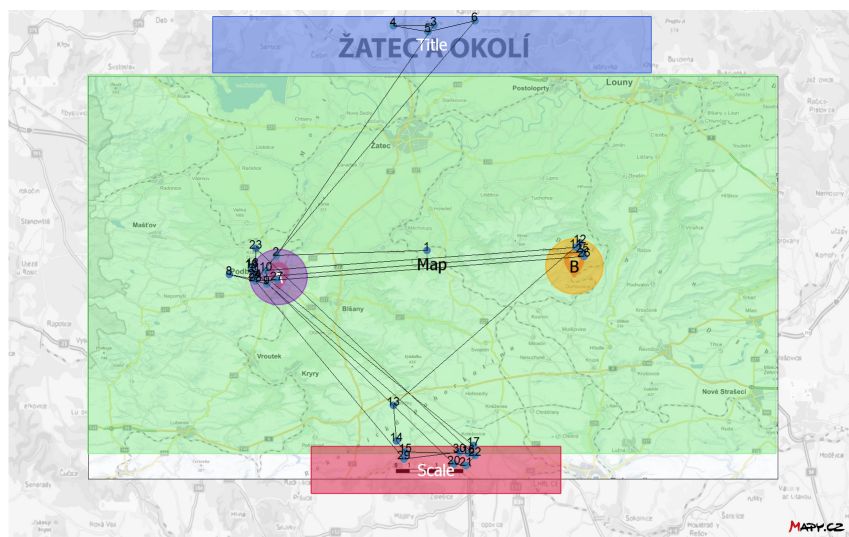


Obrázek 2.1: Rozdělení obrázku pomocí fixní mřížky. Rozměry čtverců jsou 64x64 pixelů.

### 2.1.2. Metoda oblastí zájmu (AOI)

Tato technika řeší nevýhodu metody fixní mřížky, kdy rozdělujeme obrázek bez ohledu na jeho obsah. Metoda AOI spočívá v definování oblastí zájmu - významných částí obrázku (areas of interest). Poté postupujeme stejně jako v případě fixní mřížky, kdy dvě fixace  $f_i$  a  $f_j$  považujeme za opakující se, pokud obě fixace leží ve stejné oblasti zájmu. Na obrázku 2.2 je možné vidět příklad rozdělení obrázku na oblasti zájmu - v našem případě se jedná o rozdělení mapy na titulek (modrá), mapu (zelená), měřítko (červená), bod A (fialová), bod B (žlutá) a bílé místo.

U této metody je nevýhodou to, že 2 fixace jsou rekurentní, i v případě, kdy leží daleko od sebe. A navíc stejně jako u metody fixní mřížky nemusí být 2 fixace rekurentní, i když leží téměř u sebe.



Obrázek 2.2: Rozdělení obrázku - metoda oblastí zájmu.

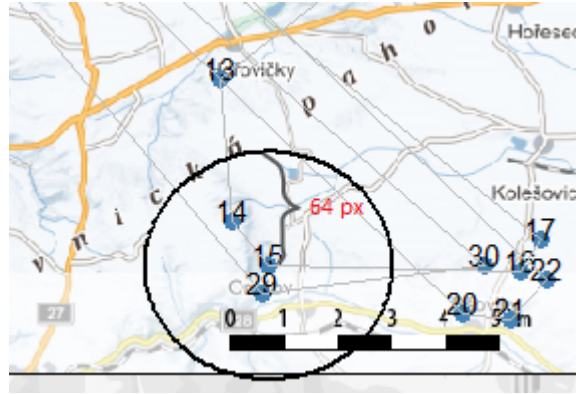
### 2.1.3. Metoda fixní vzdálenosti

Nedostatky obou předchozích metod při stanovení rekurence eliminuje metoda fixní vzdálenosti. Pomocí této metody definujeme dvě fixace  $f_i$  a  $f_j$  jako opakující se, jestliže  $d(f_i, f_j) \leq \rho$ , kde  $d$  je zvolená metrika vzdálenosti (např. listonošská, eukleidovská, ...) a  $\rho$  je zvolený poloměr. Pokud zvolíme eukleidovskou metriku, tak je vzdálenost  $d$  mezi dvěma fixacemi dána vztahem

$$d(f_i, f_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \text{ kde } i, j = 1, \dots, N.$$



Na obrázku 2.3 můžeme vidět fixaci 15, která je rekurentní s fixacemi 14 a 29, jelikož jsou ve vzdálenosti menší než zvolený poloměr - 64 pixelů.



Obrázek 2.3: Rekurence fixací - metoda fixní vzdálenosti.

## 2.2. Matice rekurence

Matice rekurence je čtvercová matice  $N \times N$ , která obsahuje pouze hodnoty 0 nebo 1. Prvek rekurentní matice  $r_{ij}$  je roven 1, pokud jsou fixace  $f_i$  a  $f_j$  rekurentní. Pokud fixace  $f_i$  a  $f_j$  nejsou rekurentní, pak je prvek  $r_{ij}$  roven 0.

V případě, že si zvolíme metodu fixní vzdálenosti při zjišťování rekurentních bodů, je důležitým krokem před tvorbou matice rekurence vytvoření matice vzdálenosti. Jedná se o čtvercovou matici  $N \times N$ , jejíž prvek  $d_{ij}$  značí vzdálenost fixací  $f_i$  a  $f_j$ . Vzdálenostní matice je symetrická dle hlavní diagonály, to znamená, že  $d(f_i, f_j) = d(f_j, f_i)$ . Ze vzdálenostní matice poté vytvoříme matici rekurence.

Prvky rekurentní matice  $r_{ij}$  definujeme jako

$$r_{ij} = \begin{cases} 1, & d(f_i, f_j) \leq \rho \\ 0, & \text{jinak} \end{cases} \quad (2.1)$$

kde  $d$  je zvolená metrika vzdálenosti (např. listonožská, eukleidovská, ...) a  $\rho$  je zvolený poloměr.

Na obrázku 2.4 můžeme vlevo vidět matici vzdáleností pro prvních deset fixací. Vzdálenost byla počítána pomocí eukleidovské metriky. Následně jsme vytvořili matici rekurence, která má na pozici  $r_{ij}$  hodnotu 1, pokud je vzdálenost mezi fixacemi  $f_i$  a  $f_j$  menší než zvolený poloměr - 64 pixelů.

↑	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	↑	1	2	3	4	5	6	7	8	9	10	
1	0	312	507	510	491	526	379	412	339	336	1	1	0	0	0	0	0	0	0	0	0	0
2	312	0	613	571	593	669	83	106	67	40	2	0	1	0	0	0	0	0	0	0	0	1
3	507	613	0	84	20	85	692	703	677	652	3	0	0	1	0	1	0	0	0	0	0	0
4	510	571	84	0	73	169	647	654	637	610	4	0	0	0	1	0	0	0	0	0	0	0
5	491	593	20	73	0	100	672	683	657	633	5	0	0	1	0	1	0	0	0	0	0	0
6	526	669	85	169	100	0	750	764	731	709	6	0	0	0	0	0	1	0	0	0	0	0
7	379	83	692	647	672	750	0	37	43	46	7	0	0	0	0	0	0	1	1	1	1	1
8	412	106	703	654	683	764	37	0	81	76	8	0	0	0	0	0	0	1	1	0	0	0
9	339	67	677	637	657	731	43	81	0	30	9	0	0	0	0	0	0	1	0	1	1	1
10	336	40	652	610	633	709	46	76	30	0	10	0	1	0	0	0	0	1	0	1	1	1

Obrázek 2.4: Vlevo matice vzdálenosti, vpravo matice rekurence - bez uvažování doby trvání.

Pokud budeme uvažovat dobu trvání fixací, potom bude mít matice rekurence prvky  $r_{ij}$ , které definujeme jako

$$r_{ij}^t = \begin{cases} t_i + t_j, & d(f_i, f_j) \leq \rho \\ 0, & \text{jinak} \end{cases} \quad (2.2)$$

kde  $d$  je zvolená metrika vzdálenosti (např. listonožská, eukleidovská, ...),  $\rho$  je zvolený poloměr a  $t_i, t_j$  označují dobu trvání  $i$ -té a  $j$ -té fixace.

V případě, že uvažujeme dobu trvání, pak u matice rekurence vycházíme ze stejné matice vzdáleností, neboť vzdálenost mezi fixacemi se nezměnila, u matice rekurence však na pozici  $r_{ij}$  bude hodnota, která odpovídá součtu doby trvání fixací  $f_i$  a  $f_j$ . Na obrázku 2.5 vidíme matici vzdáleností (vlevo), pro prvních deset fixací, ze které opět vycházíme při tvorbě matice rekurence, která nyní obsahuje místo hodnot 1 a 0 součty dob trvání jednotlivých fixací (vpravo), pokud jsou ve vzdálenosti menší než 64 pixelů.

↑	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	↑	1	2	3	4	5	6	7	8	9	10	
1	0	312	507	510	491	526	379	412	339	336	1	559	0	0	0	0	0	0	0	0	0	0
2	312	0	613	571	593	669	83	106	67	40	2	0	288	0	0	0	0	0	0	0	0	315
3	507	613	0	84	20	85	692	703	677	652	3	0	0	761	0	855	0	0	0	0	0	0
4	510	571	84	0	73	169	647	654	637	610	4	0	0	0	750	0	0	0	0	0	0	0
5	491	593	20	73	0	100	672	683	657	633	5	0	0	855	0	948	0	0	0	0	0	0
6	526	669	85	169	100	0	750	764	731	709	6	0	0	0	0	0	2503	0	0	0	0	0
7	379	83	692	647	672	750	0	37	43	46	7	0	0	0	0	0	0	328	284	242	335	335
8	412	106	703	654	683	764	37	0	81	76	8	0	0	0	0	0	0	284	241	0	0	0
9	339	67	677	637	657	731	43	81	0	30	9	0	0	0	0	0	0	242	0	156	249	249
10	336	40	652	610	633	709	46	76	30	0	10	0	315	0	0	0	0	335	0	249	342	342

Obrázek 2.5: Vlevo je matice vzdálenosti, vpravo matice rekurence s dobou trvání.



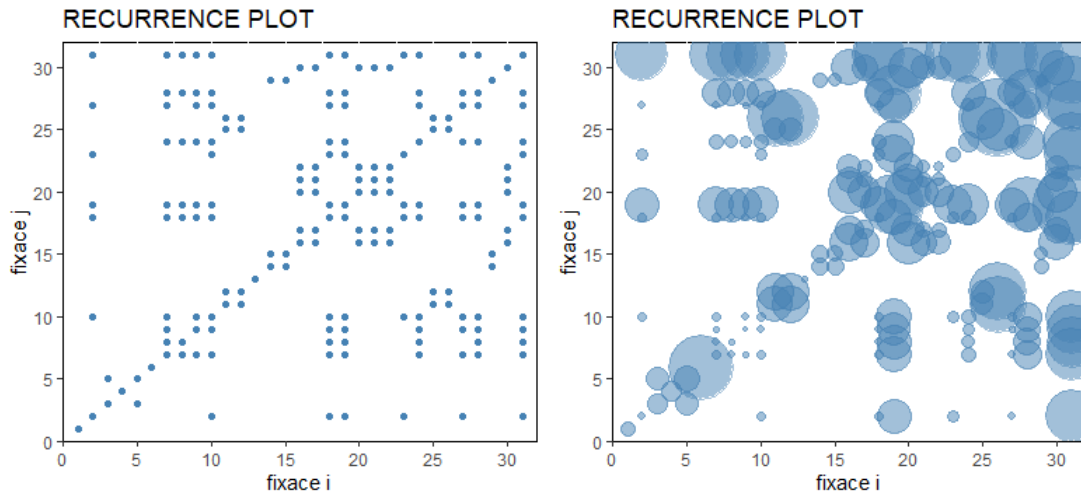
## 2.3. Graf rekurence

Graf rekurence je nástroj, který slouží k vizuálnímu zobrazení rekurentních bodů. Jedná se o grafické znázornění rekurentní matice. Pokud se fixace  $f_i$  a  $f_j$  opakují ( $r_{ij} = 1$ ), pak je na pozici  $i, j$  v grafu vykreslena barevná tečka. V případě, že fixace  $f_i$  a  $f_j$  nejsou rekurentní ( $r_{ij} = 0$ ), tak na pozici  $i, j$  není vykresleno nic (obrázek 2.6 - levý graf). Pro každou posloupnost fixací generujeme samostatný rekurentní graf.

Rekurentní graf má vždy zvýrazněnou hlavní diagonálu, jelikož porovnáváme fixace se sebou samými ( $i = j$ ). Protože jsou metriky vzdálenosti symetrické, tak jsou symetrické i rekurentní grafy - dle hlavní diagonály, která se nazývá incidenční linie.

Fixace jsou vykreslovány postupně, čím větší je vzdálenost mezi rekurentním bodem a hlavní diagonálou, tím delší je doba (určená počtem fixací), mezi původní fixací a fixací opětvnou. Například pokud jsou fixace 2 a 20 rekurentní, pak to znamená, že jsme si místo prohlédli jako 2. v pořadí (téměř ihned na začátku) a poté jsme se na stejné místo vrátili až ve 20. fixaci.

Pokud budeme uvažovat dobu trvání, pak jsou-li fixace  $f_i$  a  $f_j$  rekurentní, je velikost bodu v rekurentním grafu na pozici  $i, j$  dána součtem doby trvání fixací  $f_i$  a  $f_j$ . Na obrázku 2.6 (vpravo) odpovídá velikost bodu.



Obrázek 2.6: Vlevo graf rekurence bez doby trvání. Vpravo graf rekurence s dobou trvání.

## 2.4. Míry rekurence

Rekurentní graf nám poskytuje pouze vizuální prezentaci opakujících se bodů v posloupnosti fixací. V případě, že budeme chtít srovnat různé respondenty, různé podmínky nebo různá zadání, je nutné rekurentní graf doplnit o kvantitativní míry rekurence. Jedná se o míry, které nám umožní charakterizovat chování systému [10].

Následující charakteristiky se dají využít pro analýzu sledování očí a každá z nich má v této oblasti přesně danou interpretaci. Kompletní seznam měr, které RQA využívá je možné najít v publikaci [26].

Z důvodu symetrie rekurentního grafu můžeme kvantitativní míry zjišťovat pouze z hodnot v horním trojúhelníku nad diagonálou a zároveň můžeme vynechat i hlavní diagonálu (linie incidence), protože nám nedává žádné nové informace – pouze říká, že každá fixace je sama se sebou rekurentní.

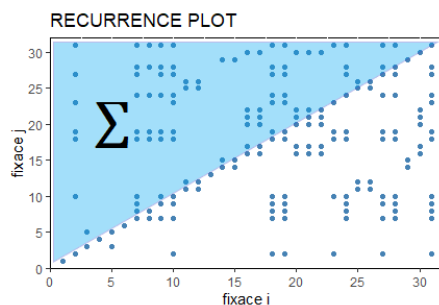
### 2.4.1. Míra rekurence

První charakteristikou, kterou budeme využívat, je míra rekurence, představující pro posloupnost  $N$  fixací procento bodů, které se opakují. Můžeme říct, že se jedná o míru, která určuje, jak často se pozorovatelé opakovaně dívají na již v minulosti prohlédnutý bod [37].

Míru rekurence definujeme jako

$$REC = 100 \frac{2R}{N(N-1)}, \quad (2.3)$$

kde  $R$  je součet opakujících se bodů v horním trojúhelníku v rekurentním grafu,  $R = \sum_{i=1}^{N-1} \sum_{j=i+1}^N r_{ij}$  (obrázek 2.7) a  $N$  je počet fixací, které v testu provedeme.



Obrázek 2.7:  $R$  představuje součet bodů v horním trojúhelníku rekurentního grafu.

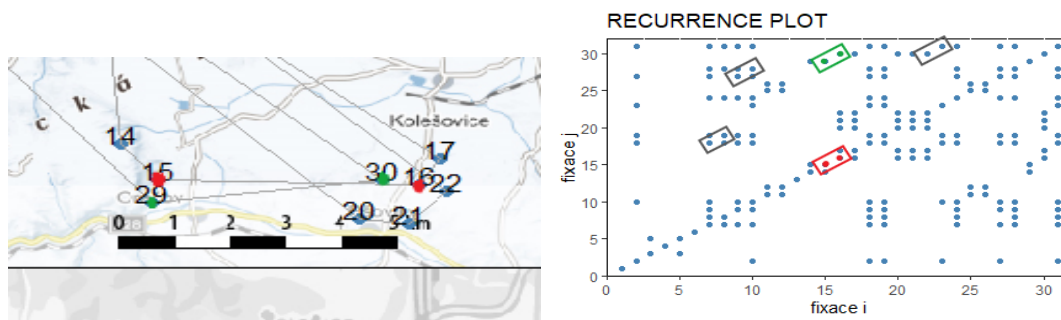
Pokud budeme uvažovat délku doby trvání fixací, pak míra rekurence označuje, jaké procento času respondent věnoval rekurentním fixacím a můžeme ji předefinovat jako

$$REC^t = 100 \frac{R^t}{(N-1)T}, \quad (2.4)$$

kde  $R^t$  je součet dob trvání opakujících se fixací v horním trojúhelníku,  $R^t = \sum_{i=1}^{N-1} \sum_{j=i+1}^N r_{ij}^t$  a  $T$  je celková doba všech fixací,  $T = \sum_{i=1}^N t_i$ .

## 2.4.2. Míra determinismu

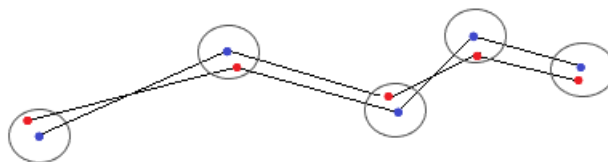
Determinismus nám poskytuje údaje o předvídatelnosti systému. Míra determinismu představuje podíl opakujících se bodů, které tvoří diagonální linie a vyjadřuje, jak často lidé při sledování obrázku opakují určité vzory.



Obrázek 2.8: Pohyby očí, které vytvořily diagonální linii délky 3.

Minimální délka diagonální linie byla stanovena na hodnotu  $L = 2$ . Délka diagonální linie vyjadřuje počet fixací, které tvoří opakující se vzory pohybu očí. Na obrázku 2.8 můžeme vidět, že fixace 29 a 30 (zelená barva) vytvořily v rekurentním grafu diagonální linii délky 2, protože opakovaly stejný vzorec pohybu očí, jako fixace 15 a 16 (červená barva).

Ukázka možného vzoru, který by vytvořil diagonální linii délky  $L = 5$  je na obrázku 2.9.



Obrázek 2.9: Vzor pohledu pro diagonálu délky 5.

Míru determinismu definujeme jako

$$DET = 100 \frac{|D_L|}{R}, \quad (2.5)$$

kde  $|D_L|$  označuje počet bodů v horním trojúhelníku v rekurentním grafu, které tvoří diagonální linii s minimální délkou  $L$  a  $R$  je součet opakujících se bodů v horním trojúhelníku v rekurentním grafu (viz míra rekurence).

Pokud budeme uvažovat dobu trvání fixací, pak míra determinismu označuje procento času rekurentních fixací, které opakovaly nějaký vzor. Míru determinismu můžeme předefinovat jako

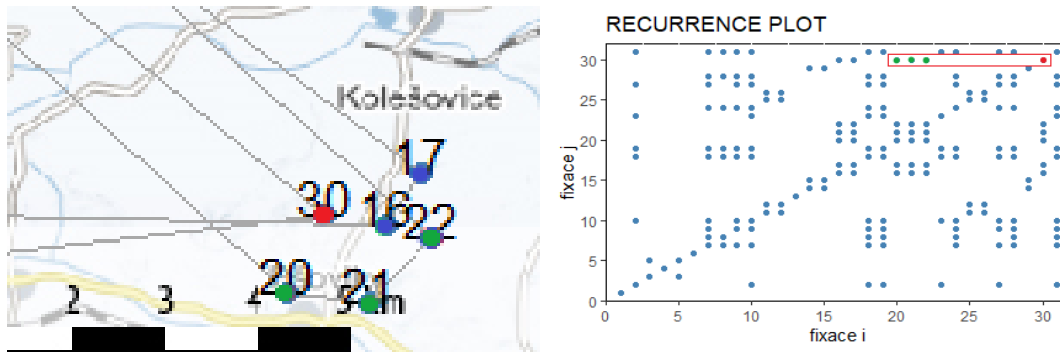
$$DET^t = \frac{100}{R^t} \sum_{(i,j) \in D_L} r_{ij}^t, \quad (2.6)$$

kde  $R^t$  je součet dob trvání opakujících se fixací v horním trojúhelníku, a suma označuje součet prvků matice rekurence, které tvoří diagonální linie.

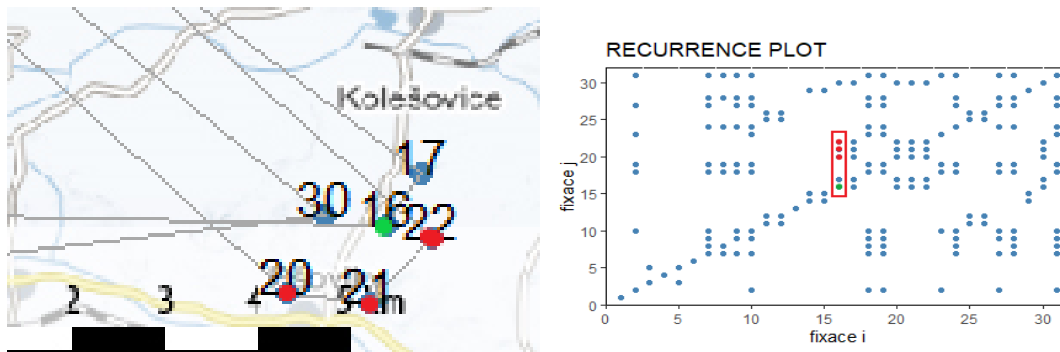
### 2.4.3. Míra laminarity

Laminarita obecně ukazuje ty oblasti obrázku, které jsou prohlíženy opakovaně. Míra laminarity je ovlivňována horizontálními a vertikálními liniemi. Vodorovné linie představují oblasti, které jsme nejdříve pozorovali do detailu a později jsme se k nim letmo vrátili - na obrázku 2.10 jsme nejdříve detailněji pozorovali místo při fixacích 20, 21 a 22 (zelená barva) a poté jsme se do této oblasti jen rychle vrátili při fixaci 30 (červená barva), to způsobilo vytvoření horizontální linie délky 3 v rekurentním grafu.

Naopak svislé linie představují oblasti, které jsme si nejdříve letmo prohlédli, později se k nim vrátili a zkoumali je více podrobně. To můžeme vidět na obrázku 2.11, kdy jsme si místo rychle prohlédli - fixace 16 (zelená barva), později jsme se do oblasti vrátili - fixace 20, 21, 22 (červená barva) a pozorovali místo detailněji.



Obrázek 2.10: Pohyby očí, které vytvořily horizontální linii délky 3.



Obrázek 2.11: Pohyby očí, které vytvořily vertikální linii délky 3.

Opět byla stanovena minimální délka linií na hodnotu  $L = 2$ . V případě, že graf rekurence obsahuje větší shluky opakujících se bodů (s horizontálními i vertikálními liniemi), toto naznačuje detailní prohlížení oblasti a jejího okolí.

Míru laminarity definujeme jako

$$LAM = 100 \frac{|H_L| + |V_L|}{2R}, \quad (2.7)$$

kde  $|H_L|$  označuje počet bodů v horním trojúhelníku v rekurentním grafu, které tvoří horizontální linii s minimální délkou  $L$ ,  $|V_L|$  označuje počet bodů v horním trojúhelníku rekurentního grafu, které tvoří vertikální linii s minimální délkou  $L$  a  $R$  je součet opakujících se bodů v horním trojúhelníku v rekurentním grafu (viz míra rekurence).

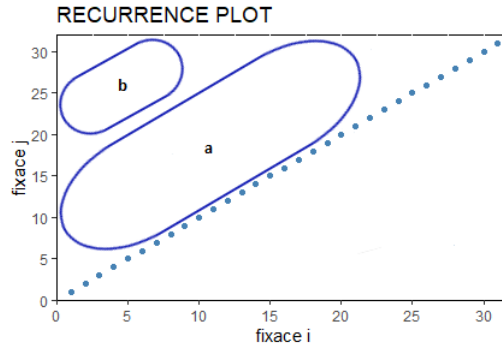
Pokud budeme uvažovat dobu trvání fixací, pak míra laminarity vyjadřuje procento celkového času rekurentních fixací, který byl věnován opětovnému detailnímu prohlížení vybraných oblastí či jejich důkladnému prohlížení s pozdějším krátkým návratem. Míru laminarity můžeme předefinovat jako

$$LAM^t = \frac{100}{2R^t} \left( \sum_{(i,j) \in H_L} r_{ij}^t + \sum_{(i,j) \in V_L} r_{ij}^t \right), \quad (2.8)$$

kde  $R^t = \sum_{i=1}^{N-1} \sum_{j=i+1}^N r_{ij}^t$  a sumy představují součet prvků  $r_{ij}^t$  matice rekurence, které tvoří horizontální (vertikální) linie.

#### 2.4.4. CORM

CORM (*center of recurrent mass*) vyjadřuje, kde přibližně se vyskytuje většina opakujících se bodů (v čase).



Obrázek 2.12: Ilustrace rozložení bodů rekurence.

Jedná se o znormovanou veličinu, jejíž maximální hodnota je rovna 100. Malé hodnoty CORM značí, že opětovné fixace mají tendenci nastávat brzo po prvním prohlédnutí oblasti, zatímco velké hodnoty značí, že opětovné fixace nastávají později po prvním prohlédnutí. Na obrázku 2.12 je vidět, že hodnota CORM je malá, jestliže jsou rekurentní body blízko incidenční linie (a). Pokud k opakování dochází později v čase, rekurentní body jsou od incidenční linie dál (b) a hodnota CORM bude vysoká. Například pokud se subjekt dívá detailně na několik částí obrázku a poté se již k těmto oblastem v čase nevrátí, tak většina rekurentních bodů bude blízko incidenční linie, což bude znamenat malou hodnotu CORM.

CORM definujeme jako vzdálenost středu většiny opakujících se bodů od incidenční linie (hlavní diagonály).

$$CORM = 100 \frac{\sum_{i=1}^{N-1} \sum_{j=i}^N (j-i) r_{ij}}{(N-1)R}, \quad (2.9)$$

kde  $r_{ij}$  představuje prvek rekurentní matice na pozici  $i, j$  a  $R$  je součet opakujících se bodů v horním trojúhelníku v rekurentním grafu (viz míra rekurence).

V případě, že budeme uvažovat dobu trvání fixací, pak CORM označuje, u kterých rekurentních fixací trávíme nejvíce času. CORM můžeme následně předefinovat jako

$$CORM^t = 100 \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N (j-i) r_{ij}^t}{(N-1)^2 T}, \quad (2.10)$$

kde  $r_{ij}^t$  představuje prvek rekurentní matice na pozici  $i, j$ , a  $T$  je součet doby trvání všech fixací, tj.  $T = \sum_{i=1}^N t_i$ .

### 2.4.5. Shrnutí

Míry opakování a hodnota CORM zachycují celkovou strukturu posloupnosti fixací. Měří, kolikrát byl pohled opakovaně zaměřen na dané části obrázku (opakování) a jestli tyto opětovné fixace nastávají dříve nebo později v průběhu testu (CORM).

Determinismus a laminarita jsou míry, které zkoumají strukturu podrobněji. Vyjadřují posloupnosti fixací, které se opakují (determinismus) a body, ve kterých se vyskytuje podrobnější (detailní) prohlížení oblasti obrázku (laminarita).

V případě, kdy budeme uvažovat i dobu trvání času, dávají nám míry rekurentní kvantifikační analýzy ještě podrobnější informace o průběhu testu.

# Kapitola 3

## Praktická část

V praktické části této bakalářské práce bylo hlavním cílem ve statistickém prostředí R zprogramovat funkce pro rekurentní kvantifikační analýzu a na konkrétním příkladu ukázat jak RQA funguje.

Předmětem studie bylo odhadnout vzdálenost dvou bodů na předložené kartografické mapě. V laboratoři Katedry geoinformatiky na Univerzitě Palackého v Olomouci byl na skupině 45 respondentů proveden eye-tracking test pomocí eye-trackeru značky SMI, kdy byla respondentům na monitoru ukázána kartografická mapa. Následně byla naměřená data zpracována pomocí softwaru BeGaze a exportována formou textového editoru.

Data obsahovala záznamy o čísle testu, mapě, na kterou se subjekt díval, označení účastníka, jeho jméno, pohlaví, číslo fixace, začátek fixace v milisekundách, konec fixace v milisekundách, celkovou dobu trvání, x-ovou souřadnici fixace v pixelech, y-ovou souřadnici v pixelech a příslušnou oblast zájmu (obrázek 3.1).

Trial	Stimulus	Participant	Jmeno	Pohlavi	Index	Event Start Trial Time [ms]	Event End Trial Time [ms]	Event Duration [ms]	Fixation Position X [px]	Fixation Position Y [px]	AOI Name
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	1	1.9	321.9	320.0	924.7	631.3	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	2	385.9	521.9	136.0	922.6	1072.6	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	3	557.9	705.0	147.0	949.2	1043.1	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	4	729.8	941.8	212.0	1043.7	1040.0	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	5	973.8	1177.8	204.0	869.8	1039.9	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	6	1213.8	3789.4	2575.6	884.6	1040.7	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	7	3973.4	6573.1	2599.7	883.0	1046.4	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	8	6612.9	6896.9	284.0	1051.3	1039.0	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	9	6964.9	7252.8	287.9	1479.5	722.3	B
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	10	7332.9	7708.0	427.9	1235.9	622.2	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	11	7852.7	8524.6	671.9	920.1	1035.6	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	12	8696.6	8920.6	224.0	807.0	375.4	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	13	9084.5	9468.5	463.9	1521.3	761.8	B
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	14	9552.6	10588.3	1035.7	888.6	1034.6	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	15	10632.3	11048.3	416.0	882.5	1020.0	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	16	11212.3	11708.2	495.9	883.2	1028.3	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	17	11744.2	12040.2	296.0	1054.4	1032.9	Scale
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	18	12124.1	12408.1	276.0	1442.1	720.9	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	19	12440.1	13140.0	699.9	1351.4	654.3	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	20	12676.0	13084.0	408.0	1232.1	636.0	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	21	13116.0	13715.9	599.9	1121.0	615.7	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	22	13779.8	14523.8	743.9	920.6	523.0	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	23	14579.7	14923.7	344.0	814.3	414.8	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	24	14983.7	15127.7	143.9	750.7	335.8	Map
Trial034	09-MI-CX-SI-VE..jpg	P16	Trnová	Žena	25	15151.7	16127.5	975.8	700.6	348.4	Map

Obrázek 3.1: Ukázka naměřených dat.



Hlavním nástrojem pro zpracování dat bylo programovací prostředí RStudio [14]. Před analýzou dat jsme nahráli příslušný datový soubor do statistického softwaru RStudio. To jsme provedli pomocí příkazu:

```
data=read.delim("Map09+Map03-Event_Statistics-Single.txt",sep="t")
```

Pro vizualizaci dat jsme využili funkce z balíčku `ggplot2` [35], `ggrepel` [27], `gridExtra` [22] a `rapportools` [3]. Podle [12] jsme si pro vložení obrázku (mapy) pod graf museli nainstalovat balíček `png` [31]. V případě setřídění dat v maticích jsme využili funkce z balíčku `tidyr` [36].

V RStudiu jsme nenašli balíček, který by byl lehce využitelný pro naše data, proto jsme museli funkce nezbytné pro rekurentní kvantifikační analýzu vytvořit sami. V následující části jsou uvedeny pouze nejdůležitější funkce, které jsme v praktické části využili. Celý algoritmus funkcí pro všechny metody je možné nalézt v příloze (příloha 2). Kompletní algoritmus funkcí s poznámkami pouze pro metodu fixní vzdálenosti je v příloze 1. Základními kroky analýzy bylo, pomocí funkce `zakladni_grafy` vizualizovat jednotlivé fixace do spojnicového grafu, pod který jsme vložili obrázek mapy, na který se subjekt díval.

```
zakladni_grafy = function (data,x = dataX, y = dataY, delka_fixace,  
T, type = typ_okoli, h = rozmer_okoli_graf,r = polomer  
(rozmer_ctverce), s_dobou_fixace)
```

Pro zjištění vzdálenosti mezi dvěma fixacemi jsme využili funkci `vzdalenost`, která počítá vzdálenost mezi dvěma fixacemi se zvolenou eukleidovskou metrikou. Vstupní data funkce  $x1$  a  $x2$  představují x-ové souřadnice fixací  $f_i$  a  $f_j$ ,  $y1$  a  $y2$  představují jejich y-ové souřadnice.

```
vzdalenost = function (x1, x2, y1, y2)
```

Před vytvořením rekurentní matice, jsme vytvořili matici vzdáleností. Pomocí funkce `matice_vzdalenosti` jsme vytvořili matici  $N \times N$ , jejíž prvky jsou vzdálenosti mezi dvěma fixacemi.

```
matice_vzdalenosti = function (x = dataX, y = dataY)
```

K výrobě matice rekurence, která obsahuje na pozici  $r_{ij}$  hodnotu 1, pokud je vzdálenost mezi fixacemi  $f_i$  a  $f_j$  menší než zvolený poloměr a hodnotu 0 na pozici  $r_{ij}$ , pokud je vzdálenost větší, jsme použili funkci `matice_rekurence`. vstupní data této funkce jsou x-ové a y-ové souřadnice fixací, velikost poloměru a také informace, zda uvažujeme délku trvání fixací. V případě uvažování doby trvání fixací by na pozici  $r_{ij}$  byla hodnota, která odpovídá součtu doby trvání fixací  $f_i$  a  $f_j$ , pokud je vzdálenost mezi těmito dvěma fixacemi menší než zvolený poloměr.

```
matice_rekurence = function (x = dataX, y = dataY, r = polomer,
s_dobou_fixace)
```

Z matice rekurence jsme nebyli schopni vytvořit graf rekurence, proto jsme matici setřídili. Funkce `recurrence_plot` umožní ze seřazené matice rekurence vytvořit graf rekurence. Kromě seřazené matice rekurence potřebujeme jako vstupní hodnoty do funkce zadat ještě celkovou dobu fixací, typ - metodu, kterou jsme si vybrali a informaci, zda uvažujeme dobu trvání - T (true), F (false).

```
recurrence_plot = function (x = serazena_m_rekurence, T, type,
s_dobou_fixace)
```

Pro vypočítání všech měr rekurence, jsme využili vytvořenou funkci `charakteristika`, jejíž vstupní hodnoty jsou matice rekurence, počet bodů, které tvoří diagonální linie, počet bodů, které tvoří horizontální linie, počet bodů, které tvoří vertikální linie a informaci, jestli uvažujeme dobu trvání.

```
charakteristika = function(mat_rekurence, D_L, H_L, V_L,
s_dobou_fixace)
```

Pro rekurentní kvantifikační analýzu jsme si vybrali respondenta 16, který se díval na mapu Kroměříže a okolí.

```
mapa9 = data[data$Stimulus == "09-M1-CX-SI-VE.jpg", ]
P16 = mapa9[mapa9$Participant == "P16", ]
```

Vzhledem k umístění počátku souřadnicového systému (0,0) eye-trackeru v levém horním rohu, je třeba transformovat vstupní y-ové hodnoty z dat na záporné. Pokud bychom data netransformovali, fixace by byly zobrazeny v osově souměrnosti podle osy x.

Zkrácený datový soubor vybraný pro jednoho respondenta stále obsahoval mnoho informací, které pro nás nebyly podstatné. Z tohoto důvodu jsme vytvořili tabulku pouze s daty pro naše účely a z této tabulky budeme vycházet.

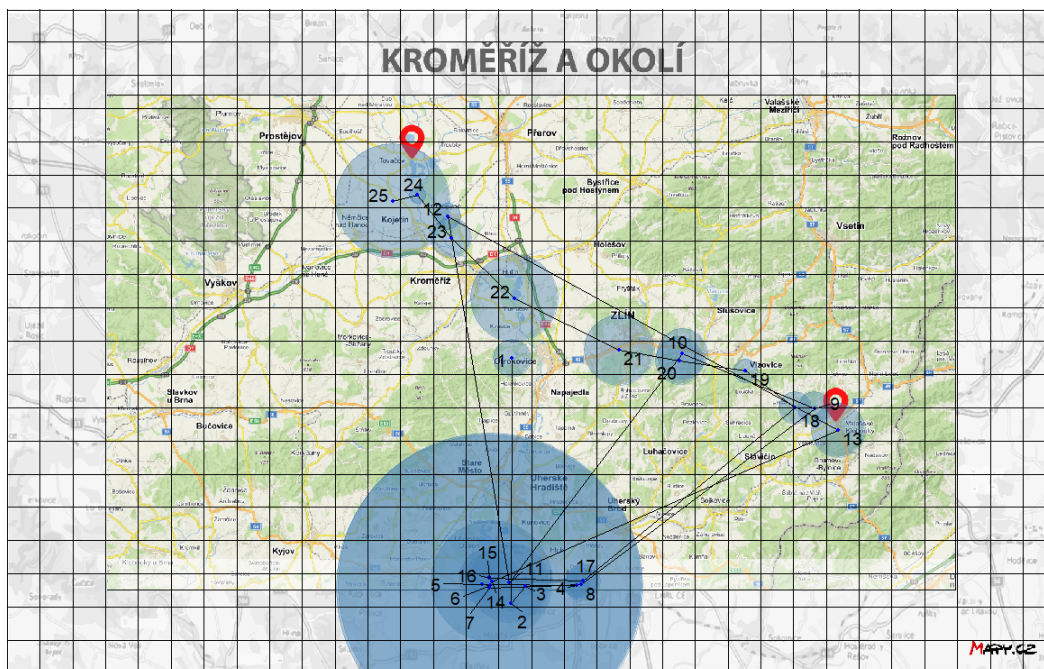
Do nové tabulky (obrázek 3.2) jsme vložili informace o souřadnicích X, Y (původní), Y (transformované) jednotlivých fixací, dobu trvání fixací, oblast zájmu, ve které fixace leží, oblast zájmu převedenou na číselné hodnoty a sloupce, které označují řádek a sloupec čtverce, ve kterém fixace leží. Poslední dva sloupce, které označují pozici čtverce, jsou předpřipravené k další analýze a proto jsou zatím nulové.

```
data = data.frame(dataX, dataY, data YN, delka_fixace,
mista_zajmu, mista_zajmu_numeric = mista_zajmu,
radek_ctverce = 0, sloupec_ctverce = 0)
```

	dataX	dataYN	dataY	delka_fixace	mista_zajmu	mista_zajmu_numeric	radek_ctverce	sloupec_ctverce
1	924.7	631.3	-631.3	320.0	Map	4	0	0
2	922.6	1072.6	-1072.6	136.0	Scale	5	0	0
3	949.2	1043.1	-1043.1	147.9	Scale	5	0	0
4	1043.7	1040.8	-1040.8	212.0	Scale	5	0	0
5	869.8	1039.9	-1039.9	204.0	Scale	5	0	0
6	884.6	1040.7	-1040.7	2575.6	Scale	5	0	0

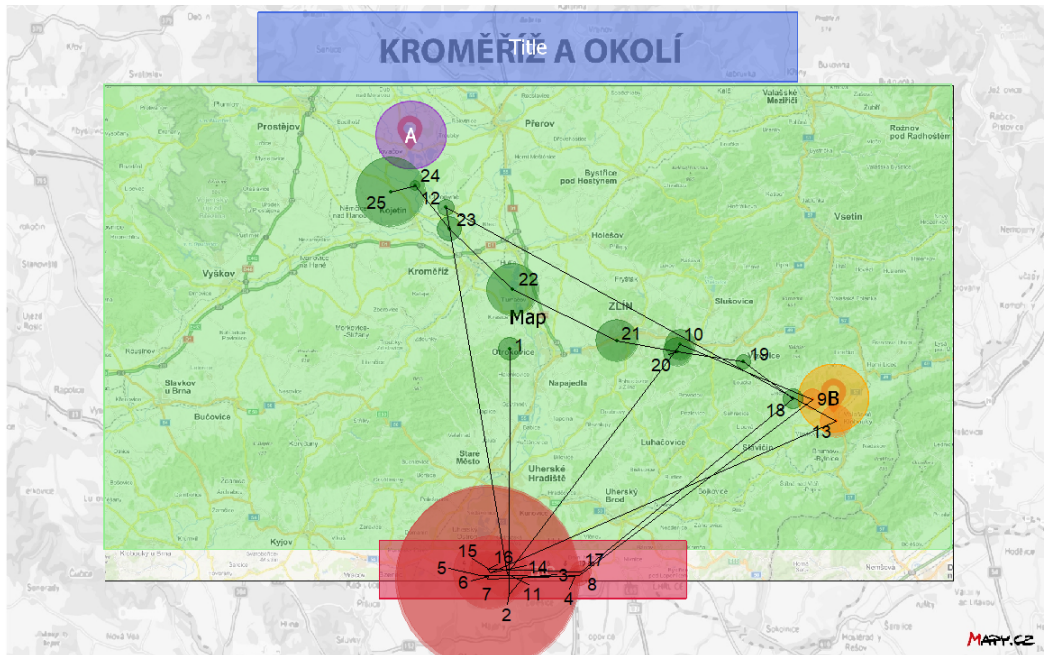
Obrázek 3.2: Data, se kterými budeme pracovat.

Nejprve si ukážeme vizualizace jednotlivých grafů pro každou z metod včetně doby trvání jednotlivých fixací. Na obrázku 3.3 můžeme vidět mapu rozdělenou pomocí fixní mřížky s rozměry čtverců 60x60 pixelů. U fixací 24 a 25 vidíme jednu z nevýhod této metody, kdy dané fixace nepovažujeme za rekurentní, i když leží těsně u sebe.



Obrázek 3.3: Rozdělení mapy pomocí metody fixní mřížky s dobou trvání.

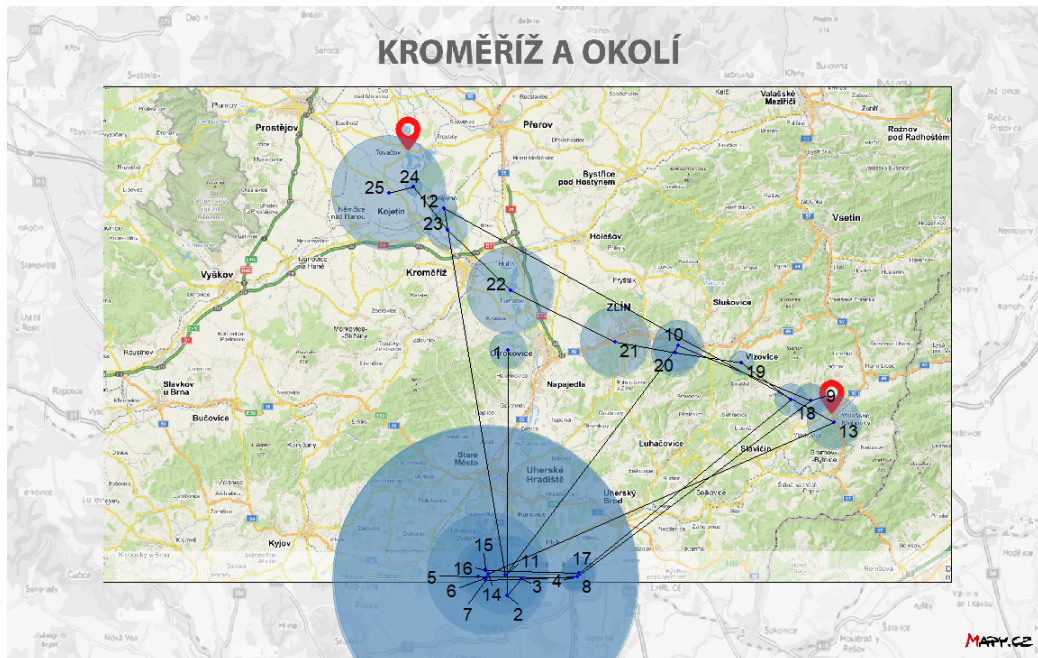
V případě metody AOI (obrázek 3.4) jsme mapu rozdělili na šest oblastí zájmu. Oblasti zájmu jsme v našem případě definovali tímto způsobem - bílé místo (v datech s číselným vyjádřením AOI označeno jako 1), bod A (2), bod B (3), mapa (4), měřítko (5) a titulek (6). Důvod rozdělení mapy na tyto oblasti souvisí se zadaným počátečním úkolem. Můžeme vidět, že oblasti měřítka věnoval respondent nejvíce času.



Obrázek 3.4: Rozdělení mapy pomocí metody AOI s dobou trvání.

U metody fixní vzdálenosti (obrázek 3.5) můžeme vidět spojnicový graf představující jednotlivé fixace. Poloměr pro metodu fixní vzdálenosti byl zvolen 60 pixelů. Poloměr a rozměry mřížky jsou volitelné parametry. Počet bodů, které jsou rekurentní, je na této volbě závislý. Pokud zvolíme poloměr, který bude příliš malý, nebudou se v grafu vyskytovat téměř žádné rekurentní body a o datech se toho mnoho nedozvíme. Na druhou stranu, pokud zvolíme poloměr příliš velký, budou všechny fixace rekurentní a nastane stejný problém jako v případě malého poloměru. V případě pohybu očí by měl být zvolený poloměr hodnotu 1-2 ° zorného úhlu [1]. Při velikosti obrázku mapy na monitoru 1920x1200 pixelů, od kterého byl respondent vzdálený 60 cm, představuje 60 pixelů 1,52 ° zorného úhlu, takže se jedná o ideální hodnotu [28].





Obrázek 3.5: Vizualizace fixací v případě fixní vzdálenosti s dobou trvání.

Pro daného respondenta ukážeme, jak vypadaly matice rekurence, které jsou nezbytné pro tvorbu grafu rekurence. Na obrázku 3.6 můžeme vidět jednotlivé matice rekurence pro prvních patnáct fixací pro každou z metod. Jelikož neuvažujeme dobu trvání, tak na pozici  $r_{ij}$  bude 1, pokud jsou fixace  $f_i$  a  $f_j$  rekurentní, jinak 0.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1
6	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1
7	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1
8	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1
15	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1

(a) Matice rekurence - metoda fixní mřížky.

↕	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
2	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
3	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
4	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
5	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
6	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
7	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
8	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
9	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
10	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
11	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
12	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
14	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1
15	0	1	1	1	1	1	1	1	0	0	1	0	0	1	1

(b) Matice rekurence - metoda oblastí zájmu

↕	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0
3	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	1	1	1	0	0	0	1	0	0	1	1
6	0	1	0	0	1	1	1	0	0	0	1	0	0	1	1
7	0	1	0	0	1	1	1	0	0	0	1	0	0	1	1
8	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	1	1	0	1	1	1	0	0	0	1	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
14	0	1	0	0	1	1	1	0	0	0	1	0	0	1	1
15	0	0	0	0	1	1	1	0	0	0	1	0	0	1	1

(c) Matice rekurence - metoda fixní vzdálenosti.

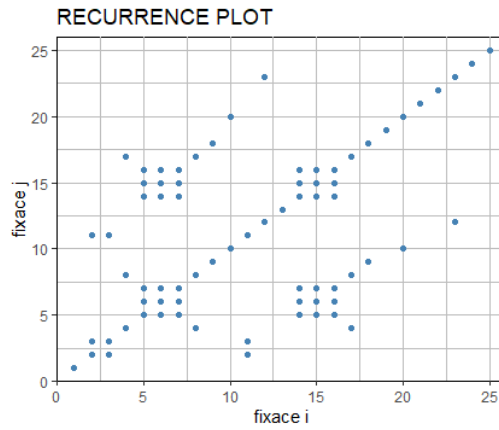
Obrázek 3.6: Matice rekurence pro jednotlivé metody, bez uvažování doby trvání.

U grafu rekurence pro metodu fixní mřížky - obrázek 3.7(a), můžeme vidět, že se malý počet bodů vyskytuje v grafu samostatně, většina rekurentních bodů tvoří shluky, což značí detailnější prohlížení oblasti. Pokud srovnáme graf s mapou 3.3, pak zjistíme, že se jedná o oblast měřítka, do které se respondent vracel pohledem. Podle rekurentního grafu můžeme říct, že si respondent tuto oblast nejdříve detailně prohlížel a následně se k ní ještě na další podrobné prohlížení vrátil. Nejdelsí diagonální linie je délky 5 – fixace 14, 15, 16, 17 a 18 opakovaly stejný vzor pohledů jako fixace 5, 6, 7, 8 a 9 (viz obrázek 3.3). Hodnotu CORM můžeme očekávat poměrně malou a podobnou jako v případě metody oblastí zájmu, neboť několik opětovných fixací nastalo později v čase.

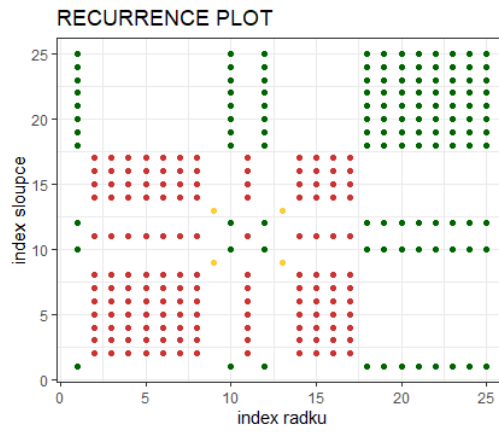
V případě metody AOI obsahuje graf rekurence (obrázek 3.7(b)) velké množství rekurentních bodů, což je z důvodu definovaných oblastí zájmů. V kombinaci s obrázkem 3.4 zjistíme, že respondent nejprve velmi detailně sledoval měřítka, po pár fixacích mimo tuto oblast, se do oblasti měřítka vrátil na podrobnější sledování a zbytek testu se věnoval pouze mapě. Je zajímavé, že při zjišťování vzdálenosti mezi body A a B se respondent do těchto oblastí téměř nedíval. Do bodu B se podíval při fixacích 9 a 13, do bodu A se nepodíval vůbec. Můžeme očekávat, že hodnota CORM bude relativně malá, ale díky několika fixacím, které nastaly později v čase, by měla být větší než v případě metody fixní mřížky.

U rekurentního grafu pro metodu fixní vzdálenosti (obrázek 3.7(c)) můžeme vidět výrazně méně rekurentních bodů, než v případě metody oblastí zájmu, ale na druhou stranu pevně zvolený poloměr nám odhalí několik rekurentních bodů, které jsme v případě metody fixní mřížky neviděli. Nejdelsí diagonála, která se v rekurentním grafu vyskytuje, je délky 5 a jedná se o stejnou diagonálu jako v případě metody fixní mřížky.

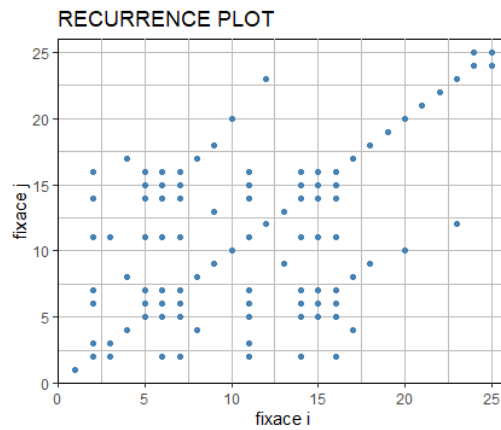
Podle metody fixní mřížky je počet rekurentních bodů nejmenší. Metoda fixní vzdálenosti odhalí fixace, které jsou blízko sebe, jenž neodhalí metoda fixní mřížky. V případě metody oblastí zájmu nám rekurentní graf naznačí, do kterých oblastí se respondent vracel.



(a) Metoda fixní mřížky.



(b) Metoda oblastí zájmu. Měřítko (červená), mapa (zelená) a bod B (žlutá).



(c) Metoda fixní vzdálenosti.

Obrázek 3.7: Grafy rekurence pro jednotlivé metody, bez uvažování doby trvání.



Míry rekurence v případě metody fixní mřížky (tabulka 3.1) říkají, že míra rekurence byla 8 %. To znamená, že ze všech možných dvojic fixací, které mohou být rekurentní, se jich v grafu vyskytuje právě 8 %. V 54 % případů přitom tyto rekurentní fixace opakovaly určitý vzor pohybů očí - toho si můžeme všimnout například u fixací 14, 15, 16, 17 a 18, kdy opakovaly stejný vzor jako před tím fixace 5, 6, 7, 8 a 9. Jak jsme naznačili při popisu grafu rekurence, kdy jsme říkali, že většina rekurentních bodů tvoří shluky, tak jsme tuto informaci potvrdili mírou laminarity, která byla 58 %. Totéž platí v případě hodnoty CORM, která je mírně přes 28, což značí, že opětovné fixace byly provedeny relativně brzo po prvním prohlédnutí až na pár výjimky.

V případě metody oblastí zájmu jsme očekávali, že bude míra rekurence největší, což se i potvrdilo, když jsme zjistili, že míra rekurence byla 40 %. To značí, že se v grafu vyskytuje 40 % ze všech možných dvojic fixací, které mohou být rekurentní. Vysoká byla i míra determinismu, která byla 67 %, to značí, že přes dvě třetiny rekurentních bodů opakovaly určitý vzor – nyní definujeme vzor jako pohyb ve stejné oblasti zájmu. Míra laminarity byla téměř 80 %, což je jasně patrné i z grafu rekurence, kdy je velmi málo bodů, které by netvořili horizontální nebo vertikální linie. Hodnota CORM byla díky počtu opětovných pozdějších fixací 29.

Pokud budeme uvažovat metodu fixní vzdálenosti, zjistíme, že míra rekurence byla 12 %, takže se v rekurentním grafu vyskytovalo 12 % ze všech možných dvojic fixací, které mohou být rekurentní. Míra determinismu byla 36 %, to značí, že ve více než třetině případů při opětovných fixacích respondent opakoval určité vzory pohledů - stejně jako v případě metody fixní mřížky, jsme odhalili vzor fixací 14, 15, 16, 17 a 18. V 50 % případů respondent věnoval rekurentním bodům detailnější prohlížení oblasti (ať už se jednalo o detailní prohlížení s krátkým návratem, nebo letmé prohlédnutí s pozdějším detailním zkoumáním), což označuje míra laminarity, která byla 50 %. Hodnota CORM byla téměř 27, nejmenší ze všech metod.

Metoda fixní mřížky odhalila nejmenší procento rekurentních bodů ze všech metod, což odpovídá i rekurentním grafům. Míra laminarity je u metod fixní mřížky a metody fixní vzdálenosti téměř shodná. Je zajímavé, že mezi mírou determinismu a mírou laminarity je u metody fixní mřížky nejmenší rozdíl (8 %). Hodnota CORM byla u všech metod přibližně stejná.

Metoda	Počet opakování	REC [%]	DET [%]	LAM [%]	CORM
fixní mřížka	24	8	54.17	58.33	28.3
oblasti zájmu	122	40.67	67.21	79.92	29.44
fixní vzdálenost	36	12	36.11	50	26.62

Tabulka 3.1: Míry rekurence bez doby trvání.

Pokud budeme uvažovat dobu trvání, pak musíme předefinovat matice rekurence následujícím způsobem (obrázek 3.8). V případě, že jsou fixace  $f_i$  a  $f_j$  rekurentní, pak má prvek  $r_{ij}$  hodnotu, která značí součet dob trvání fixací  $f_i$  a  $f_j$ , pokud fixace rekurentní nejsou, pak má hodnotu 0.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	640	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	272	284	0	0	0	0	0	0	0	808	0	0	0	0
3	0	284	296	0	0	0	0	0	0	0	820	0	0	0	0
4	0	0	0	424	0	0	0	496	0	0	0	0	0	0	0
5	0	0	0	0	408	2780	2804	0	0	0	0	0	0	1240	620
6	0	0	0	0	2780	5151	5175	0	0	0	0	0	0	3611	2992
7	0	0	0	0	2804	5175	5199	0	0	0	0	0	0	3635	3016
8	0	0	0	496	0	0	0	568	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	576	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	856	0	0	0	0	0
11	0	808	820	0	0	0	0	0	0	0	1344	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	448	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	928	0	0
14	0	0	0	0	1240	3611	3635	0	0	0	0	0	0	2071	1452
15	0	0	0	0	620	2992	3016	0	0	0	0	0	0	1452	832

(a) Matice rekurence - metoda fixní mřížky.

↕	1 ↕	2 ↕	3 ↕	4 ↕	5 ↕	6 ↕	7 ↕	8 ↕	9 ↕	10 ↕	11 ↕	12 ↕	13 ↕	14 ↕	15 ↕
1	640	0	0	0	0	0	0	0	0	748	0	544	0	0	0
2	0	272	284	348	340	2712	2736	420	0	0	808	0	0	1172	552
3	0	284	296	360	352	2724	2748	432	0	0	820	0	0	1184	564
4	0	348	360	424	416	2788	2812	496	0	0	884	0	0	1248	628
5	0	340	352	416	408	2780	2804	488	0	0	876	0	0	1240	620
6	0	2712	2724	2788	2780	5151	5175	2860	0	0	3248	0	0	3611	2992
7	0	2736	2748	2812	2804	5175	5199	2884	0	0	3272	0	0	3635	3016
8	0	420	432	496	488	2860	2884	568	0	0	956	0	0	1320	700
9	0	0	0	0	0	0	0	0	576	0	0	0	752	0	0
10	748	0	0	0	0	0	0	0	0	856	0	652	0	0	0
11	0	808	820	884	876	3248	3272	956	0	0	1344	0	0	1708	1088
12	544	0	0	0	0	0	0	0	0	652	0	448	0	0	0
13	0	0	0	0	0	0	0	0	752	0	0	0	928	0	0
14	0	1172	1184	1248	1240	3611	3635	1320	0	0	1708	0	0	2071	1452
15	0	552	564	628	620	2992	3016	700	0	0	1088	0	0	1452	832

(b) Matice rekurence - metoda oblastí zájmu.

↕	1 ↕	2 ↕	3 ↕	4 ↕	5 ↕	6 ↕	7 ↕	8 ↕	9 ↕	10 ↕	11 ↕	12 ↕	13 ↕	14 ↕	15 ↕
1	640	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	272	284	0	0	2712	2736	0	0	0	808	0	0	1172	0
3	0	284	296	0	0	0	0	0	0	0	820	0	0	0	0
4	0	0	0	424	0	0	0	496	0	0	0	0	0	0	0
5	0	0	0	0	408	2780	2804	0	0	0	876	0	0	1240	620
6	0	2712	0	0	2780	5151	5175	0	0	0	3248	0	0	3611	2992
7	0	2736	0	0	2804	5175	5199	0	0	0	3272	0	0	3635	3016
8	0	0	0	496	0	0	0	568	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	576	0	0	0	752	0	0
10	0	0	0	0	0	0	0	0	0	856	0	0	0	0	0
11	0	808	820	0	876	3248	3272	0	0	0	1344	0	0	1708	1088
12	0	0	0	0	0	0	0	0	0	0	0	448	0	0	0
13	0	0	0	0	0	0	0	0	752	0	0	0	928	0	0
14	0	1172	0	0	1240	3611	3635	0	0	0	1708	0	0	2071	1452
15	0	0	0	0	620	2992	3016	0	0	0	1088	0	0	1452	832

(c) Matice rekurence - metoda fixní vzdálenosti.

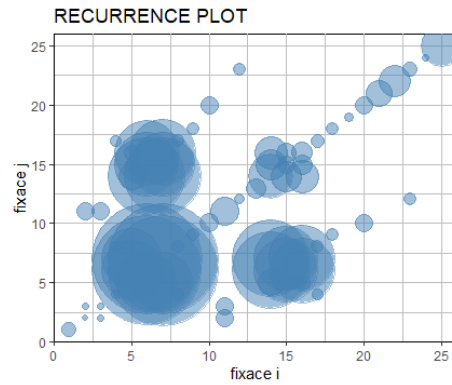
Obrázek 3.8: Matice rekurence pro jednotlivé metody s dobou trvání.

U rekurentního grafu pro metodu fixní mřížky (obrázek 3.9(a)) můžeme vidět, že při prvotním prohlédnutí oblasti se respondent věnoval oblasti delší čas než při následném návratu. Při porovnání rekurentního grafu s mapou rozdělenou pomocí fixní mřížky (obrázek 3.3) zjistíme, že oblast, které se respondent věnoval je měřítko. Dále si můžeme všimnout, že subjekt věnoval rekurentním fixacím, které netvořily shluky pouze krátký čas.

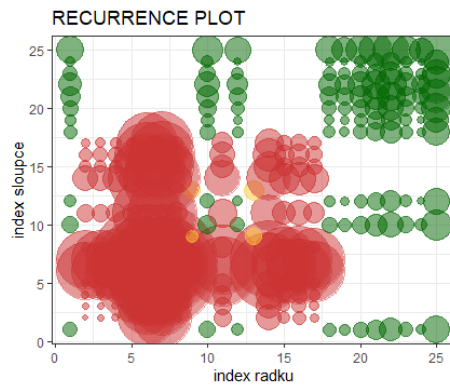
U grafu rekurence pro metodu oblastí zájmu můžeme vidět (obrázek 3.9(b)), že většina času rekurentních fixací byla věnována prvotnímu prohlížení oblasti měřítko, následná délka prohlížení oblasti mapy byla výrazně kratší.

Graf rekurence pro metodu fixní vzdálenosti (obrázek 3.9(c)) je podobný metodě oblastí zájmu, kromě míst, kdy se díval respondent do oblasti mapy. Můžeme si opět všimnout, že rekurentním fixacím, které se vyskytují samostatně, věnoval subjekt méně času než fixacím, které tvoří shluky.

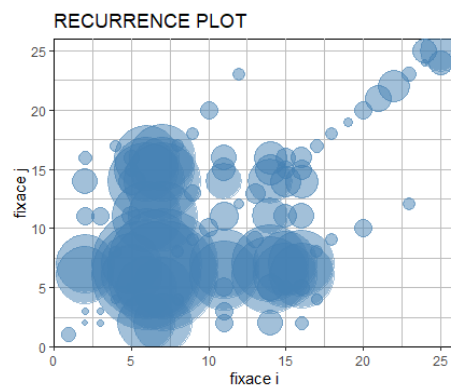
U rekurentních grafů s dobou trvání můžeme u všech metod vidět téměř shodné výsledky u rekurentních fixací, kde strávil respondent nejvíce času. Grafy rekurence s dobou trvání nám dávají detailnější informace o chování respondenta.



(a) Metoda fixní mřížky.



(b) Metoda oblastí zájmu. Měřítko (červená), mapa (zelená) a bod B (žlutá).



(c) Metoda fixní vzdálenosti.

Obrázek 3.9: Grafy rekurence pro jednotlivé metody s dobou trvání.

Pokud se zaměříme na míry rekurence s dobou trvání (tabulka 3.2), pak si můžeme všimnout, že v případě metody fixní mřížky respondent věnoval 12 % času rekurentním fixacím. 69 % času rekurentních fixací bylo v průběhu, kdy rekurentní fixace opakovaly určitý vzor a 76 % celkového času respondent věnoval rekurentním fixacím, při kterým detailně prohlížel vybranou oblast. Hodnota  $CORM^t$  byla 3, to značí, že nejvíce času respondent strávil u rekurentních fixací, které jsou blízko incidenční linie.

U metody oblastí zájmu jsme zjistili, že u rekurentních fixací strávil respondent 42 % celkového času. Rekurentním fixacím, které opakovaly určitý vzor, respondent věnoval 73 % času. Pouze 16 % času nebylo věnováno opětovnému detailnímu prohlížení. Z grafu rekurence je patrné, že hodnota  $CORM^t$  bude značně vyšší než v případě metody fixní mřížky. To se potvrdilo, když hodnota byla 11.

V případě metody fixní vzdálenosti téměř 18 % času strávil respondent u rekurentních fixací. Přes 45 % celkového času věnoval rekurentním fixacím, které měly tendenci opakovat předchozí vzor. Míra laminarity byla skoro 67 %, to značí, že více než dvě třetinu času si respondent detailně prohlížel vybrané oblasti. Hodnota  $CORM^t$  byla nepatrně větší než v případě metody fixní mřížky a to 4.

<b>Metoda</b>	<b>Doba opakování [ms]</b>	<b>REC [%]</b>	<b>DET [%]</b>	<b>LAM [%]</b>	<b>CORM</b>
fixní mřížka	42097.9	12.11	69.14	76.7	3.06
oblasti zájmu	147206.5	42.33	73.46	84.02	11.15
fixní vzdálenost	62578.6	18	46.51	51.6	4.28

Tabulka 3.2: Míry rekurence s dobou trvání

### 3.1. Shrnutí

Metody, kdy uvažujeme či neuvažujeme dobu trvání nelze srovnávat, neboť každá varianta nám přináší jiné informace. V případě uvažování doby trvání fixací se jedná o rozšíření původních informací, kdy jsme dobu trvání neuvažovali, proto je ideální využít obě varianty.

Metoda oblastí zájmu je ideální v případě, kdy chceme odhalit rekurentní fixace v jednotlivých oblastech zájmu. Míra determinismu má sníženou vypovídající hodnotu při zjišťování opakujícího vzoru.

Jestliže nás bude více zajímat, zda se u chování respondenta objevovaly určité vzory, které se opakovaly, je vhodné využít metodu fixní vzdálenosti, jelikož touto metodou získáme více informací než v případě metody fixní mřížky, která nemusí vždy odhalit všechny rekurentní body.

V případě rozšíření o dobu trvání získáme navíc informace o tom, jaký čas respondent strávil při opakování vzorů pohledů či kde měly tendenci nastávat rekurentní fixace, u kterých respondent strávil nejvíce času.

# Závěr

Ve své práci jsem ukázala, jakým způsobem se dá rekurentní kvantifikační analýza využít při charakteristice eye-tracking dat. Zejména jaké jsou rozdíly při použití různých metod stanovení rekurence. Pomocí vytvořených funkcí v programovacím prostředí R umožnit i ostatním, kteří by měli zájem, jednoduše analyzovat data pomocí tohoto softwaru.

Hodnocení jednotlivých metod závisí na tom, jaký si zvolíme počáteční cíl. Metodu fixní vzdálenosti je nejvhodnější využít, pokud chceme odhalit opakující se vzory pohledů, současně řeší nedostatky fixní mřížky, kdy někdy nepovažujeme dvě fixace za rekurentní, i když jsou blízko sebe. Metoda oblastí zájmu přehledně ukazuje, do kterých oblastí se respondent vracel.

Uvažování doby trvání navíc přináší rozšíření informací o datovém souboru a ještě detailnější popis chování respondenta.

Tvorbou celé této práce mě provázal statistický software R, se kterým jsem na začátku neměla mnoho zkušeností a tak začátky byly náročné, ale v závěru jsem velmi ráda za nabyté zkušenosti a znalosti, které mi tato práce dala.



# Literatura

- [1] Anderson N. C., Bischof W F., Laidlaw K. E. W., Risko E. F., Kingstone A.: *Recurrence quantification analysis of eye movements*. Behavior Research Methods, 2013.
- [2] Auguie B., Antonov A.: *gridExtra: Miscellaneous Functions for "Grid" Graphics* [online], 2017 [cit. 2020-04-11]. Dostupné z: <https://cran.r-project.org/web/packages/gridExtra/gridExtra.pdf>
- [3] Blagotić A., Daróczy G.: *Rapport: a report templating system* [online], 2016 [cit. 2020-03-05]. Dostupné z: <http://cran.r-project.org/package=rapport>
- [4] Bojko, A.: *Eye Tracking the User Experience: A Practical Guide to Research*. Rosenfeld Media, New York, 2013.
- [5] Davids K., Hristovski R., Araújo D., Serre N. B., Button Ch., Passos P.: *Complex Systems in Sport*[online] Routledge Research in Sport and Exercise Science, 2014 [cit. 2020-05-01]. Dostupné z: [https://books.google.cz/books?id=CFMlAgAAQBAJ&printsec=frontcover&hl=cs&source=gbs\\_ge\\_summary\\_r&cad=0v=onepage&q&f=false](https://books.google.cz/books?id=CFMlAgAAQBAJ&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0v=onepage&q&f=false)
- [6] Discombe R. M., Cotterill S. T.: *Eye tracking in sport: A guide for new and aspiring researchers* [online] Sport Exercise Psychology Review, Vol. 11 (2), The British Psychological Society, 2015 [cit. 2020-04-20]. Dostupné z: [https://www.researchgate.net/publication/312297582\\_Eye\\_tracking\\_in\\_sport\\_A\\_guide\\_for\\_new\\_and\\_aspiring\\_researchers](https://www.researchgate.net/publication/312297582_Eye_tracking_in_sport_A_guide_for_new_and_aspiring_researchers).
- [7] DLRC *Tobii TX300 Eye Tracker Informations for Users* [online] DLRC [cit. 2020-02-26]. Dostupné z: [https://stemedhub.org/resources/2122/download/Eye\\_Tracker\\_Info\\_Session.pdf](https://stemedhub.org/resources/2122/download/Eye_Tracker_Info_Session.pdf)
- [8] Duchowski A.: *Eye Tracking Methodology: Theory and Practise* Springer-Verlag London Limited, Londýn, 2007.
- [9] Duchowski A. T.: *A Breadth-first Survey of Eye Tracking Applications*. [online] Behavior Research Methods, Instruments & Computers, vol. 34(4), s. 455–470, 2002 [cit. 2020-13-03]. Dostupné z: <http://andrewd.ces.clemson.edu/research/vislab/docs/BET107cr.pdf>

- [10] Guastello S. J., Gregson R. A. M.: *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data* [online] CRC Press Taylor & Francis Group, Boca Raton, 2011 [cit. 2020-04-09]. Dostupné z: [https://books.google.cz/books?id=BKHMBQAAQBAJ&printsec=frontcover&hl=cs&source=gbs\\_ge\\_summary\\_r&cad=0v=onepage&q&f=false](https://books.google.cz/books?id=BKHMBQAAQBAJ&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0v=onepage&q&f=false)
- [11] Holmqvist K., Nyström M., Andersson R., Dewhurst R., Jarodzka H., van de Weijer J.: *Eye Tracking: A Comprehensive Guide to Methods and Measures* Oxford University Press, New York, 2011.
- [12] Hillebrand J.: *Add a background png image to ggplot2* [online] ThinkToStart, 2014 [cit. 2020-01-06]. Dostupné z: <http://thinktostart.com/add-a-background-png-to-ggplot2/>
- [13] Jindra T.: *Ir.M.A. – Klinický přístroj pro měření dynamiky akomodace avergence* [online] Praha, 2011 [cit. 2020-02-01]. Diplomová práce, Univerzita Karlova v Praze, 1. lékařská fakulta. Dostupné z: <https://is.cuni.cz/webapps/zzp/detail/109248/>
- [14] Jones O., Maillardet R., Robinson A.: *Introducing to Scientific Programming and Simulation Using R*. Chapman & Hall/CRC, 2014.
- [15] Klein Ch., Ettinger U.: *Eye Movement Research: An Introduction to its Scientific Foundations and Applications* [online] Springer Nature Switzerland AG, 2019 [cit. 2020-03-01]. Dostupné z: <https://books.google.cz/books?id=biK3DwAAQBAJ&pg=PA633&dq=Christoph+Klein&hl=cs&sa=X&ved=0ahUKEwiD0b7I5brpAhWPnxQKHfJZAXgQ6AEIdDAIv=onepage&q&f=false>
- [16] Kodera J., Van Quang T.: *Vizuální nelineární rekurentní analýza a její aplikace na český akciový trh* [online] Politická ekonomie, vol. 3, s. 305-322, 2009 [cit. 2020-02-25]. Dostupné z: <https://polek.vse.cz/pdfs/pol/2009/03/02.pdf>
- [17] Kučera M.: *Využití open-source nástrojů pro přípravu, průběh a vyhodnocení eye-tracking experimentů* [online] Olomouc, 2014 [cit. 2020-02-10]. Bakalářská práce, Univerzita Palackého v Olomouci, Přírodovědecká fakulta. Dostupné z: [https://theses.cz/id/r37645/BP\\_Michal\\_Kuera\\_2014.pdf?lang=en](https://theses.cz/id/r37645/BP_Michal_Kuera_2014.pdf?lang=en).
- [18] Lohrenz M.: *New Patent Shows Potential for Eye-Tracking Technology in Transportation Safety* [online] Volpe National Transportation Systems Center, Cambridge, 2017 [cit. 2020-02-07]. Dostupné z: <https://www.volpe.dot.gov/news/patent-shows-potential-for-eye-tracking-technology-in-transportation-safety>

- [19] Lohse G.: *Consumer eye movement patterns on Yellow Pages advertising* [online] Journal of Advertising, vol. 26(1), s. 61-73, 1997 [cit. 2020-02-04]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.39.3188rep=rep1type=pdf>
- [20] Marwan N.: *Historical Review of Recurrence Plots* [online] The European Physical Journal Special Topics, 164(1), s. 3–12, 2008 [cit. 2020-03-03]. Dostupné z: [https://www.researchgate.net/publication/225110160\\_A\\_historical\\_review\\_of\\_recurrence\\_plots](https://www.researchgate.net/publication/225110160_A_historical_review_of_recurrence_plots)
- [21] Marwan N., Romano M. C., Thiel M., Kurths J.: *Recurrence plots for the analysis of complex systems* Physics Reports 438, s. 237-329, 2007.
- [22] Murrell P.: *The Grid Graphics Package* [online], 2005. Dostupné z: <https://stat.ethz.ch/R-manual/R-patched/library/grid/html/grid-package.html>
- [23] Pavelka T.: *Možnosti eye-tracking v kartografii* [online] Brno, 2015 [cit. 2020-03-15]. Bakalářská práce, Masarykova univerzita v Brně, Přírodovědecká fakulta. Dostupné z: <https://is.muni.cz/th/ele41/Bakalarka.pdf>
- [24] Popelka, S.: *Eye-tracking (nejen) v kognitivní kartografii: praktický průvodce tvorbou a vyhodnocením experimentu*. Univerzita Palackého, Olomouc, 2013.
- [25] Popelka S., Brychtová A., Voženílek V.: *Eye-tracking a jeho využití při hodnocení map* [online] Geografický časopis, vol. 64(1), s. 71-87, 2012 [cit. 2020-04-10]. Dostupné z: [https://www.researchgate.net/publication/265763136\\_Eye-tracking\\_a\\_jeho\\_vyuziti\\_pri\\_hodnoceni\\_map](https://www.researchgate.net/publication/265763136_Eye-tracking_a_jeho_vyuziti_pri_hodnoceni_map)
- [26] Riley M. A., Van Orden G. C.: *Tutorials in contemporary nonlinear methods for the behavioral sciences* Retrieved March 1, 2005, dostupné z: <https://www.nsf.gov/sbe/bcs/pac/nmbs.jsp>.
- [27] Slowikowski K. a kol.: *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'* [online], 2020 [cit. 2020-04-10]. Dostupné z: <https://cran.r-project.org/web/packages/ggrepel/ggrepel.pdf>
- [28] SR Research EyeLink *Visual Angle Calculator* [online] SR Research Ltd, Ottawa [cit. 2020-03-10]. Dostupné z: <https://www.sr-research.com/visual-angle-calculator/>
- [29] Tobii Pro AB: *How do Tobii Eye Trackers work?* [online] Dostupné z: <https://www.tobii.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/>

- [30] Tobii Pro AB: *Packaging design* [online]. Dostupné z: <https://www.tobiipro.com/fields-of-use/marketing-consumer-research/packaging-design/>
- [31] Urbanek S.: *png: Read and write PNG images* [online], 2015 [cit. 2020-03-09]. Dostupné z: <https://cran.r-project.org/web/packages/png/png.pdf>
- [32] Vaidyanathan P., Pelz J., Alm C., Shi P., Haake A.: *Recurrence quantification analysis reveals eye-movement behavior differences between experts and novices* [online] Conference: Proceedings of the Symposium on Eye Tracking Research and Applications, 2014 [cit. 2020-04-05]. Dostupné z: [https://www.researchgate.net/publication/262154876\\_Recurrence\\_quantification\\_analysis\\_reveals\\_eye-movement\\_behavior\\_differences\\_between\\_experts\\_and\\_novices](https://www.researchgate.net/publication/262154876_Recurrence_quantification_analysis_reveals_eye-movement_behavior_differences_between_experts_and_novices)
- [33] Webber Ch. L., Ioana C., Marwan N.: *Recurrence Plots and Their Quantifications: Expanding Horizons* [online] Springer International Publishing, Switzerland, 2016 [cit. 2020-04-10]. Dostupné z: <https://books.google.cz/books?id=4fY0DAAAQBAJ&printsec=frontcover&dq=recurrence+plot+and+their+quantification&hl=cs&sa=X&ved=0ahUKEwjQsIOYhbnpAhU3UBUIHXbEAwYQ6AEIMjABv=onepage&qf=false>.
- [34] Webber Ch. L., Marwan N.: *Recurrence Quantification Analysis: Theory and Best Practices* [online] Springer International Publishing, Switzerland, 2015 [cit. 2020-03-08]. Dostupné z: <https://books.google.cz/books?id=af4sBAAAQBAJ&printsec=frontcover&dq=recurrence+plot+and+their+quantification&hl=cs&sa=X&ved=0ahUKEwjQsIOYhbnpAhU3UBUIHXbEAwYQ6AEIKDAAv=onepage&q&f=false>
- [35] Wickham H.: *ggplot2: Elegant Graphics for Data Analysis* [online] Springer-Verlag, New York, 2016 [cit. 2020-02-10]. Dostupné z: <https://ggplot2.tidyverse.org>.
- [36] Wickham H., Henry L.: *tidyr: Tidy Messy Data* [online] [cit. 2020-02-07]. Dostupné z: <https://cran.r-project.org/web/packages/tidyr/tidyr.pdf>
- [37] Wu D. W.-L., Anderson N. C., Bischof W. F., Kingstone A.: *Temporal dynamics of eye movements are related to differences in scene complexity and clutter* [online] Journal of Vision August, Vol.14(8), 2014 [cit. 2020-03-01]. Dostupné z: <https://doi.org/10.1167/14.9.8>
- [38] Yarbus, A. L., Haigh, B., Riggs, L. A.: *Eye movements and vision* [online] Plenum press, vol. 2, New York, 1967 [cit. 2020-03-15]. Dostupné z: [http://wexler.free.fr/library/files/yarbus%20\(1967\)%20eye%20movements%20and%20vision.pdf](http://wexler.free.fr/library/files/yarbus%20(1967)%20eye%20movements%20and%20vision.pdf)

- [39] Young L. R., Sheena D.: *Methods & Designs: Survey of eye movement recording methods* [online] Behavior Research Methods & Instrumentation, vol. 7(5), s. 397-429, 1975 [cit. 2020-03-06]. Dostupné z: <https://link.springer.com/content/pdf/10.3758/BF03201553.pdf>

# Přílohy

Seznam příloh:

- příloha 1 – algoritmus funkcí pro metodu fixní vzdálenosti
- příloha 2 – CD přiložené k bakalářské práci obsahující záznam eye-tracking testu a kompletní algoritmus funkcí pro všechny metody, které jsme u rekurentní kvantifikační analýzy použili

# Příloha 1

V této části představíme celý algoritmus funkcí, které vedly k výsledkům praktické části v případě metody fixní vzdálenosti bez uvažování doby trvání.

Nejprve si nahrajeme datový soubor a z něj vybereme jednoho účastníka, který se díval na určitou mapu. V našem případě se jednalo o respondenta P16, který se díval na mapu 9. Přeznačili jsme si proměnné z vybraných dat. Pro vložení obrázků pod graf využijeme funkce `rasterGrob` z knihovny `grid`. Obrázek vkládáme ve formě png - je třeba mít balíček `png`. Vytvořili jsme novou tabulku `data`, která obsahuje všechny důležité informace, které budeme potřebovat.

```
data = read.delim("Map09+Map03-Event Statistics - Single.txt", sep = "\t")
mapa9 = data[data$Stimulus == "09-M1-CX-SI-VE.jpg", ]
P16 = mapa9[mapa9$Participant == "P16", ]

dataX = P16$Fixation.Position.X.px. # X-souradnice
dataY = P16$Fixation.Position.Y.px. # Y-souradnice
delka_fixace = P16$Event.Duration.ms. # delka fixace pozorovani
mista_zajmu = P16$AOI.Name # mista zajmu
n = length(dataX) # pocet mereni
T = sum(delka_fixace) # celkova doba fixace
dataY = -dataY # Y-souradnice (transformovane)

img9 = readPNG("09-M1-CX-SI-VE.png", info = TRUE) # obrazek v png
rozmary = attr(img9, "dim") # rozmary obrazku(1.atrubut vyska, 2.atribut delka)
g = rasterGrob(img9, interpolate=FALSE)

img9_AOI = readPNG("AOI_M9.png", info = TRUE) # obrazek rozdeleny na AOI v png
f = rasterGrob(img9_AOI, interpolate=FALSE)

data = data.frame(dataX, dataY, dataY, delka_fixace, mista_zajmu, mista_zajmu_
  numeric = mista_zajmu, radek_ctverce = 0, sloupec_ctverce = 0)

levels(data$mista_zajmu_numeric) = c(c(1:length(levels(mista_zajmu)))) # zmena
  na cisla, ktere jsou as.character
data$mista_zajmu_numeric = as.numeric(data$mista_zajmu_numeric) # zmena
  na as.numeric
```

Pro vizualizaci jsme využili funkci `zakladni_grafy`, jejíž vstupní parametry jsou tabulka, se kterou pracujeme, hodnoty souřadnic  $x$  a  $y$  (transformované), délka fixací, celková doba trvání, typ metody, který uvažujeme (kruh, ctverec, AOI), parametr  $h$  určuje hodnotu, která říká, jak velký bude kruh kolem fixace v případě uvažování doby fixace. Dále zadáváme rozměr čtverce a současně s tím i poloměr. Zvolené hodnoty jsou ideální pro následné uložení grafu ve velikosti 1200 x 800 pixelů.

```

zakladni_grafy = function (data, x = dataX, y = dataY, delka_fixace, T, type =
  typ_okoli, h = rozmer_okoli_graf, r = polomer(rozmer_ctverce), s_dobou_
  fixace) {
  if (type == "kruh") {
    if (s_dobou_fixace == "F") {
      spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(g, xmin =
        0, xmax = rozmery [2], ymin = - rozmery [1], ymax = 0) +
        geom_point(col = "steelblue", size = 4) + geom_path (col = "black", size
          = 0.2) +
        geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
          jitter(width=1,height=1), size = 5) +
        labs(title = "POZICE FIXACE", x = "x", y = "y") +
        scale_x_continuous (expand = c(0,0), limits = c(0, rozmery[2]), labels =
          NULL) +
        scale_y_continuous (expand = c(0,0), limits = c(-rozmery[1], 0), labels
          = NULL) +
        theme(
          panel.background=element_blank(),
          axis.ticks = element_blank(),
          axis.text = element_blank()
        )
    }
    else if (s_dobou_fixace == "T") {
      spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(g, xmin =
        0, xmax = rozmery [2], ymin = -rozmery [1], ymax = 0) +
        geom_point(shape = 20, color = "steelblue" , fill = "steelblue", size =
          (delka_fixace/T)*h, alpha = 0.4) +
        geom_path(col = "black", size = 0.2) + geom_point(size = 1.5, col = "
          blue") +
        geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
          jitter(width=1,height=1), size = 5) +
        labs(title = "POZICE FIXACE", x = "x", y = "y") +
        scale_x_continuous (expand = c(0,0), limits = c(0, rozmery[2]), labels =
          NULL) +
        scale_y_continuous (expand = c(0,0), limits = c(-rozmery[1], 0), labels
          = NULL) +
        theme(
          panel.background=element_blank(),
          axis.ticks = element_blank(),
          axis.text = element_blank()
        )
    }
    print (spojnicovy_graf)
  }
  if (type == "ctverec") {
    if (s_dobou_fixace == "F") {
      spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(g, xmin =
        0, xmax = rozmery [2], ymin = -rozmery [1], ymax = 0) +
        geom_point(col = "steelblue", size = 4) +
        geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
          jitter(width=1,height=1), size = 5) +
        geom_path (col = "black", size = 0.2) +
        labs(title = "POZICE FIXACE", x = "x", y = "y") +
        scale_x_continuous (expand = c(0,0), limits = c(0, rozmery[2]), labels =
          NULL) +
        scale_y_continuous (expand = c(0,0), limits = c(-rozmery[1], 0), labels
          = NULL) +
        geom_hline (yintercept=seq(-rozmery[1], 0, by = r), col = "black", size
          = 0.3) +
        geom_vline (xintercept=seq(0, rozmery[2], by = r), col = "black", size =
          0.3) +
        theme(
          panel.background=element_blank(),
          axis.ticks = element_blank(),

```



```

    axis.text = element_blank()
  )
}
else if (s_dobou_fixace == "T") {
  spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(g, xmin =
    0, xmax = rozmer [2], ymin = -rozmer [1], ymax = 0) +
  geom_point(size = (delka_fixace/T)*h, alpha = 0.4, shape = 20, colour =
    "steelblue") +
  geom_path(col = "black") + geom_point(size = 1.5, col = "blue") +
  geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
    jitter(width=1,height=1), size = 5) +
  labs(title = "POZICE FIXACE", x = "x", y = "y") +
  scale_x_continuous (expand = c(0,0), limits = c(0, rozmer[2]), labels =
    NULL) +
  scale_y_continuous (expand = c(0,0), limits = c(-rozmer[1], 0), labels
    = NULL) +
  geom_hline (yintercept=seq(-rozmer[1], 0, by = r), col = "black") +
  geom_vline (xintercept=seq(0, rozmer[2], by = r), col = "black") +
  theme(
    panel.background=element_blank(),
    axis.ticks = element_blank(),
    axis.text = element_blank()
  )
}
print (spojnicovy_graf)
}
if (type == "AOI") {
  if (s_dobou_fixace == "F") {
    spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(f, xmin =
      0, xmax = rozmer[2], ymin = -rozmer [1], ymax = 0) +
    geom_point(aes(color = mista_zajmu), size = 4) + scale_color_manual(
      values = c("-" = "CCCCCC", "A" = "#993399", "B" = "#FF6600", "Map" =
        "#006600", "Scale" = "#CC3333", "Title" = "#0000FF")) +
    geom_path(col = "black", size = 0.2) +
    geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
      jitter(width=1,height=1), size = 5) +
    labs(title = "POZICE FIXACE", x = "x", y = "y", color = "oblasti zajmu")
    +
    scale_x_continuous (expand = c(0,0), limits = c(0, rozmer[2]), labels =
      NULL) +
    scale_y_continuous (expand = c(0,0), limits = c(-rozmer[1], 0), labels
      = NULL) +
    theme(
      panel.background=element_blank(),
      axis.ticks = element_blank(),
      axis.text = element_blank(),
      legend.position = "none"
    )
  }
  else if (s_dobou_fixace == "T") {
    spojnicovy_graf = ggplot (data, aes (x, y)) + annotation_custom(f, xmin =
      0, xmax = rozmer [2], ymin = -rozmer [1], ymax = 0) +
    geom_point(aes(color = mista_zajmu), size = (delka_fixace/T)*h, alpha =
      0.4) + scale_color_manual(values = c("-" = "CCCCCC", "A" = "#993399",
        "B" = "#FF6600", "Map" = "#006600", "Scale" = "#CC3333", "Title" =
        "#0000FF")) +
    geom_point(aes(color = mista_zajmu), size = 1.5) +
    geom_path(col = "black", size = 0.2) +
    geom_text_repel(aes (label = c(1:n)), col = "black", position=position_
      jitter(width=1,height=1), size = 5) +
    labs(title = "POZICE FIXACE", x = "x", y = "y", color = "oblasti zajmu")
    +
    scale_x_continuous (expand = c(0,0), limits = c(0, rozmer[2]), labels =
      NULL) +
    scale_y_continuous (expand = c(0,0), limits = c(-rozmer[1], 0), labels

```

```

        = NULL) +
    theme(
      panel.background=element_blank(),
      axis.ticks = element_blank(),
      axis.text = element_blank(),
      legend.position = "none"
    )
  }
  print (spojnicovy_graf)
}
}

grafy = zakladni_grafy (data, dataX, dataY, delka_fixace, T,"kruh", 400, 60,"F")

```

Funkce `vzdalenost` nám umožnila určit pomocí eukleidovské metriky vzdálenost dvou fixací. Vstupní hodnoty jsou x-ové a y-ové souřadnice fixací  $f_i$  a  $f_j$ .

```

vzdalenost = function (x1, x2, y1, y2)
{
  v = sqrt ((x1 - x2)^2 + (y1 - y2)^2) # eukleidovska metrika
  return (v)
}

```

Funkce `matice_vzdalenosti`, s využitím předchozí funkce `vzdalenost`, vytvoří matici, jejíž prvky jsou jednotlivé vzdálenosti mezi dvěma fixacemi.

```

matice_vzdalenosti = function (x = dataX, y = dataY) {
  MV = matrix (data = NA, nrow = n, ncol = n)
  for (i in (1 : n))
    for (j in (1 : n)) {
      MV[i,j] = vzdalenost (x[i],x[j], y[i], y[j])
    }
  MV = as.data.frame(MV)
  print (MV)
}

mat_vzdalenosti = matice_vzdalenosti (data$dataX, data$dataY)

```

Pro vytvoření matice rekurence jsme použili funkci `matice_rekurence`, jejíž vstupní hodnoty jsou x-ové souřadnice, y-ové souřadnice, poloměr a informace, zda uvažujeme dobu trvání `T` - `true` (uvažujeme), nebo `F` - `false` (neuvažujeme).

```

matice_rekurence = function (x=dataX, y=dataY, delka_fixace, r=polomer, s_dobou_fixace) {
  MR = matrix (data = NA, nrow = n, ncol = n) # do matice ukladame nase hodnoty
  if (s_dobou_fixace == "F") {
    for (i in (1 : n))
      for (j in (1 : n)) {
        if (vzdalenost (x[i], x[j], y[i], y[j]) < r ) {MR [i,j] = 1
        } else {MR [i,j] = 0} # 1 pokud je mensi nez polomer, jinak 0
        MR = as.data.frame (MR) # jako data.frame
        colnames (MR) = c(c(1:n)) # prejmenovani sloupce 1:n
      }
    print(MR)
  }
  else if (s_dobou_fixace == "T") {
    for (i in (1 : n))

```

```

    for (j in (1 : n)) {
      if (vzdalenost (x[i], x[j], y[i], y[j]) < r ) {MR [i,j] = (delka_fixace
        [i] + delka_fixace [j])
      } else {MR [i,j] = 0} # soucet casu pokud je mensi nez polomer, jinak 0
      MR = as.data.frame (MR) # jako data.frame
      colnames (MR) = c(c(1:n)) # prejmenovani sloupce 1:n
    }
  }
  print(MR)
}
}
mat_rekurence = matice_rekurence(data$dataX, data$dataY, data$delka_fixace, 60,
  "F") # matice rekurence pro polomer mensi nez 60, bez delky fixace

```

Z rekurentní matice jsme nebyli schopni vytvořit graf rekurence, proto jsme nejdříve matici seřadili pomocí funkce `serazeni`. Z původní matice rekurence jsme vytvořili seřazenou matici, jejíž sloupce označují číslo řádku matice, číslo sloupce matice a hodnotu na této pozici. Současně jsme ponechali pouze prvky, které mají hodnotu 1. Do vstupních hodnot zadáváme matici rekurence a také typ okolí - kruh, ctverec, AOI.

```

serazeni = function (x = matice_rekurence, type = typ_okoli) {
  if (type == "kruh") {
    x = cbind (c(1:n), x) # pridani sloupce s indexy radku
    colnames (x) = c("radek", c(1:n)) # prejmenovani sloupce
    S2 = gather (x, key = "sloupec", value = "hodnoty", - "radek") # spojime
      do matice...radek, sloupec, hodnoty
    S2 [S2 == 0] = NA # misto 0 piseme NA
    S1 = na.omit (S2) # ze serazene matice, vybereme jenom 1 (bez NA)
    S = as.data.frame (sapply(S1, as.numeric))
  }

  else if ((type == "AOI" | (type == "ctverec"))) {
    x = cbind (c(1:n), x) # pridani sloupce s indexy radku
    colnames (x) = c("radek", c(1:n)) # prejmenovani sloupce
    S2 = gather (x, key = "sloupec", value = "hodnoty", - "radek") # spojime
      do matice...radek, sloupec, hodnoty
    S2 [S2 == 0] = NA # misto 0 piseme NA
    S1 = na.omit (S2) # ze serazene matice, vybereme jenom 1 (bez NA)
    S = as.data.frame (sapply(S1, as.numeric))
  }
}

serazena_m_rekurence = serazeni(mat_rekurence, "kruh")

```

Následně už bylo možné vytvořit graf rekurence pomocí funkce `recurrence_plot`. Vstupní hodnoty, které zadáváme do funkce jsou seřazená matice rekurence, celková doba trvání a informace, zda uvažujeme dobu trvání T - true (uvažujeme), nebo F - false (neuvažujeme).

```

recurrence_plot = function (x = serazena_m_rekurence, T, type, s_dobou_fixace) {
  if (s_dobou_fixace == "F") {
    if ((type == "kruh") | (type == "ctverec")) {
      RP = ggplot(x, aes (radek, sloupec)) +
        geom_point(col = "steelblue", size = 1.5) + geom_point(col="steelblue",
          size = 1.5) + labs (title = "RECURRENCE PLOT", x = "index radku", y =
            "index sloupce") + theme_bw()
    }
  }
}

```

```

if (type == "AOI") {
  RP = ggplot(x, aes (radek, sloupec, color=as.factor(barva))) +
    geom_point(aes(color = x$barva), size = 1.5) + scale_color_manual(values
      = c("-" = "CCCCCC", "A" = "#993399", "B" = "#FFCC33", "Map" = "
        #006600", "Scale" = "#CC3333", "Title" = "#0000FF")) +
    labs (title = "RECURRENCE PLOT", x = "index radku", y = "index sloupce")
    + guides(col=FALSE) +
    theme_bw()
}
}
if (s_dobou_fixace == "T"){
  if ((type == "kruh") | (type == "ctverec")) {
    RP = ggplot(x, aes (radek, sloupec)) + geom_point(col="steelblue", alpha =
      0.5, size = x$hodnoty/T*100) + geom_point(col = "dark blue",shape = 20,
      size = 0.6) + labs(title = "RECURRENCE PLOT", x = "index radku", y = "
      index sloupce") + theme_bw()
  }
  if (type == "AOI") {
    RP = ggplot(x, aes (radek, sloupec, color=as.factor(barva))) +
      geom_point(aes(color = x$barva), alpha = 0.5, size = x$hodnoty/T*100) +
      scale_color_manual(values = c("-" = "CCCCCC", "A" = "#993399", "B" =
        "#FFCC33", "Map" = "#006600", "Scale" = "#CC3333", "Title" = "#0000
        FF")) +
      labs (title = "RECURRENCE PLOT", x = "index radku", y = "index sloupce")
      + guides(col=FALSE) +
      theme_bw()
  }
}
}
print(RP)
}
graf_rekurence = recurrence_plot(serazena_m_rekurence, T, "kruh", "F")

```

Pro stanovení charakteristik rekurence potřebujeme pouze data z horního trojúhelníku rekurentního grafu. Využili jsme funkci `jen_horni_trojuhelnik`, která vybírá ze seřazené matice rekurence prvky, pro které platí, že hodnota ve sloupci je větší než hodnota v řádku. Následně jsme k těmto vybraným datům přidali sloupce, které označují délku diagonály, délku horizontály a délku vertikály. Všechny zatím mají hodnotu NA.

```

jen_horni_trojuhelnik = function (x = serazena_m_rekurence) { # vyber dat jen z
  horniho trojuhelniku
  HT1 = which (x$radek < x$sloupec) # pro kttere prvky plati, ze je hodnota ve
    sloupci vetsi nez v radku
  HT = x[HT1, ]
  print (HT)
}

HT1 = jen_horni_trojuhelnik (serazena_m_rekurence)
HT = data.frame(HT1, delka_diagonaly = NA, delka_horizontaly = NA, delka_
  vertikaly = NA) # nova tabulka s daty z horniho trojuhelniku, pridame
  sloupce

```

První z několika funkcí nutných pro výpočet charakteristik, bylo zjištění délky diagonálních linií. K tomu jsme použili funkci `diagonaly`. Pro vstupní data k této funkci použijeme tabulku s daty z horního trojúhelníku, s přidanými sloupci pro délky jednotlivých linií. Výsledkem této funkce jsou hodnoty určující délky

diagonál, které z bodů vychází. Hlavní myšlenkou, kterou jsme použili je, že pokud z bodu se souřadnicemi  $\langle x, y \rangle$  vychází diagonální linie, pak se musí v datech vyskytovat i prvek se souřadnicemi  $\langle x + 1, y + 1 \rangle$  - diagonála délky 2,  $\langle x + 2, y + 2 \rangle$  - diagonála délky 3, atd.

```

diagonaly = function (x = horni_trojuhelnik) {
  for (i in c(1:nrow (x))) {
    endOfLine = FALSE
    Counter <- 1
    while(endOfLine == FALSE) {
      nextPointExists <- as.numeric(which((x$radek == (x$radek[i] + Counter)) &
        (x$sloupec == x$sloupec[i] + Counter)))
      if(is.empty (nextPointExists)) {
        endOfLine = TRUE
        x$delka_diagonaly [i] = Counter
      }
      Counter <- Counter + 1
    }
  }
  for (i in c(1:nrow (x))) {
    if (x$delka_diagonaly [i] == 1) { # pokud diagonalu tvorí jen 1 bod, pak
      prepiseme hodnotu na 0
      x$delka_diagonaly [i] = 0
    }
  }
  duration = as.numeric (max(x$delka_diagonaly)) # zjisteni maximalni delky
  diagonaly
  for (j in c(duration:2)) {
    a = x$radek[x$delka_diagonaly == j]
    b = x$sloupec [x$delka_diagonaly == j]
    if(is.empty(a)) {
      next
    }
    else {
      c = array(data = c(a,b), dim = c(length(a), 2))
      for (k in c(1:length(a))) {
        for (i in c(1:(j-1))) {
          vdiagonale = as.numeric(which((x$radek == c[k,1] + i) & (x$sloupec ==
            c[k,2]+i)))
          x$delka_diagonaly [vdiagonale] = 1
          for (i in c(1:nrow (x))) {
            if (x$delka_diagonaly [i] == 1) {
              x$delka_diagonaly [i] = 0
            }
          }
        }
      }
    }
  }
  x = as.data.frame(x)
  x$delka_diagonaly = as.numeric(x$delka_diagonaly)
}
print(x)
}

HT_D = diagonaly (HT)

```

Následně jsme schopni určit  $|D_L|$  - počet bodů, které tvoří diagonální linie, pomocí funkce D\_L, která říká, kolik diagonál se v grafu vyskytuje a sečte hodnoty bodů, které tvoří diagonální linie.

```

D_L = function (x = horni_trojuhelnik, s_dobou_fixace) {
  if (s_dobou_fixace == "F") {
    x$delka_diagonaly = as.numeric(x$delka_diagonaly)
    body_v_diagonale = sum(x$delka_diagonaly)
    pocet_diagonal = length (x$delka_diagonaly[x$delka_diagonaly != 0])
    return(list(pocet_diagonal = pocet_diagonal, body_v_diagonale = body_v_
      diagonale))
  }
  if (s_dobou_fixace == "T") {
    pocet_diagonal = length(x$delka_diagonaly[x$delka_diagonaly != 0])
    duration = as.numeric (max(x$delka_diagonaly))
    if (duration == 0) {
      body_v_diagonale = sum(x$delka_diagonaly[x$delka_diagonaly != 0])
      hodnoty_v_diagonale = sum(x$hodnoty [x$delka_diagonaly != 0])
      return(list(pocet_diagonal = pocet_diagonal, body_v_diagonale = body_v_
        diagonale, hodnoty_v_diagonale = hodnoty_v_diagonale))
    }
    else {
      for (j in c(duration:2)) {
        a = x$radek[x$delka_diagonaly == j]
        b = x$sloupec [x$delka_diagonaly == j]
        if(is.empty(a)) {
          next
        }
        else {
          c = array(data = c(a,b), dim = c(length(a), 2))
          for (k in c(1:length(a))) {
            for (i in c(1:(j-1))) {
              vdiagonale = as.numeric(which((x$radek == c[k,1] + i) & (x$sloupec
                == c[k,2]+i)))
              x$delka_diagonaly [vdiagonale] = 1
            }
          }
        }
      }
      for (i in c(1:nrow(x))) {
        if (x$delka_diagonaly [i] != 0) {
          x$delka_diagonaly [i] = 1
        }
      }
      body_v_diagonale = sum(x$delka_diagonaly[x$delka_diagonaly != 0])
      hodnoty_v_diagonale = sum(x$hodnoty [x$delka_diagonaly != 0])
      return(list(pocet_diagonal = pocet_diagonal, body_v_diagonale = body_v_
        diagonale, hodnoty_v_diagonale = hodnoty_v_diagonale))
    }
  }
}

```

V případě zjištění horizontálních linií jsme použili funkci `horizontaly`, která vychází z toho, že pokud z bodu se souřadnicemi  $\langle x, y \rangle$  vychází horizontální linie, pak musí být v datech i bod se souřadnicemi  $\langle x + 1, y \rangle$ ,  $\langle x + 2, y \rangle$ .

```

horizontaly = function (x = horni_trojuhelnik) {
  for (i in c(1:nrow (x))) {
    endOfLine = FALSE
    Counter <- 1
    while(endOfLine == FALSE) {
      nextPointExists <- as.numeric(which((x$radek == (x$radek[i] + Counter)) &
        (x$sloupec == x$sloupec[i])))
      if(is.empty (nextPointExists)) {
        endOfLine = TRUE
        x$delka_horizontaly [i] = Counter
      }
    }
  }
}

```

```

    }
    Counter <- Counter + 1
  }
}
for (i in c(1:nrow (x))) {
  if (x$delka_horizontaly [i] == 1) { # pokud horizontalu tvori jen 1 bod,
    pak prepiseme hodnotu na 0
    x$delka_horizontaly [i] = 0
  }
}
duration = as.numeric (max(x$delka_horizontaly)) # zjisti maximalni delky
horizontaly
for (j in c(duration:2)) {
  a = x$radek[x$delka_horizontaly == j]
  b = x$sloupec [x$delka_horizontaly == j]
  if(is.empty(a)) {
    next
  }
  else {
    c = array(data = c(a,b), dim = c(length(a), 2))
    for (k in c(1:length(a))) {
      for (i in c(1:(j-1))) {
        vhorizontale = as.numeric(which((x$radek == c[k,1] + i) & (x$sloupec
          == c[k,2])))
        x$delka_horizontaly [vhorizontale] = 1
        for (i in c(1:nrow (x))) {
          if (x$delka_horizontaly [i] == 1) {
            x$delka_horizontaly [i] = 0
          }
        }
      }
    }
  }
}
x = as.data.frame(x)
x$delka_horizontaly = as.numeric(x$delka_horizontaly)
}
}
print(x)
}
HT_H = horizontaly (HT_D)

```

Ke zjištění počtu bodů, které tvoří horizontální linie  $|H_L|$  jsme využili funkci `H_L`. Určuje, kolik horizontálních linií je v grafu a sčítá hodnoty rekurentních bodů, které tvoří horizontální linie.

```

H_L = function (x = horni_trojuhelnik, s_dobou_fixace) {
  if (s_dobou_fixace == "F") {
    x$delka_horizontaly = as.numeric(x$delka_horizontaly)
    body_v_horizontale = sum(x$delka_horizontaly)
    pocet_horizontal = length (x$delka_horizontaly[x$delka_horizontaly != 0])
    return(list(pocet_horizontal = pocet_horizontal, body_v_horizontale = body_v_
      _horizontale))
  }
  if (s_dobou_fixace == "T") {
    pocet_horizontal = length(x$delka_horizontaly[x$delka_horizontaly != 0])
    duration = as.numeric (max(x$delka_horizontaly))
    if (duration == 0) {
      body_v_horizontale = sum(x$delka_horizontaly[x$delka_horizontaly != 0])
      hodnoty_v_horizontale = sum(x$hodnoty [x$delka_horizontaly != 0])
      return(list(pocet_horizontal = pocet_horizontal, body_v_horizontale = body_
        _v_horizontale, hodnoty_v_horizontale = hodnoty_v_horizontale))
    }
  }
}

```

```

else {
  for (j in c(duration:2)) {
    a = x$radek[x$delka_horizontaly == j]
    b = x$sloupec [x$delka_horizontaly == j]
    if(is.empty(a)) {
      next
    }
    else {
      c = array(data = c(a,b), dim = c(length(a), 2))
      for (k in c(1:length(a))) {
        for (i in c(1:(j-1))) {
          vhorizontale = as.numeric(which ((x$radek == c[k,1] + i) & (x$
            sloupec == c[k,2])))
          x$delka_horizontaly [vhorizontale] = 1
        }
      }
    }
  }
  for (i in c(1:nrow(x))) {
    if (x$delka_horizontaly [i] != 0) {
      x$delka_horizontaly [i] = 1
    }
  }
  body_v_horizontale = sum(x$delka_horizontaly[x$delka_horizontaly != 0])
  hodnoty_v_horizontale = sum(x$hodnoty [x$delka_horizontaly != 0])
  return(list(pocet_horizontaly = pocet_horizontaly, body_v_horizontale = body
    _v_horizontale, hodnoty_v_horizontale = hodnoty_v_horizontale))
}
}
}

```

Pro určení délky vertikálních linií jsme ve funkci `vertikaly` vycházeli z toho, že pokud z bodu se souřadnicemi  $\langle x, y \rangle$  vychází vertikální linie, pak se musí v datech vyskytovat i prvek se souřadnicemi  $\langle x, y + 1 \rangle$ ,  $\langle x, y + 2 \rangle$ . Všechny délky linií jsme uložili do tabulky `horni_trojuhelnik_v`.

```

vertikaly = function (x = horni_trojuhelnik) {
  for (i in c(1:nrow (x))) {
    endOfLine = FALSE
    Counter <- 1
    while(endOfLine == FALSE) {
      nextPointExists <- as.numeric(which((x$radek == (x$radek[i])) & (x$sloupec
        == x$sloupec[i] + Counter)))
      if(is.empty (nextPointExists)) {
        endOfLine = TRUE
        x$delka_vertikaly [i] = Counter
      }
      Counter <- Counter + 1
    }
  }
  for (i in c(1:nrow (x))) {
    if (x$delka_vertikaly [i] == 1) { # pokud vertikalu tvori jen 1 bod, pak
      prepiseme hodnotu na 0
      x$delka_vertikaly [i] = 0
    }
  }
  duration = as.numeric (max(x$delka_vertikaly)) # zjisteni maximalni delky
  vertikaly
  for (j in c(duration:2)) {
    a = x$radek[x$delka_vertikaly == j]
    b = x$sloupec [x$delka_vertikaly == j]
  }
}

```



```

if(is.empty(a)) {
  next
}
else {
  c = array(data = c(a,b), dim = c(length(a), 2))
  for (k in c(1:length(a))) {
    for (i in c(1:(j-1))) {
      vevertikale = as.numeric(which((x$radek == c[k,1]) & (x$sloupec == c[k
,2] + i)))
      x$delka_vertikaly [vevertikale] = 1
      for (i in c(1:nrow (x))) {
        if (x$delka_vertikaly [i] == 1) {
          x$delka_vertikaly [i] = 0
        }
      }
    }
  }
  x = as.data.frame(x)
  x$delka_vertikaly = as.numeric(x$delka_vertikaly)
}
print(x)
}

HT_V = vertikaly (HT_H)

```

Poslední, co nám chybělo, abychom mohli určit míry rekurence je počet bodů, které tvoří vertikální linie  $|V_L|$ . To jsme zjistili díky funkci  $V_L$ , která nám říká, kolik vertikálních linií se v grafu objevuje a sčítá hodnoty rekurentních bodů, které tvoří vertikální linie.

```

V_L = function (x = horni_trojuhelnik, s_dobou_fixace) {
  if (s_dobou_fixace == "F") {
    x$delka_vertikaly = as.numeric(x$delka_vertikaly)
    body_ve_vertikale = sum(x$delka_vertikaly)
    pocet_vertikal = length (x$delka_vertikaly[x$delka_vertikaly != 0])
    return(list(pocet_vertikal = pocet_vertikal, body_ve_vertikale = body_ve_
vertikale))
  }
  if (s_dobou_fixace == "T") {
    pocet_vertikal = length(x$delka_vertikaly[x$delka_vertikaly != 0])
    duration = as.numeric (max(x$delka_vertikaly))
    if (duration == 0) {
      body_ve_vertikale = sum(x$delka_vertikaly[x$delka_vertikaly != 0])
      hodnoty_ve_vertikale = sum(x$hodnoty [x$delka_vertikaly != 0])
      return(list(pocet_vertikal = pocet_vertikal, body_ve_vertikale = body_ve_
vertikale, hodnoty_ve_vertikale = hodnoty_ve_vertikale))
    }
    else {
      for (j in c(duration:2)) {
        a = x$radek[x$delka_vertikaly == j]
        b = x$sloupec [x$delka_vertikaly == j]
        if(is.empty(a)) {
          next
        }
        else {
          c = array(data = c(a,b), dim = c(length(a), 2))
          for (k in c(1:length(a))) {
            for (i in c(1:(j-1))) {
              vevertikale = as.numeric(which ((x$radek == c[k,1]) & (x$sloupec
== c[k,2] + i)))
            }
          }
        }
      }
    }
  }
}

```

```

        x$delka_vertikaly [vevertikale] = 1
      }
    }
  }
  for (i in c(1:nrow(x))) {
    if (x$delka_vertikaly [i] != 0) {
      x$delka_vertikaly [i] = 1
    }
  }
  body_ve_vertikale = sum(x$delka_vertikaly[x$delka_vertikaly != 0])
  hodnoty_ve_vertikale = sum(x$hodnoty [x$delka_vertikaly != 0])
  return(list(pocet_vertikal = pocet_vertikal, body_ve_vertikale = body_ve_
    vertikale, hodnoty_ve_vertikale = hodnoty_ve_vertikale))
}
}
}

```

Pro určení měr rekurence jsme využili funkci `charakteristika`, jejíž vstupní hodnoty jsou příslušná matice rekurence, součet hodnot rekurentních bodů, které tvoří diagonální linie, součet hodnot rekurentních dobů, které tvoří horizontální linie, součet hodnot bodů, které tvoří vertikální linie a informace, zda uvažujeme dobu trvání fixací.

```

D_L1 = D_L (HT_V, "F")
H_L1 = H_L (HT_V, "F")
V_L1 = V_L (HT_V, "F")

```

```

charakteristika = function(x = matice_rekurence, D_L, H_L, V_L, s_dobou_fixace)
{
  if (s_dobou_fixace == "F"){
    R = 0
    for (i in (1:(n-1))) {
      for (j in ((i+1):n)) {
        R = R + sum (x [i,j]) # pocet 1 v hornim trojuhelniku
      }
    }
    REC = 100*(2*R/(n*(n-1))) # procento rekurentnich bodu
    DET = 100*(D_L$body_v_diagonale/R)
    LAM = 100*(H_L$body_v_horizontale + V_L$body_ve_vertikale)/(2*R)
    CORM = 0
    for (i in (1:(n-1))) {
      for (j in ((i+1):n)) {
        CORM = CORM + (((j-i)*x[i,j]))/((n-1)*R)* 100
      }
    }
    return(list(pocet_opakovani=R, mira_rekurence=REC, mira_determinismu=DET,
      mira_laminarity=LAM, CORM=CORM))
  }
  if (s_dobou_fixace == "T") {
    R_t = 0 # pocet 1 v hornim trojuhelniku
    for (i in (1:(n-1))) {
      for (j in ((i+1):n)) {
        R_t = R_t + sum (x[i,j])
      }
    }
    REC_t = 100*R_t/((n-1)*T)
    DET_t = (100/R_t)*(D_L$hodnoty_v_diagonale)
    LAM_t = (100/(2*R_t))*(H_L$hodnoty_v_horizontale + V_L$hodnoty_ve_vertikale)
    CORM_t = 0
    for (i in (1:(n-1))) {
      for (j in ((i+1):n)) {

```

```
        CORM_t = CORM_t + 100*(((j-i)*x[i,j]))/(((n-1)^2)*T) # potreba mit
        matici rekurence s dobou trvani!
    }
}
return(list(doba_opakovani=R_t, mira_rekurence_t=REC_t, mira_determinismu_t=
    DET_t,
        mira_laminarity_t=LAM_t, CORM_t=CORM_t))
}
}
miry=charakteristika(mat_rekurence, D_L1, H_L1, V_L1, "F")
```