

**Univerzita Palackého v Olomouci**

**Přírodovědecká fakulta**

**Katedra geoinformatiky**

**Bc. Jan PROCHÁZKA**

**VYUŽITÍ ČASOVÝCH ŘAD V ANALÝZE DAT  
Z EYE TRACKING SYSTÉMU**

**Magisterská práce**

**Vedoucí práce: Doc. Mgr. Jiří Dvorský, Ph.D.**

**Olomouc 2013**

## **Čestné prohlášení**

Prohlašuji, že jsem magisterskou práci magisterského studia oboru Geoinformatika vypracoval samostatně pod vedením Doc. Mgr. Jiřího Dvorského, Ph.D.

Všechny použité materiály a zdroje jsou citovány s ohledem na vědeckou etiku, autorská práva a zákony na ochranu duševního vlastnictví.

Všechna poskytnutá i vytvořená digitální data nebudu bez souhlasu školy poskytovat.

V Olomouci 21. dubna 2013

---

Děkuji vedoucímu práce Doc. Mgr. Jiřímu Dvorskému, Ph.D. za podněty a připomínky při vypracování práce a Bc. Jakubu Klusovi za cenné rady, zejména při práci v programu MATLAB.

Dále děkuji Ing. Tomáši Kocyanovi za poskytnutý program a Mgr. Stanislavu Popelkovi a Mgr. Alžbětě Brychtové za poskytnutá data.

Vložený originál **zadání** bakalářské/magisterské práce (s podpisy vedoucího katedry, vedoucího práce a razítkem katedry). Ve druhém výtisku práce je vevázána fotokopie zadání.

# OBSAH

<b>ÚVOD</b> .....	<b>7</b>
<b>1 CÍLE PRÁCE</b> .....	<b>8</b>
<b>2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ</b> .....	<b>9</b>
2.1 Použitá data .....	9
2.2 Požité programy .....	9
2.3 Postup zpracování .....	10
<b>3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY</b> .....	<b>12</b>
3.1 Eye - tracking .....	12
3.1.1 Metody měření.....	13
3.1.2 Možnosti vizualizace .....	14
3.2 Voting Experts algoritmus .....	15
3.2.1 Sestavení stromu n-gramů a standardizace výsledků .....	16
3.2.2 Hlasování expertů .....	18
3.2.3 Vytvoření nových segmentů .....	18
3.2.4 Zvýšení výkonu Voting Experts algoritmu.....	19
3.3 Borcení časové osy.....	19
3.3.1 Princip DTW.....	20
3.3.2 Využití DTW při hledání podsekvencí .....	22
3.4 Voting Experts s DTW post-procesem .....	23
<b>4 PŘEDZPRACOVÁNÍ DAT</b> .....	<b>26</b>
4.1 Mortonův rozklad.....	26
4.2 Analýza souřadnicových komponent .....	27
4.2.1 Načtení proměnných.....	27
4.2.2 Zjednodušení dat.....	28
4.2.3 Rozdělení dat na fixace a sakády.....	29
4.2.4 Vytvoření HeatMap .....	30
4.2.5 Vykreslení sakád.....	31
4.2.6 Vytvoření nových proměnných .....	32
4.2.7 Vytvoření videozáznamu .....	32
<b>5 IMPLEMENTACE VOTING EXPERTS ALGORITMU S DTW POST-PROCESEM</b> .....	<b>33</b>
5.1 Testování nastavení parametrů programu VE Console.....	33
5.2 Nasazení programu VE Console .....	35
5.3 Sčítání hlasů .....	35

<b>6</b>	<b>VIZUALIZACE A PROVNÁVÁNÍ SEGMENTŮ</b> .....	<b>37</b>
6.1	Vizuální porovnání.....	37
6.2	Porovnávání pomocí histogramů.....	38
<b>7</b>	<b>VÝSLEDKY</b> .....	<b>40</b>
<b>8</b>	<b>DISKUZE</b> .....	<b>45</b>
<b>9</b>	<b>ZÁVĚR</b> .....	<b>47</b>
	<b>POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE</b>	
	<b>SUMMARY</b>	
	<b>PŘÍLOHY</b>	

## ÚVOD

V současnosti existuje mnoho oblastí využití metody eye-tracking, jenž je založena na principu sledování pohybu lidských očí při vnímání obrazu. V této práci je využito dat naměřených touto metodou při již dříve probíhajícím testování na Katedře geoinformatiky UP v Olomouci, které se u respondentů zabývalo rozdílností vnímání mezi 2D a 3D mapami. Hlavním cílem této práce je analýzou využívajících časových řad, vyhledat podobné segmenty v těchto datech, které by svědčily o podobném přístupu čtení předkládaných map respondenty.

K nalezení těchto segmentů je konkrétně použito algoritmů Voting Experts a Borcení časovou osou. Jako zvolená implementace těchto algoritmů je použit program vytvořený Ing. Tomášem Kocyanem z Vysoké školy báňské - Technické univerzity Ostrava, který tyto algoritmy vzájemně kombinuje. Jelikož tento program vyžaduje data v jednorozměrném formátu, je v práci představena metodika, která pomocí skriptů v programu MATLAB, dvourozměrná data z eye-tracking systému předzpracovává do požadované podoby a následně provede řezy, které vytvoří požadované segmenty.

Tyto segmenty jsou na závěr graficky vizualizovány a vzájemně mezi sebou porovnány .

# 1 CÍLE PRÁCE

Hlavním cílem magisterské práce je vyhledat podobné segmenty v experimentálních datech z eye-tracking systému, svědčící o podobném postupu řešení úkolů respondenty. Jedním z dílčích úkolů práce je definice vhodných úloh pro využití eye-tracking, čímž se získají data k pozdějšímu zpracování. Následně je potřeba tyto experimentální data předzpracovat do takové podoby, aby byly použitelné pro analýzu pomocí časových řad, například Borcením časovou osou, Voting Experts algoritmem atd.

Výsledné segmenty získané analýzou je potřeba mezi sebou vzájemně porovnat. Tyto výsledky je nakonec nutné vhodným způsobem interpretovat a vizualizovat.



## 2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ

### 2.1 Použitá data

Jedním z cílů diplomové práce bylo definovat vhodné úlohy, nad kterými by probíhalo testování pomocí eye-tracking systému. Vzhledem k tomu, že na Katedře geoinformatiky Univerzity Palackého v Olomouci bylo Mgr. Stanislavem Popelkou a Mgr. Alžbětou Brychtovou již dříve řešeno testování založené na rozdílnosti vnímání mezi 2D a 3D mapami, bylo se svolením autorů využito těchto naměřených dat. Testování se zúčastnilo přibližně dvacet respondentů absolvujících minimálně jeden semestr výuky kartografie a dvacet dalších respondentů, kteří neměli žádné kartografické základy. Všem uživatelům bylo zobrazeno několik dvojic 2D a 3D map vždy vedle sebe (v náhodné pozici) a na základě předem položené otázky bylo testováno vnímání těchto map. Testování probíhalo na přístroji SMI RED 250 s vzorkovací frekvencí 120 Hz, kde přesná poloha oka byla snímána každých 8ms.

Bylo rozhodnuto, že jako testovací kolekce bude k účelu této diplomové práce použito tzv. raw dat (vysvětleno v kap. 3.1.3), nakonec pouze od dvaceti uživatelů (9 kartografů, 11 ne-kartografů), nad jednou dvojicí 2D a 3D map. Takováto vstupní data jsou uložena v textovém souboru (.txt) a nesou mj. informace o souřadnicové poloze pohledu pravého (R POR X [px], R POR Y [px]) i levého oka v daném čase (Time), nebo název předkládaného obrazu (Stimulus). Konkrétně tyto čtyři informace byly využity k dalšímu zpracování dat (Obr. 1).

Time	Type	Trial	L Dia X [px]	L Dia Y [px]	R Dia X [px]	R Dia Y [px]	L POR X [px]	L POR Y [px]	R POR X [px]	R POR Y [px]	Tir	Latency	L	V	R	V	Pug	Stimulus
3630860617	SMP	1	10.86	10.86	11.19	11.19	839.74	542.58	839.74	542.58	0	1582	0	0	1	0	10	1010 - 3d-2d.jpg
3630868839	SMP	1	10.73	10.73	11.17	11.17	842.85	545.43	842.85	545.43	0	1704	0	0	1	0	10	1010 - 3d-2d.jpg
3630877234	SMP	1	10.86	10.86	11.11	11.11	843.43	551.89	843.43	551.89	0	1594	0	0	1	0	10	1010 - 3d-2d.jpg
3630885584	SMP	1	10.86	10.86	11.11	11.11	843.21	552.32	843.21	552.32	0	1566	0	0	1	0	10	1010 - 3d-2d.jpg
3630893846	SMP	1	10.79	10.79	11.12	11.12	844.52	552.65	844.52	552.65	0	1683	0	0	1	0	10	1010 - 3d-2d.jpg
3630902212	SMP	1	10.53	10.53	11.17	11.17	846.03	554.85	846.03	554.85	0	1607	0	0	1	0	10	1010 - 3d-2d.jpg
3630910597	SMP	1	10.58	10.58	11.9	11.9	845.58	559.64	845.58	559.64	0	1608	0	0	1	0	10	1010 - 3d-2d.jpg
3630918911	SMP	1	10.61	10.61	11.8	11.8	846.78	559.55	846.78	559.55	0	1871	0	0	1	0	10	1010 - 3d-2d.jpg
3630927293	SMP	1	10.71	10.71	11.9	11.9	846.96	564.26	846.96	564.26	0	1965	0	0	1	0	10	1010 - 3d-2d.jpg

Obr. 1: Ukázka dat z eye-tracking systému s vyznačením používaných atributů.

### 2.2 Požité programy

Nejdůležitější část práce, tedy tvorba skriptů, kde první předzpracovával vstupní data, druhý vytvářel videozáznam uživatelského pohybu očí nad daným obrazem a poslední pomocí stanovené prahové hodnoty rozsekal výstupní data z programu VE Console na jednotlivé segmenty, které graficky vizualizoval, byly vytvořeny v programu MATLAB R2010b.

Jako zvolená implementace Voting Experts algoritmu byl použit program VE Console, vytvořený studentem doktorského studia na Vysoké škole báňské - Technické univerzitě Ostrava, Ing. Tomášem Kocyanem. Jedná se o program, jenž využívá Voting Experts

algoritmus obohacený o oboustranný průchod a tzv. DTW post-proces. Zatímco všechny funkce jsou volány pouze jako knihovny, nastavení jednotlivých parametrů je uloženo přímo ve vytvořené aplikaci. Kvůli jejich změně bylo zapotřebí upravit spouštěcí program, k čemuž bylo využito Microsoft Visual Studio 2010.

Pro tvorbu vektorových obrázků, na kterých bylo testováno nastavení parametrů předešlého programu byl využit editor vektorové grafiky Inkscape a pro výsledné vykreslení segmentů bylo užito sázecího systému LaTeX.

## 2.3 Postup zpracování

Po seznámení se s problematikou eye-tracking systému a nastudováním principu fungování algoritmů pracujících s časovými řadami, především pak Voting Experts a DTW (*Dynamic Time Warping, Borcení časové osy*) algoritmu, včetně jejich nejruznějších modifikací, bylo potřeba definovat vhodné úlohy, nad kterými by bylo prováděno testování eye-tracking systémem. Jak již bylo zmíněno v předcházející kapitole, pro účely této práce bylo rozhodnuto využít data z již dříve probíhajícího testování

Pro potřeby pozdější práce bylo však nutné tyto experimentální data předzpracovat a upravit je do potřebné podoby. V programu MATLAB byl tedy vytvořen skript, který nejprve načte potřebné hodnoty z dat neměřených metodou eye-tracking pro libovolně zvoleného uživatele nad vybraným stimulem (zkoumaným obrazem).

Vzhledem k tomu, že data obsahují přesný záznam toho, kam všude se uživatel díval, jedná se o velmi složitou a často nepřehlednou křivku. Aby bylo možné následně porovnávat jednotlivé segmenty této křivky, bylo třeba záznam zjednodušit. K tomuto účelu byla použita dolní propust (*Low-pass filter*). Skript následně pro představu vykreslil graf a obrázek jak data vypadala před a po použití filtru.

Další součástí skriptu bylo rozdělení vstupních dat na fixace a sakády. Toho bylo docíleno metodou, kdy na základě výpočtu hodnot délek mezi dvěma sousedními body křivky, byla rovněž vypočtena průměrná hodnota těchto délek a ta byla definována jako prahová hodnota. Potom všechny body křivky nad touto hodnotou byly označeny za sakády a body křivky pod prahovou hodnotou za fixace.

Aby bylo i lépe graficky názorné, jak uživatel vnímal předloženou mapu, bylo použito jedné z klasických vizualizačních metod nad daty eye-tracking systému - HeatMap. HeatMapa byla vypočtena konvolucí s Gaussovou funkcí.

Jelikož bylo rozhodnuto, že hledání podobných segmentů bude následně prováděno pouze nad sakádama, poněvadž v místech fixací by nalezené segmenty příliš nevyhovovaly o komplexním principu čtení mapy, byly rovněž vykresleny tyto sakády nad danou mapou.

Poslední součástí skriptu bylo vypočítání polárních souřadnic a euklidovské vzdálenosti dvou sousedních bodů, každému bodu definujícímu polohu sakád (důvod

bude uveden níže). Tyto hodnoty byly společně s X a Y souřadnicemi bodů zapsány do samostatných textových souborů.

Pro názornou ukázkou toho, kam všude a jak dlouho se uživatel na předkládanou mapu díval byl vytvořen další skript, který generuje videozáznam s těmito informacemi. Zde bylo použito obdobné metodiky jako při vykreslování HeatMap.

Pro implementaci Voting Experts algoritmu byl použit program VE Console, představený v předešlé kapitole. Aby v něm bylo docíleno optimální nastavení parametrů, byly v grafickém vektorovém editoru Inkscape vytvořeny vzorové obrázky, na kterých bylo nastavení těchto parametrů testováno.

Jelikož zmíněný program je primárně vytvořen pro vstup dat v jedno-číselném, případně textovém formátu bylo potřeba vymyslet, jak do něj nahrát data ve dvou-číselném zápise (poloha oka X a Y). Prvotním pokusem bylo využít Mortonova rozkladu (*Z-order*), což se ovšem nakonec ukázalo jako ne příliš vhodná varianta. Proto bylo vytvořeno pět dříve zmíněných textových souborů - *poloha X*, *poloha Y*, *poloha  $\varphi$* , *poloha r* a *euklidovská vzdálenost mezi sousedními body*. Tyto soubory následně vstupovaly do programu samostatně a výstupem bylo opět pět souborů, kde každý obsahoval počty hlasů od hlasujících expertů.

Pro konečné rozhodnutí, kde budou sakády rozděleny na kratší segmenty byl vytvořen třetí skript. Ten sečetl hlasy ze všech pěti textových souborů a na základě zadané prahové hodnoty provedl řezy nad křivkou. Výsledné segmenty graficky vizualizoval a uložil do textového souboru.

Aby bylo možné výsledné segmenty mezi sebou vzájemně porovnat, byla vymyšlena metodika, kdy se nad každým segmentem vypočítalo těžiště a se středem v tomto těžišti se vytvořil šestnácti-dílný prostor. Následně bylo vypočteno kolik bodů daného segmentu spadá do každého dílu takto vzniklého prostoru, čímž byly vytvořeny tzv. histogramy četností bodů, příslušných k danému rozdělení prostoru. Na závěr byly dle těchto hodnot histogramů určeny podobnosti jednotlivých segmentů.

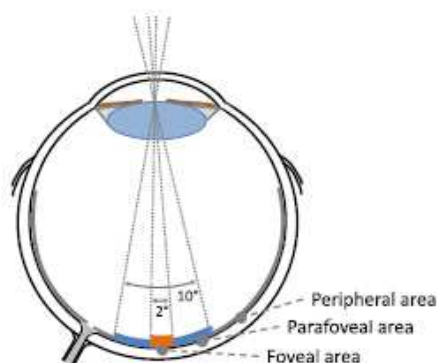
### 3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Stěžejním cílem a hlavní problematikou této diplomové práce je pomocí algoritmů pracujících s časovými řadami nalézt podobné segmenty v datech naměřených technikou eye-tracking. Proto bude v této kapitole vysvětlen základní princip eye-tracking systému a následně budou představeny algoritmy pracující s časovými řadami. Konkrétně Voting Experts algoritmus, Borcení časovou osou a jejich vzájemná kombinace, jež budou využity k nalezení zmíněných podobných segmentů.

#### 3.1 Eye - tracking

Technologie eye-tracking je poměrně novou moderní metodou založenou na principu snímání pohybu lidských očí při vnímání vizuálního vjemu a následného vyhodnocování. Zařízení, jež je schopné tyto pohyby sledovat se označuje jako eye-tracker.

Lidské oko může vnímat pouze omezenou část vizuálního světa v daný čas, na jednom místě. Během fixace, což je okamžik, kdy se oči neustále dívají na jedno místo v pozorované scéně, obě oči společně poskytují zhruba elipsovitý obraz prostoru, který je přibližně 200° zorného pole na šířku a 130° na výšku. Ovšem z důvodu nestejně struktury sítnice oka nejsou všechna místa tohoto pohledu vnímány se stejnou ostrostití. V centrální oblasti sítnice tzv. Fovea je malá oblast sítnice kolem 2° až 3° zorného úhlu, kde dochází k maximální zrakové ostrostiti. Tato oblast se nachází ve středu sítnice a pokrývá oblast kolem fixačního bodu oka. S rostoucí vzdáleností od foveální oblasti se snižuje zraková ostrost. Na 50% při úhlu 5° od fovey a při 40° je ostrost pouze 10%. Fovea je dále ohraničena parafoveální a následně periferní oblastí. (upraveno dle Biedert, 2010). (Obr. 2) Vjem obrazu touto větší částí našeho zorného pole, který je nazýván periferním viděním, je lépe přizpůsoben slabému světlu, kde registruje pohyby a kontrasty mezi barvami a tvary (Obr. 3). Tento vjem rozmazaný a málo pestrý (Tobii Technology, 2010).



Obr. 2: Fovea, parafovea, periferní oblast  
(zdroj: Biedert, 2010).



Obr. 3: Zorné pole lidského oka  
(zdroj: Tobii Technology, 2010).

Pohyby očí se dělí na dva základní typy - fixace a sakády. Jak již bylo výše uvedeno, k fixaci dochází pokud je sítnice stabilizována nad jedno určité místo pozorovaného vjemu. Naopak sakády jsou velmi rychlé pohyby očí používané k přemístění fovey na novou oblast pozorovaného vjemu (Duchowski, 2007). Dochází zde tedy k velmi rychlým pohybům očí z místa na místo. Jak uvádí (Biedert, 2010) kromě fixací a sakád jsou definovány ještě daleko menší pohyby očí o vysoké frekvenci, kterých si lidé ovšem nejsou vědomi. Jejich funkcí je zabránit efektu saturace receptorů na sítnici, který by vedl ke zeslabení vnímání. Těmito pohyby jsou myšleny tremory, drifty a mikrosakády.

Na základě statistické analýzy fixací, sakád, jejich vzájemného poměru a dalších charakteristik je možné rozpoznat určité rysy respondentova chování (Popelka, 2012). Dochází-li například k situaci, kdy uživatel hodně kmitá očima nad pozorovaným stimulem z místa na místo, což je charakterizováno vysokým výskytem sakád v naměřených datech, poukazuje tento jev na nízkou efektivitu při vyhledávání. Což ovšem v závislosti na zkoumaném jevu rovněž může odborníka informovat o přístupu respondenta, který požívá při kognici, nebo o nevhodnosti kompozice předkládaného stimulu.

Rovněž dle zjištěných fixací lze charakterizovat respondentův přístup k vnímanému obrazu. Dle Brodersena a kol. (2001) delší fixace naznačuje oblasti s obtížnou informací, nebo nedostatek relevantních objektů. Byl zjištěn silný vztah mezi počtem fixací a hodnotou informací na dané oblasti pozorovaného obrazu.

### **3.1.1 Metody měření**

Jak uvádí Duchowski (2007) s odvoláním na Younga a Sheena (1975) existují dva typy technik pozorování lidských očí. Metody měření polohy oka vzhledem k hlavě a metody určující polohu oka v prostoru, tzv. Point of Regard.

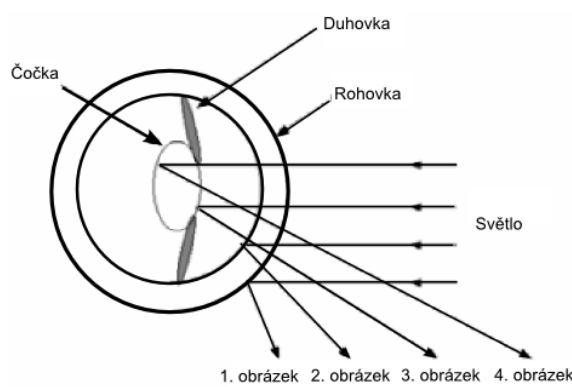
Podle Duchowského (2007) lze metody sledování pohybu očí rozdělit do čtyř hlavních kategorií:

- elektrookulografie (EOG, využití elektrod umístěných v okolí oka),
- metody využívající kontaktní čočky,
- bezkontaktní metody.

EOG je založena na principu měření rozdílu kožního elektrického potenciálu pomocí elektrod rozmístěných kolem očí, přičemž pohyb očí je měřen vůči pozici hlavy. Tato metoda byla nejvíce využívána zhruba před 40 lety.

Využití speciálních kontaktních čoček je jednou z nejpřesnějších metod měření pohybu lidského oka. Na kontaktní čočku je připojeno nejrůznější mechanické, nebo optické zařízení a čočka je následně nasazena přímo do oka. Je zapotřebí aby kontaktní čočky byly větších rozměrů, přesahujících rohovku a zorničku. I v tomto případě je měřena pozice oka vzhledem k hlavě. Ačkoliv se jedná o nejpřesnější metodu, je zároveň také tou nejvíce nepohodlnou.

Bezkontaktní metody v současnosti patří mezi nejvíce používané. Tento přístup spočívá v měření zornice a korneálního odrazu. Přičemž nejvyššího prostorového a časového rozlišení může být dosaženo použitím tzv. dual-Purkyňových eye-trackeru (Gienko a Levin, 2005 s odkazem na Cornsweet, 1973). Tyto eye-trackery vyzařují směrem k oku infračervené záření a sledují jeho odrazy od vnější strany rohovky a zadní strany čočky. Tyto odrazy se nazývají 1. a 4. Purkyňův obrázek (Obr. 4) Tyto dva obrázky se pohybují ve stejné vzdálenosti a stejném směru oka. V průběhu rotace oka se odděleně mezi dvěma obrazy úměrně změní sinus úhlu natočení. Tímto lze získat úhlové pozice očí z relativních pozic dvou obrazů (KU Leuven, 2011).



Obr. 4: Čtyři Purkyňovy obrázky jsou odrazem dopadajícího světla na čočku a rohovku.

Volba vhodného eye-trackeru zcela závisí na potřebách řešené problematiky. Přístroje mohou být statické, nebo mobilní. V prvním případě je zařízení umístěno na stole, v tom druhém je respondentem nošen na hlavě. Důležitým parametrem přístroje je časové rozlišení, které je udáváno v hertzech (Hz).

### 3.1.2 Možnosti vizualizace

Pokud nejsou při exportu dat nastavena jiná specifika, jsou výstupem z eye-trackingu tzv. raw data. Jedná se o kontinuální záznam všech naměřených hodnot polohy oka v dané časové periodě. Tyto data je následně nutno dle určitého přístupu klasifikovat na fixace a sáky, které již mohou být následně analyzovány, nebo vhodně vizualizovány.

Dle různých přístupů existuje k vizualizaci dat z eye-trackingu několik hlavních metod. Zmíňme zde například HeatMapy, Focus Mapy, GazePlot, GazeReplay, Space-Time-Cube, Area of Interest atd. (Holmqvist a kol., 2011), (Li a kol, 2010), (Consumer Insights Group, 2011) Výběr metody se odvíjí od požadované představy vizualizace vhodné k dané problematice.

#### HeatMap

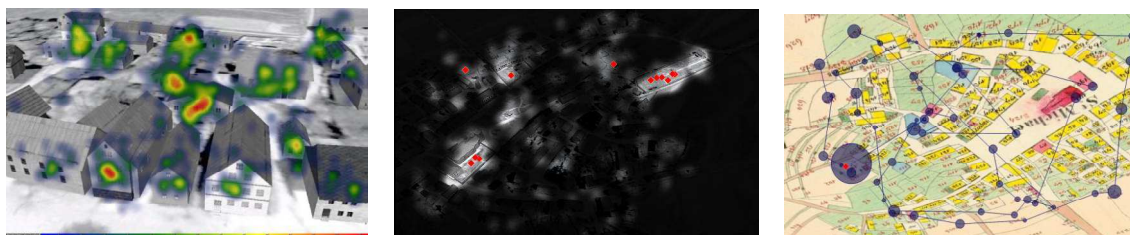
Při tomto typu vizualizace jsou nad sledovaným obrazem vytvořeny barevné "skvrny", jež zobrazují, na které části obrazu se respondent více soustředil. V centrální, tedy

nejvíce koncentrované oblasti je červená, žlutou je zobrazen přechod na střední množství pozornosti a konečné blednutí k zelené, nebo modré indikuje místa s nejmenším zájmem (Consumer Insights Group, 2011). Jsou tedy vhodné především k rychlému a názornému přehledu o oblastech, kde uživatel soustřeďuje svou zrak více a kde naopak méně.

### **GazePlot / GazeReplay**

Tato metoda bývá rovněž někdy nazývaná jako ScanPath. Zobrazuje trajektorii sakád, spojující fixace, které jsou znázorněny kruhy o různých poloměrech. Čím větší poloměr kruhu, tím docházelo k delší době fixace. Nevýhodou ovšem bývá dojde-li je zpracovávání většího množství dat, kdy dochází k překryvu jednotlivých fixací a záznam se tak stává nepřehledným.

Jistou alternativou je metoda GazeReplay, kde jsou rovněž vykreslovány trajektorie sakád a fixace s proměnlivou velikostí kruhu v závislosti na délce fixace. Je ale navíc obohacena o časovou složku. Lze tedy přehrát videozáznam zobrazující pohybu očí uživatele nad daným obrazem. Záznam je v tomto případě přehledný, avšak jeho pozdější analýza je značně komplikovaná.



Obr. 5: Ukázka HeatMap, Focus Map a GazePlot  
(zdroj: Dědková, 2012).

### **Areas of Interest**

Pozorovaný obraz je rozdělen do několika oblastí zájmů a následně je možno analyzovat a počítat kolik času strávil uživatel v dané oblasti, v jakém pořadí navštívil oblasti, nebo kolik pozornosti se dostalo jednotlivé oblasti ve srovnání ostatními.

## **3.2 Voting Experts algoritmus**

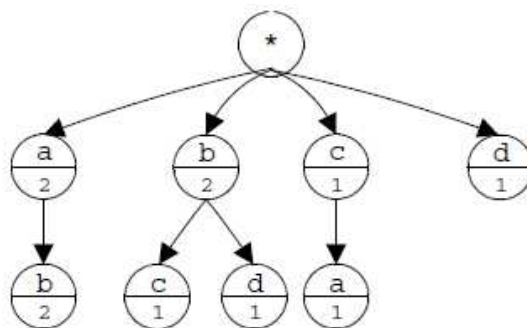
Voting Experts je doménově nezávislý algoritmus (učící se bez učitele) určený pro segmentaci kategoriálních časových řad do smysluplných episod. Poprvé byl představen dvojicí autorů Cohen a Adams (2001). Základní myšlenka algoritmu je založena na jednoduché hypotéze, a to že typické vzory nalezené v datové kolekci jsou obvykle doprovázeny dvěma statistickými indikátory: nízkou interní entropií uvnitř těchto vzorů a vysokou entropií na jejich okrajích (Cohen a kol., 2006). Jak dále Cohen a Adams (2001) uvádějí, základní Voting Expertes algoritmus zahrnuje experty hlasující podle okrajové entropie (*entropy expert*) a frekvence (*frequency expert*). Je ovšem velmi snadno

rozšiřitelný o experty hlasující dle dalších charakteristik. Zde jsou uvedeny jeho tři základní kroky:

- nejprve se sestaví strom všech  $n$ -gramů a vypočítají se statistiky pro každý jeho uzel (okrajová entropie a frekvence). Ty jsou standardizovány pro hodnoty stejné délky těchto  $n$ -gramů,
- následně se posouvá plovoucí okno o velikosti  $n$  skrze všechny vstupní hodnoty a experti hlasují. Každý z expertů má svůj vlastní pohled na aktuální kontext (aktuální obsah plovoucího okna) a hlasuje pro lokalitu, která je dle jeho úsudku nejvhodnější pro rozdělení dlouhé sekvence na kratší,
- nakonec jsou na základě sečtených hlasů určeny lokality, kde dojde k rozdělení na kratší sekvence.

### 3.2.1 Sestavení stromu $n$ -gramů a standardizace výsledků

Dle Cohena a kol. (2006), strom  $n$ -gramů o hloubce  $n + 1$  představuje podsekvence o délce  $n$  vstupní sekvence. Například sekvence  $a b c a b d$  s hloubkou stromu 3 je znázorněna na (Obr. 6). Každý  $n$ -gram o délce 2 a méně je v sekvenci  $a b c b d$  reprezentován jako uzel stromu. Čísla ve spodní polovině kružnic nesou informaci o četnosti výskytu každé podsekvence. Například posloupnost  $ab$  se vyskytuje dvakrát a za každým  $a$  následuje  $b$ . Podsekvence  $b c a b d$  se vyskytuje pouze jednou, takže frekvence výskytu podsekvence  $b$  je 2.



Obr. 6: Strom  $n$ -gramů o délce 2 pro sekvenci  $a b c a b d$   
(zdroj: Cohen a kol., 2006).

Okrajová entropie  $n$ -gramu je entropií rozdělení tokenů (znaků), které mohou následně rozšířit  $n$ -gram. Entropie rozdělení náhodné proměnné  $x$  je vypočtena z  $n$ -stromu dle vztahu:

$$-\sum p(x) \log p(x),$$

kde  $p$  je pravděpodobnost výskytu proměnné  $x$  (Cohen a Adams, 2001). Například pro uzel  $a$  má entropii rovnou nule, jelikož má pouze jednoho potomka, zatímco entropie uzlu



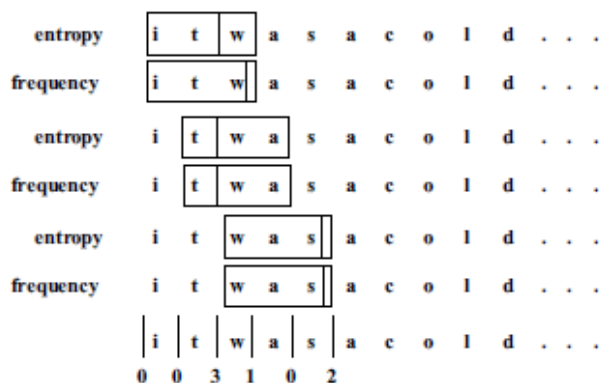
$b$  je 1, protože má potomky dva. Tedy, výhradně prvních  $n$  pater stromu  $n$ -gramů o hloubce  $n + 1$  může dostat hlasy.

V algoritmu existuje vzájemný vztah mezi délkou podsekvence a četností jejího výskytu (frekvence): Krátké vzory se vyskytují častěji, než dlouhé. Tento vztah byl ověřen na prvních 10 000 znacích románu 1984 od Orwella (Cohen a Adams, 2001). Mezi těmito znaky byly odstraněny mezery mezi jednotlivými slovy. Výsledkem bylo, že 64 ze 100 nejfrekventovanějších vzorů jsou vzory o délce 2, 23 o délce 3 a tak dále. Voting Experts neporovnává frekvenci a okrajovou entropii u  $n$ -gramů odlišných délek. Ale ve všech případech bude porovnávat, jak neobvyklé jsou tyto entropie a frekvence relativně k dalším  $n$ -gramům o stejné délce. Pro názornost, byla spočtena frekvence u vzorů  $a$  a  $an$  opět u prvních 10 000 znaků románu 1984: "a" se vyskytovalo 743 krát, "an" 124 krát, avšak "a" se vyskytovalo jen trochu častěji než-li ostatní jedno-znakové  $n$ -gramy. Zatímco "an" se vyskytovalo mnohem vícekrát než ostatní dvou-znakové  $n$ -gramy. V tomto smyslu by "a" bylo častější, než "an" které by bylo neobvyklé. Přestože je tedy "a" mnohem častější, nežli "an" je mnohem méně relativně neobvyklé k ostatním  $n$ -gramům o stejné délce.

K definování pojmu neobvyklý výskyt je tedy třeba standardizovat frekvenci a okrajovou entropii  $n$ -gramů. Pro standardizaci daného vzoru je od vzoru odečtena průměrná hodnota výskytu a vše je následně poděleno standardní odchylkou:

$$z_i = (x_i - \bar{x})/s.$$

Kde  $z_i$  tedy znamená vzdálenost hodnoty výskytu daného vzoru od průměrné hodnoty výskytu vzoru stejné délky. Ve výše uvedeném příkladu je standardizovaná hodnota frekvence pro vzor "a" 1,3 zatímco pro "an" je 3,7. Jinými slovy lze říct, že frekvence "an" je o 3,7 standardní odchylky nad průměrnou hodnotou výskytu sekvencí o stejné délce. Standardizace okrajové entropie proběhla stejným způsobem.



Obr. 7: Příklad výpočtu voting experts algoritmu  
(zdroj: Cohen a kol., 2006).

### 3.2.2 Hlasování expertů

Jak uvádí Cohen a kol. (2006), proces přiřazování hlasů je znázorněn na (Obr. 7). Plovoucí okno o délce 3 prochází skrz celou sekvenci, v tomto případě *itwascold*. Nejdříve okno pokrývá *itw* a nad těmito prvky experti hlasující podle okrajové entropie a frekvence rozhodují, kde by mohli co nejlépe sekvenci rozdělit. Entropy expert upřednostňuje hranici mezi *t* a *w*, zatímco frequency expert mezi *w* a jakýmkoliv dalším znakem v pořadí (vysvětleno níže). Následně se plovoucí okno posune o jednu pozici doprava a celý proces se opakuje. Tentokrát oba experti rozhodnou umístit hranici mezi *t* a *w*. Okno se opět posune o pozici doprava a experti zvolí umístění hranice za *s*, které je posledním prvkem v okně.

Jak je patrné, tak každé potenciální umístění hranice (např. mezi *t* a *w*) je posuzováno plovoucím oknem o velikosti  $n$ ,  $n$  krát. Pokaždé ke ale posuzováno v mírně odlišném kontextu. Nejdříve experti posuzují hranici mezi *w* a *a* v okně *itw* a nakonec v okně *was*. Tímto způsobem každá hranice se účastní  $2n$  hlasování. Hranice *wa* dostane jeden hlas, hranice *tw* tři hlasy a *sa* hranice dva hlasy.

Experti využívají různých metod pro určení hranic v sekvenci. Nyní je podrobněji popsáno rozhodování dvou hlavních expertů votig experts algoritmu, entropy a frequency experta. Zaměříme se tedy nejprve na entropy experta, který se rozhoduje nad oknem *itw*. Nejdříve jsou spočítány  $n$ -gramy: *i*, *it* a *itw*, které jsou následně standardizovány. Entropy expert poté hlasuje pro pozice tvořené  $n$ -gramem s nejvyšší okrajovou entropií. Konkrétně opět u románu 1984 jsou standardizované okrajové entropie pro *i* 0,2, pro *it* 1,39 a pro *itw* 0,02. Entropy expert tedy rozhodl o rozdělení sekvence za  $n$ -gramem *it*.

Frequency expert umístí hranice tak, aby se maximalizoval součet standardizovaných frekvencí zleva i zprava od hranice. Zaměříme-li se opět na první okno *itw*, tak pokud je hranice vložena za *i*, tak standardizovaná frekvence *i* a *tw* je 1,73. Pokud je hranice vložena za *it*, potom je standardizovaná frekvence *it* a *w* 2,9. A pokud je hranice vložena za *itw*, potom algoritmus počítá pouze standardizovanou frekvenci *itw* a ta je 4,0. Proto frequency expert rozhodne sekvenci rozdělit za *itw*.

### 3.2.3 Vytvoření nových segmentů

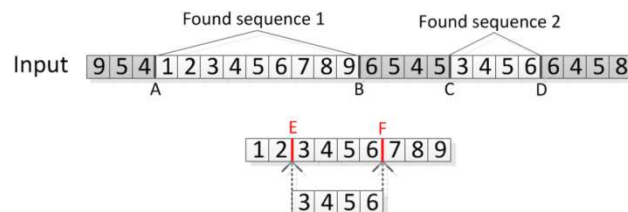
Když experti odhlasují a na základě jejich hlasů jsou vytvořeny potenciální hranice k rozsekání původní sekvence, musí algoritmus tyto hranice vyhodnotit a rozhodnout kde sekvenci rozdělí. Metoda je známa jako *zero crossing*: Pokud má potenciální hranice lokální maximum počtu hlasů, v tomto maximu sekvenci rozdělí. V příkladu uvedeném výše, zmíněné pravidlo rozdělí sekvenci *itwascold* za *it* a *was*. Problém se *zero crossing* nastane, pokud je jeden hlas lokálním maximem delší sekvence hlasů (na okolních pozicích jsou nulové hlasy), což způsobuje daleko frekventovanější rozdělení sekvence. Empiricky bylo zjištěno, že čím víc hlasů pozice dostane, tím větší je pravděpodobnost správného rozdělení. Tento přístup rozdělí sekvenci podle jednoho hlasu se stejnou jistotou, jako když ji rozděljuje podle deseti hlasů, pokud jsou obě čísla lokální maxima.

Aby předešel této tendenci rozdělit sekvenci dle malého počtu hlasů, je stanovena minimální hodnota, kterou musí počet hlasů přesáhnout, aby došlo k rozdělení sekvence. (upraveno dle Cohen a kol., 2006).

### 3.2.4 Zvýšení výkonu Voting Experts algoritmu

Existuje mnoho způsobů, kterými lze zdokonalit základní myšlenku Voting Experts algoritmu a tím zvýšit jeho účinnost. Prvním řešením je přidání vlastního experta do hlasovacího procesu, čímž lze získat zcela nový pohled na vstup. Dalším možným řešením je využít metodu založenou na opakovaném hierarchickém dělení vstupních hodnot (Miller a Stoytchev, 2008).

Jednou z nejjednodušších cest, kterou lze zvýšit výkon segmentace, je dvoucestný průchod plovoucího okna nad vstupními daty. Což znamená nejprve využít základní verzi Voting Experts algoritmu a následně ji použít na obráceném vstupu (Kocyan a kol., 2011 s odkazem na Hewlett a Cohen, 2009). Základní myšlenkou je, že pokud jsou ve stupních datech (Obr. 8) nalezeny díky oboustrannému přístupu dvě sekvence s vysoce přesnými řezy (sekvence 1: A - B, sekvence 2: C - D) a sekvence 2 je podsekvencí 1, pak vzniknou nové řezy E a F, které se nacházejí na okrajích sekvence 2, zarovnané v sekvenci 1.



Obr. 8: Hledání velmi přesných řezů.

(zdroj: Kocyan a kol., 2012)

Otočením vstupu a jeho opětovnou segmentací je vytvořena nová odlišná množina potenciálních řezů. Pokud dojde k průniku těchto dvou množin řezů dosáhne se velmi vysoké přesnosti, avšak také ke ztrátě úplnosti (Kocyan a kol., 2011).

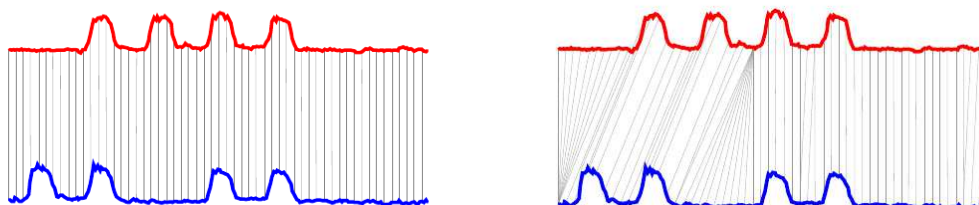
### 3.3 Borcení časové osy

Borcení časové osy (*Dynamic Time Warping, DTW*) je metodou sloužící k nalezení optimálního zarovnání dvou (časově závislých) sekvencí dle předem určených pravidel. V podstatě se jedná o nelineární spárování těchto sekvencí tak, aby si co nejvíce odpovídaly.

Při porovnání dvou sekvencí podobného tvaru, ale odlišného uspořádání z hlediska časové složky, je použití Euklidovské vzdálenosti pro tento typ signálu nevhodné. Jedná se totiž o lineární transformaci, která porovnává protější prvky obou sekvencí, tedy prvky

na stejných indexech. Čímž při pouze drobném posunu, který nemusí mít v systému žádný význam, celý výpočet podobnosti překazí. (Obr. 9, vlevo)

Oproti tomu metoda DTW využívá nelineární časové normalizace, u které jsou rozdíly v časové ose modelovány časově nelineární "bortivou" funkcí, jak již bylo zmíněno s předem určenými pravidly. Rozdílnost v čase mezi dvěma sekvencemi je tedy eliminována borcením jedné z časových os tak, aby došlo k nejlepší možné shodě s druhou sekvencí. (Obr. 9, vpravo)



Obr. 9: Rozdíl mezi užitím Euklidovské vzdálenosti (vlevo) a metody DTW (vpravo).

(zdroj: <http://www.izbicki.me/blog>, 2011)

Tato metoda byla poprvé použita k porovnání dvou odlišných hlasů při jejich automatickém porovnávání. V současnosti je využívána především v oblastech, kde dochází ke zpracovávání biologických signálů.

### 3.3.1 Princip DTW

Dle Müllera (2007) a Senina (2008) je cílem metody DTW tedy porovnat dvě časově závislé sekvence  $X$  a  $Y$  definované:

$$X = (x_1, x_2, \dots, x_N) \text{ délky } N \in \mathbf{N} \text{ a } Y = (y_1, y_2, \dots, y_M) \text{ délky } M \in \mathbf{N}.$$

Tyto sekvence mohou být diskrétní časové řady, nebo obecné sekvence vlastností vzorkovaných v ekvidistantních časových intervalech. Pokud je tedy prostor vlastností označen jako  $F$ , pak jsou sekvence  $X$  a  $Y$  definovány jako:

$$x_n, y_m \in F \text{ pro } n \in [1:N] \text{ a } m \in [1:M].$$

K vyjádření vzdálenosti dvou různých prvků těchto sekvencí  $x, y \in F$ , je nutné definovat metriku  $c$  (nazývanou *local cost measure*, popř. *local distance measure*). Ta přiřadí zmíněným dílčím prvkům obecně nízkou hodnotu, jsou-li si podobné, nebo naopak vysokou, pokud si podobné nejsou.

$$c: F \times F \rightarrow \mathbf{R}_{\geq 0}$$

Tato vzdálenost se vypočte pro každou dvojici  $x, y$  obou sekvencí a sestaví se do tzv. matice vzdáleností (*cost matrix*)  $C \in \mathbf{R}^{N \times M}$ , která je definována jako  $C(n, m) = c(x_n, y_m)$ . V tomto okamžiku je vše připraveno k nalezení spárování dvou sekvencí.

Ke konečnému spárování je nakonec ještě potřeba nalézt takovou deformační cestu, která je provedena za nejnižší cenu. Samotná deformační cesta (*warping path*) je posloupnost  $p = (p_1, p_2, \dots, p_L)$ , kde  $p_l = (n_l, m_l) \in [1: N] \times [1: M]$  a  $l \in [1: L]$ , splňující následující tři podmínky:

- 1) Hraniční podmínka:  $p_1 = (1, 1)$  a  $p_L = (N, M)$ .
- 2) Podmínka monotónnosti:  $n_1 \leq n_2 \leq \dots \leq n_L$  a  $m_1 \leq m_2 \leq \dots \leq m_L$ .
- 3) Podmínka velikosti kroku:  $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$  pro  $l \in [1: L - 1]$ .

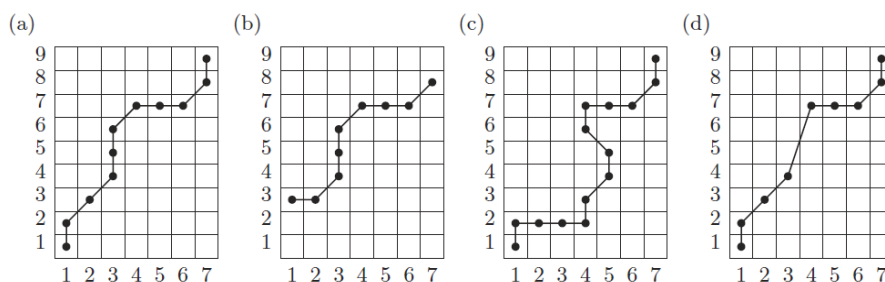
Hraniční podmínka definuje, že první a poslední prvky sekvence  $X$  a  $Y$  budou spárovány na sebe navzájem, což zajistí, že budou zarovnáány celé sekvence a žádný z krajních bodu nebude vynechán. Podmínka monotónnosti zaručuje stoupající trend deformační cesty a poslední podmínka velikosti kroku vyjadřuje jakousi kontinuitu pohybu prvků v rámci matice, tak aby žádný prvek nebyl vynechán, nebo duplikován (Obr. 10).

Celková cena (*total cost*)  $c_p(X, Y)$  nalezené deformační cesty mezi sekvencemi  $X$  a  $Y$  s ohledem na metriku  $c$  ke definována jako:

$$c_p(X, Y) = \sum_{l=1}^L c(x_{n_l}, y_{m_l}).$$

Avšak v závislosti na délce sekvencí existuje vyhovujících deformačních cest mezi  $X$  a  $Y$  různě mnoho. Proto je tou nevhodnější cestou  $p^*$  zvolena ta, která prokáže minimální náklady ze všech deformačních cest. Hledání té nevhodnější je definováno jako:

$$DTW(X, Y) = c_{p^*}(X, Y) = \min\{c_p(X, Y) \mid \text{kde } p \text{ je } (N, M) - \text{deformační cesta}\}.$$



Obr. 10: Nalezení deformační cesty pro sekvenci  $X$  o délce 9 a sekvenci  $Y$  o délce 7. (a) Správné nalezení cesty. (b) Hraniční podmínka nebyla dodržena. (c) Podmínka monotónnosti nebyla dodržena. (d) Podmínka velikosti kroku nebyla dodržena.

(zdroj: Müller 2007)

Takovýto přístup je ovšem dosti zdlouhavý. Proto je výpočet zefektivněn sestavením tzv. akumulované matice vzdáleností (*accumulated cost matrix*)  $D$ , jenž je tvořena prvky s nejnižší možnou cenou, za kterou se lze dostat na aktuální prvek z bodu  $(x_1, x_2)$ . Kde hodnoty akumulované matice vzdáleností jsou:

$$D(n, 1) = \sum_{k=1}^n c(x_k, y_1) \quad \text{pro } n \in [1: N]$$

$$D(1, n) = \sum_{k=1}^n c(x_1, y_k) \quad \text{pro } m \in [1: M]$$

$$D(n, m) = \min \{D(n-1, m-1), D(n-1, m), D(n, m-1) + c(x_n, y_m)\},$$

$$\text{pro } 1 < n \leq N \text{ a } 1 < m \leq M$$

Optimální cesta  $p^* = (p_1, \dots, p_L)$  je následně vypočtena v obráceném pořadí, kde začíná v bodě  $p_L = (N, M)$ . Předpokládejme tedy, že  $p_l = (n, m)$  je již spočítáno. V případě, že je dosaženo  $(n, m) = (1, 1)$ , a při tom  $l = 1$  je výpočet ukončen. Za podmíněk, že:

$$p_{l-1} = \begin{cases} (1, m-1), \text{ pokud } n = 1 \\ (n-1, 1), \text{ pokud } m = 1 \\ \min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\}, \text{ v ostatních případech.} \end{cases}$$

### 3.3.2 Využití DTW při hledání podsekvencí

Kromě výše uvedeného principu nalezení spárování dvou libovolných sekvencí, umí metoda DTW na základě menší úpravy výpočtu akumulované matice vzdáleností pracovat s kratšími podsekvencemi, čímž se stala ještě daleko populárnější. Je tedy schopna nalézt nejvhodnější umístění kratší podsekvence  $X$  v druhé, delší, sekvenci  $Y$ .

Jak uvádí Müller (2007) lze si tento problém demonstrovat na příkladu, kdy si delší sekvenci  $Y$  představíme jako určitou databázi a kratší sekvenci  $X$  jako dotaz, pro který chceme vyhledat co nejpřesnější odpověď. Problematika nalezení optimálního spárování podsekvence s využitím upravené metody DTW bude na základě Müllera (2007) popsána níže.

Nechť máme sekvence  $X = (x_1, x_2, \dots, x_N)$  a  $Y = (y_1, y_2, \dots, y_M)$ , kde se předpokládá, že  $M$  je mnohem delší, než-li  $N$ . Cílem je nalezení takové podsekvence  $Y(a^*: b^*) = (y_{a^*}, y_{a^*+1}, \dots, y_{b^*})$ , kde  $1 \leq a^* \leq b^* \leq M$ , která minimalizuje cenu k  $X$ , skrz všechny možné podsekvence databáze  $Y$ , tedy:

$$(a^*: b^*) = \min (DTW((X, Y(a:b))))$$

Hodnoty  $a^*$  a  $b^*$  jsou počátečním a konečným indexem nejlepšího spárování dotazu  $X$  v databázi  $Y$ .

Základní myšlenkou při hledání podsekvencí je neopomíjet prvky vyskytující se před začátkem a za koncem dotazovaného  $X$ . Ale umožnit tedy posun dotazu  $X$  oběma směry v rámci databáze  $Y$ . Cože je zajištěno pomocí změny definice akumulované matice vzdáleností z tvaru:

$$D(1, m) = \sum_{k=1}^m c(x_1, y_k) \text{ pro } m \in [1: M]$$

na tvar:

$$D(1, m) = c(x_1, y_m) \text{ pro } m \in [1: M]$$

Pokud je akumulovaná matice upravena, začne se hledat deformační cesta podsekvencí. Hledání začíná na koncovém indexu  $b^*$ , který lze určit ze zmíněné akumulované matice:

$$b^* = \min_{b \in [1:m]} D(N, b)$$

Optimální deformační cesta  $p^* = (p_1, p_2, \dots, p_L)$  je následně počítána z bodu  $p_L = (N, b^*)$  a končí v bodě  $p_1 = (a^*, 1)$ , pro nějaké  $l \in [1: L]$

Tímto postupem je tedy nalezena optimální deformační cesta. K výpočtu se ještě přidává definice prahové hodnoty, která udává minimální cenu, kterou je třeba překročit.

### 3.4 Voting Experts s DTW post-procesem

Kombinací dvou výše uvedených metod, tedy Voting Experts algoritmu a Borcení časovou osou se zabývá ve svém příspěvku Kocyan a kol. (2012). Autoři si berou za cíl obohatit Voting Experts jednak dvoucestným průchodem plovoucího okna nad vstupem, čímž se dostanou vysoce přesné řezy, které budou výchozím bodem pro nalezení typických vzorů nacházejících se ve vstupních datech. A také o post-processing využívající metodu DTW. Jelikož je v následující práci využíván tento autorův přístup a je tudíž stěžejní součástí segmentace dat z eye-tracking systému, bude mu zde věnován větší prostor.

Princip dvoucestného Voting Experts algoritmu byl nastíněn již v kapitole 3.2.4. Pojdme si tedy nyní představit samotný DTW post-processing. Dle autorů je shrnut do následujících bodů:

- 1) Nejdříve se provede hlasování metodou Voting Experts v jednom a následně i v opačném směru s vysokým prahem tak, aby se získaly vysoce přesné řezy.
- 2) Unikátní nalezené sekvence v kroku jedna označme jako  $m$ .

- 3) Sestaví se matice vzdáleností  $m \times m$ .
- 4) Pro všechny dvojice této matice, kde délka *sekvence 1* je větší než *sekvence 2*:
  - A) Pomocí metody DTW upravené pro hledání podsekvencí se nalezne optimální spárování kratší *sekvence 2* do delší *sekvence 1*.
  - B) V případě, že cena za spárování nepřesáhne stanovený práh, uloží si *sekvence 1* kratší *sekvenci 2* do interního seznamu podobných sekvencí. Takto každá sekvence získá seznam velmi podobných kratších sekvencí.
  - C) Každá kratší sekvence svým spárováním s delší sekvencí ukazuje, kde by podle ní měla být delší sekvence rozdělena. Jelikož je podobných kratších sekvencí zpravidla více, je také ukázáno na více míst, přičemž mnohdy se tato místa duplikují. Z tohoto důvodu jsou hlasy sbírány do interního pole hlasů.
  - D) Jakmile jsou všechny hlasy uděleny, provede se v interním poli hlasů hledání lokálních maxim, překračujících určitý práh. Tato místa jsou navrhnutá k rozdělení. Těmto lokalitám jsou uděleny hlasy zcela stejným způsobem jako od tradičních dvou expertů (frequency a entropy).
- 5) Tyto jsou hlasy jsou sečteny s hlasy od dvou tradičních expertů. Následně se kontrolují lokální maxima překonávající zvolený práh. V těchto místech jsou provedeny řezy a vzniknou nové segmenty.
- 6) Algoritmus končí, nebo může opětovně pokračovat bodem 2) a dlouhé sekvence dále rozmělnit.

Pro vylepšení celého post-procesu bylo testováno několik variant úprav. Existuje mnoho dalších proměnných a přístupů, které mohou vstupovat do algoritmu. Konkrétní popis a výsledné vyhodnocení uvádějí autoři ve svém článku (Kocyan a kol., 2012). Trošku podrobněji zde zmiňme již pouze nastavení prahové hodnoty, kterou bylo třeba nastavit v pozdější práci. Zde je nastíněna metoda autorů algoritmu, ačkoliv v samotné práci byla prahová hodnota zvolena odlišným přístupem (kap. 5.3).

V bodě 4B této kapitoly bylo uvedeno, že se pracuje pouze se sekvencemi, které překročí určitý práh. Testováním, které provedli autoři se ukázalo, že pokud nedojde k omezení délky podsekvencí na rozumný relativní průměr vůči délce dlouhé sekvence, dojde k rozbití vstupu na velké množství drobných sekvencí. Což sice zvýší úplnost, ale rapidně zhorší přesnost.

Z tohoto důvodu se zavedly dva parametry - dělitel a offset. Aby kratší sekvence nebyla z hlasování vyloučena, musí mít tedy minimální délku :

$$\|shorter\ sequence\| > \frac{\|longer\ sequence\|}{divisor} + offset$$

Obr. 11: Nastavení prahové hodnoty.

(zdroj: Kocyan a kol., 2012)



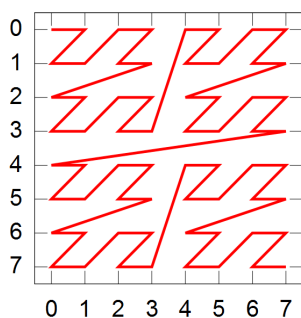
Veškeré testování nastavování proměnných bylo z důvodu dobré zpětné kontroly, provedeno na textu knihy Jaroslava Haška, Osudy dobrého vojáka Švejka v české i anglické podobě, tedy datech typu kategoriálních časových řad.

## 4 PŘEDZPRACOVÁNÍ DAT

Jelikož bylo pro pozdější implementaci výše představených algoritmů využito programu vytvořeného Ing. Tomášem Kocyanem (více v kapitole 5), který vyžaduje vstupní data v jedno-číselném formátu, bylo potřeba do takovéto podoby převést dvojrozměrná data z eye-tracking systému. K tomuto převodu byl nejprve použit Mortonův rozklad, který se ovšem posléze ukázal být jako ne zcela ideálním řešením. Proto bylo potřeba vymyslet nový přístup, kterým by bylo možné dvojrozměrná data dostat do zmíněného programu. Tímto přístupem bylo vytvoření skriptu, který mj. vytvořil další proměnné, jenž následně vstupovaly do programu VE Console. Obě metodiky budou podrobně v této kapitole popsány níže.

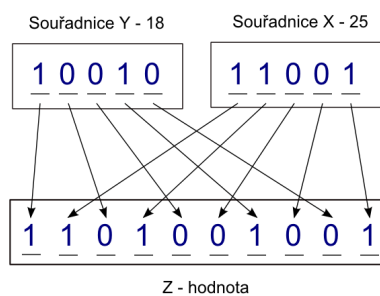
### 4.1 Mortonův rozklad

Mortonův rozklad (také *Z-order curve*, nebo *Morton code*), je funkce která mapuje vícerozměrný prostor do jednorozměrného. Jednorozměrná hodnota (Z-hodnota) souřadnice je jednoduše vypočtena střídavým prokládáním hodnot binární reprezentace této souřadnice. Vícerozměrný, nejčastěji dvourozměrný, prostor je procházen v daném pořadí, které je zobrazeno na (Obr.12). Princip prokládání je uveden na (Obr. 13). Předpokládejme, že máme souřadnici  $[25, 18]$ . Každé z těchto čísel je reprezentováno binárním zápisem, konkrétně  $[11001, 10010]$ . Následně je vždy vybrána nejprve první hodnota souřadnice  $Y$  ( $1$ ), následována první souřadnicí  $X$  ( $1$ ). Poté druhá souřadnice  $Y$  ( $0$ ) a druhá s souřadnice  $X$  ( $1$ ), takto je postupováno až k poslední hodnotě souřadnice  $X$ .



Obr. 12: Pořadí průchodu dvojrozměrným prostorem při Mortonově rozkladu.

(zdroj: Wikipedie, 2013)



Obr. 13: Princip prokládání hodnot binární reprezentace souřadnice  $[25,18]$ .

Ačkoliv tato metoda zachovává při převodu jednoznačnost vstupních hodnota a zaručuje také přímou možnost zpětné transformace do dvourozměrného prostoru, zanedbává prostorové závislosti vstupních dat, tedy způsobí, že nejsou zachovány poměrové vzdálenosti mezi body. Například před transformací je mezi souřadnicemi  $[4, 5]$  a  $[4, 6]$  vzdálenost jedna. Po transformaci jsou tyto body binárně zapsány jako  $110010$  a  $111000$ , mezi nimiž je vzdálenost 6. Porovnáme-li tyto souřadnice s další dvojicí

souřadnic, kde je vzdálenost jedna, např.  $[4, 7]$  a  $[4, 8]$ , které jsou po transformaci zapsány jako  $111010$  a  $1100000$ , dostaneme vzdálenost 38. Z čehož vyplývá, že transformace nezachovává relativní vzdálenosti mezi vstupními body.

Dalším problémem je velký rozsah možných hodnot po transformaci. Jelikož bylo pracováno s daty nad obrázky s rozlišením 1680 na 1050 pixelů, existuje v takovém případě od souřadnice  $[0, 0]$  po  $[1680, 1050]$  1 766 731 možných bodů. Vzhledem k tak velkému rozsahu hodnot nedochází v rámci jednoho záznamu k jejich opakování, což vede k problému s pozdější implementací Voting Experts algoritmu, který a ani s využitím DTW post-procesu není schopen smysluplně s křivkou pracovat. Všechny segmenty křivky se zdají jako odlišné, ačkoliv mohou být některé z nich relativně blízko sebe. Tento problém bylo zkoušeno vyřešit vzorkováním naměřených dat do upraveného rozsahu, nebo-li snižováním přesnosti dat. Například při vydělení dat před transformací, deseti, by se z původního rozsahu  $[1680, 1050]$  získal výrazně zmenšený rozsah  $[168, 105]$ , kde by se pravděpodobnost opakování hodnot výrazně zvýšila. Možných bodů by v takovémto případě bylo již pouze 17 914, tedy sto krát méně. Bylo vyzkoušeno podělit data číslem 2 a 4, ale ani takovéto snížení přesnosti nevedlo k adekvátním výsledkům. Ukázalo se tedy, že princip využití Mortonova rozkladu není pro účely této práce ideálním řešením k převodu dat z dvourozměrného prostoru do jednorozměrného.

## 4.2 Analýza souřadnicových komponent

Po selhání předchozí metody byla vymyšlena nová metodika, která umožnila nahrát dvourozměrná data do programu VE Console a následně zajistila efektivní segmentaci vstupních dat. Tento přístup lze označit jako analýza souřadnicových komponent a byla zpracována do skriptu v programu MATLAB.

### 4.2.1 Načtení proměnných

Ve skriptu (*Příloha 1\_1*) je nejprve potřeba přepsat název vstupního souboru a název vybraného stimulu, čímž je určeno pro kterého uživatele a který konkrétní předkládaný obrázek budou data následně upravena. K účelu diplomové práce byly data předzpracovány pro dvacet uživatelů a vybraným stimulem byl obrázek *010 - 3d-2d.jpg*. Jednalo se o vedle sebe umístěné mapy 3D a 2D, kde před promítnutím obrázku byla uživateli položen úkol, aby v této dvojici map našel místo s největším výskytem čerpacích stanic. Po přepsání těchto dvou parametrů je skript již zcela automatický a do ostatních konfigurací není potřeba zasahovat.

Do skriptu jsou pro jeho případnou větší variabilitu nahrány data všech atributů ze vstupních dat, ačkoliv je dále využito pouze času (*Time*) a souřadnic levého oka (*X* a *Y*). Jelikož se ve vstupních datech také občas objevovaly na pozicích souřadnic nulové a záporné hodnoty, byla sestavena podmínka, kdy bylo dále pracováno pouze s hodnotami většími než nula a zároveň musely být menší u hodnot *Y* než 1050 a u *X* menší než 1680,

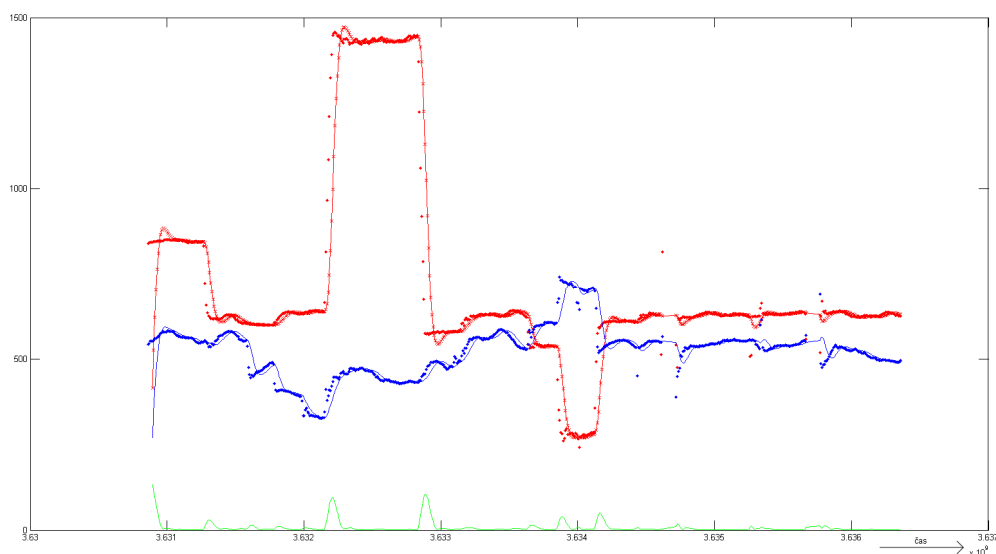
což je velikost rozlišení monitoru na kterém probíhalo testování a tudíž i rozlišení obrázků. K záporným hodnotám mohlo docházet v případě, že se uživatel na chvíli podíval mimo rozměry monitoru a nulové hodnoty vypovídají o ztrátě signálu, což mohlo být způsobeno delším mrknutím oka.

#### 4.2.2 Zjednodušení dat

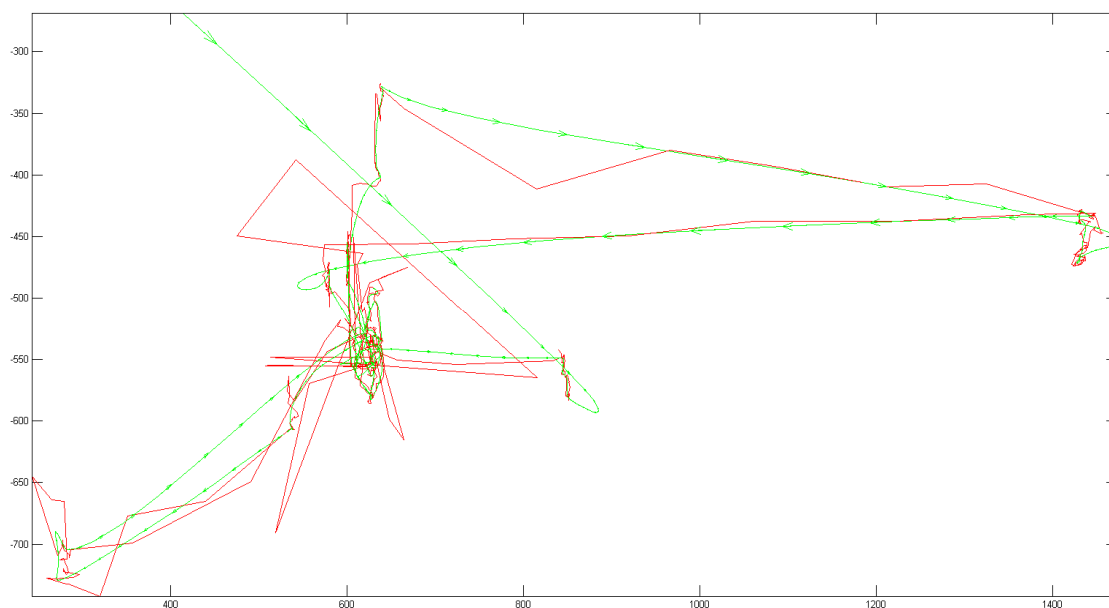
Vzhledem k tomu, že data obsahují přesný záznam toho, kam všude se uživatel díval, jedná se o velmi nepřehlednou a složitou křivku. Aby bylo možné v takovéto křivce následně nalézt podobné opakující se segmenty, bylo potřeba záznam zjednodušit. K tomuto účelu byla použita dolní propust (*Low-pass filter*), filtr, který nepropouští signál vyšších frekvencí.

Jedinou menší nevýhodou dolní propusti je zpoždění filtru. To znamená, že aby filtr byl schopen zjednodušit data, potřebuje k výpočtu pár hodnot před samotným začátkem upravované křivky. V tomto případě bylo zvoleno, že se před první bod křivky doplní dalších pět nových bodů, které se umístí postupně za sebe od tohoto počátečního bodu křivky směrem k počátku souřadnicového systému, který se nachází v levém horním rohu. Tato skutečnost má za následek, že nyní všechna takto upravená data budou začínat křivkou vedoucí z horního levého rohu směrem ke středu obrázku, tedy k místu, kde respondent začal pozorovat obrázek. Takovéto předkreslení křivky však nebylo pro účely práce považováno za problém. Ve velké většině případů byla právě tato část křivky zvolena při segmentaci za první segment, tudíž jen v závěrečném porovnávání bylo potřeba tuto skutečnost neopomenout.

Skript následně vykreslil graf (Obr. 14) a obrázek (Obr. 15) se srovnáním, jak data vypadaly před a po použití dolní propusti.



Obr. 14: Srovnání hodnot X a Y v závislosti na čase, před a po použitím dolní propusti. Červené body - vstupní hodnoty X; modré body - vstupní hodnoty Y; červená linie - hodnoty X po použití filtru; modrá linie - hodnoty Y po použití filtru; zelená linie - rozdíl vzdáleností dvou sousedních bodů.



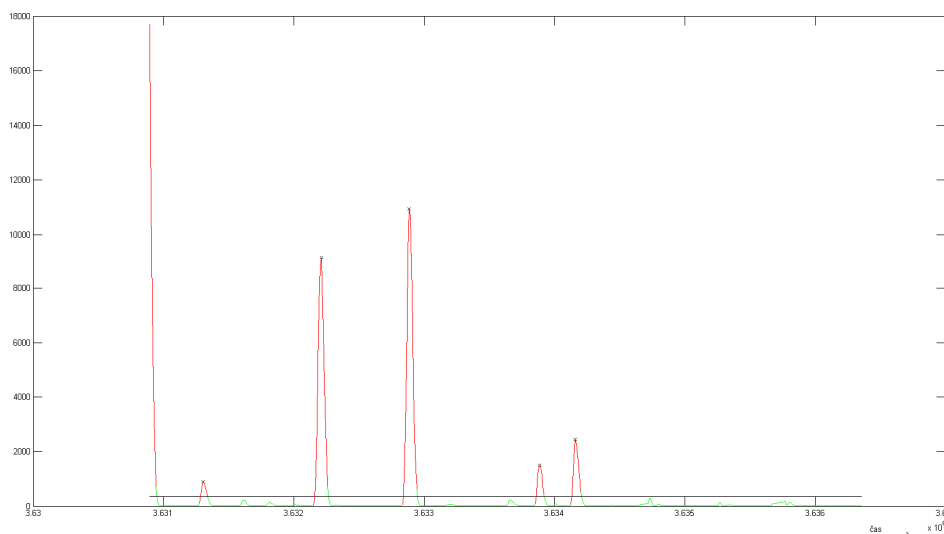
Obr. 15: Srovnání vstupních dat před a po použití dolní propusti.  
 Červená - vstupní data před použitím dolní propusti; zelená - data po  
 použití dolní propusti.

### 4.2.3 Rozdělení dat na fixace a sakády

V místech fixací, tedy místech kde se uživatel díval delší dobu na jedno a totéž místo, je záznam reprezentován spleť velkého množství bodů křivky, nad velmi malou lokalitou mapy. Jelikož hledat smysluplné epizody křivky v místech těchto fixací, by bylo dosti bezpředmětné a nikterak by to nevypovídalo o principu uživatelského čtení mapy, bylo rozhodnuto, že pro pozdější práci budou využity pouze sakády. Další součástí skriptu bylo tudíž rozdělení dat na fixace a sakády.

Nejprve byly dle euklidovské metriky vypočteny hodnoty vzdáleností dvou sousedních bodů, což je znázorněno i na (Obr. 14), zelenou barvou. Poté byla vypočtena průměrná hodnota těchto vzdáleností, která byla dále definována za prahovou hodnotu. Dle spočtených maxim nad tímto prahem bylo následně vypočteno, že všechny hodnoty křivky, které se nacházejí nad prahovou hodnotou jsou definovány jako sakády, zatímco všechny hodnoty pod stanoveným prahem jsou považovány za fixace (Obr. 16).

Jelikož v těsném okolí prahové hodnoty dochází k přechodné oblasti při dělení dat na fixace a sakády, nebylo vhodné data striktně rozdělovat přesně touto hodnotou. Při hledání fixací byla tedy tato průměrná hodnota vzdáleností vynásobena číslem 0,8 a v případě hledání sakád číslem 0,9. Tyto hodnoty byly stanoveny empiricky.



Obr. 16: Rozdělení dat na sakády a fixace.  
*Červená - sakády; zelená - fixace; černá - prahová hodnota.*

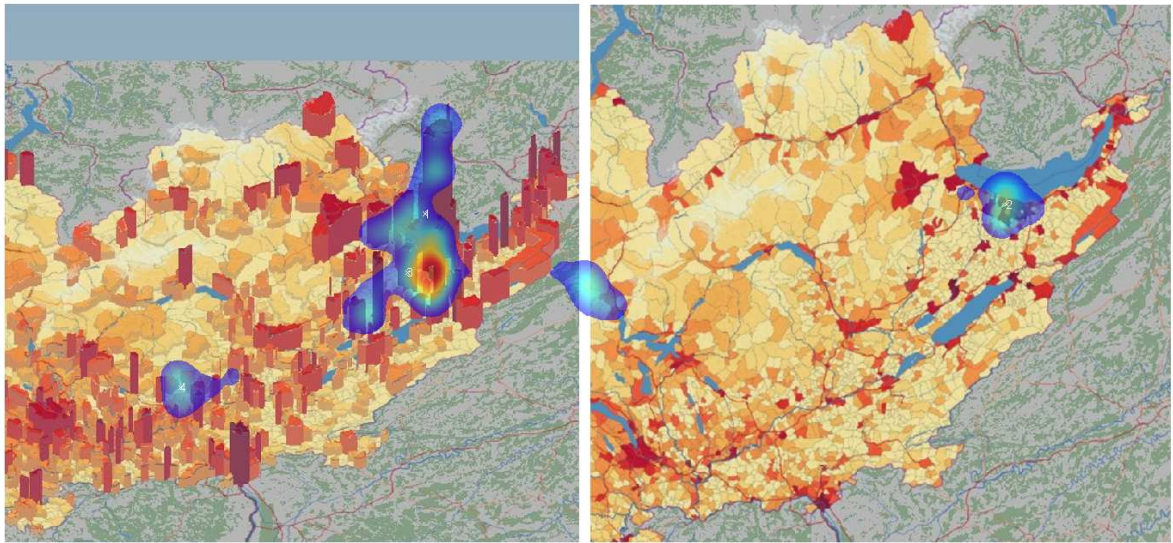
#### 4.2.4 Vytvoření HeatMap

Aby byly oblasti fixací lépe zřejmé a aby bylo vůbec lépe graficky zobrazeno, jak daný uživatel předloženou mapu vnímal, bylo využito jedné z nejpoužívanějších vizualizačních metod eye-tracking systému - HeatMap. Pro tvorbu HeatMap byl využit Image Processing Toolbox vývojového prostředí MATLAB. Následně bude proto uveden jen obecný postup tvorby těchto map.

Nejprve bylo vytvořeno pole o rozměrech 1680 na 1050 pixelů, do bodů tohoto pole byla dosazena četnost, s jakou se oko v daném místě vykytovalo. Takto byl vytvořen umělý obrázek, který obsahoval nenulové hodnoty pixelů pouze v místech, kde byla naměřena poloha oka.

Následně byla provedena konvoluce s Gaussovou funkcí, to znamená, že hodnota každého pixelu byla rozprostřena do okolních 100 x 100 pixelů a posléze byly všechny hodnoty sečteny. Aby bylo možné zobrazit HeatMapu nad podkladovým obrázkem, bylo ještě potřeba definovat průhlednost. Tohoto bylo docíleno nastavením úplné průhlednosti pro hodnoty HeatMapy nižší než námi definovaný práh a poloprůhlednosti pro všechny ostatní. Na závěr už byly pouze do obrázku zakresleny křížky v centrálních místech fixací, s číselným označením dle pořadí v závislosti na čase pozorování. (Obr. 17)



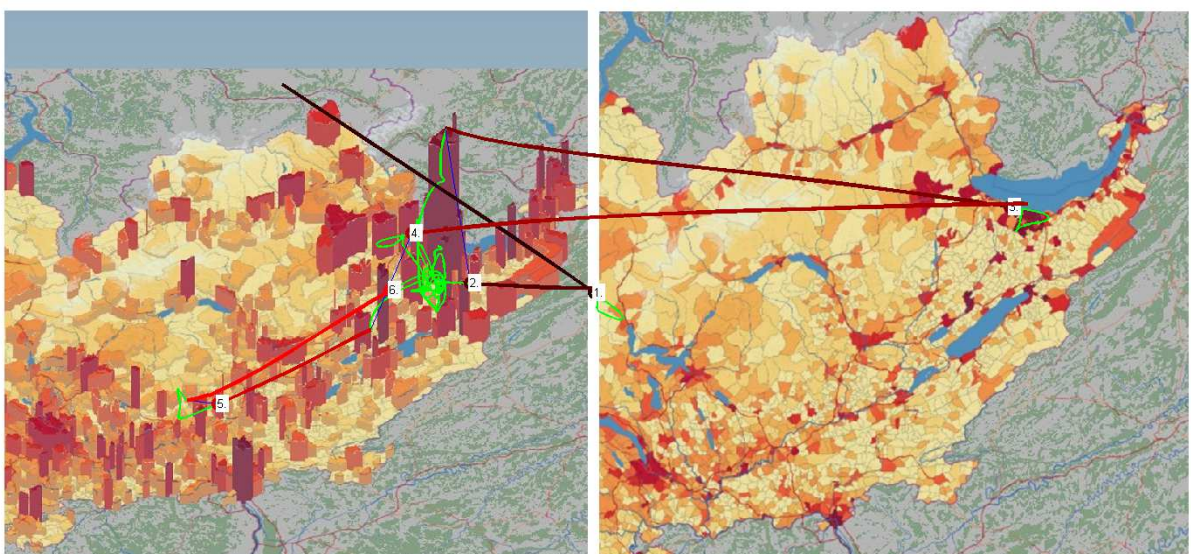


Obr. 17: HeatMapa.

#### 4.2.5 Vykreslení sakád

Vzhledem k tomu, že pro pozdější práci při segmentaci dat bylo rozhodnuto využívat pouze sakády, byl dalším grafickým výstupem skriptu obrázek (Obr. 18), na kterém jsou tyto sakády vykresleny.

Jelikož byly ze vstupních dat odstraněny body křivky v místech fixací, tak z důvodu aby byly sakády vykresleny ve stanoveném pořadí za sebe, byl vždy spojen nejkratší možnou vzdáleností koncový bod jedné sakády s počátečním bodem sakády následující. Tyto úseky jsou tedy definovány pouze dvěma body.



Obr. 18: Vykreslení sakád.

*Zelená - vstupní data; červená škála - sakády, modrá - spojnice sakád.*

#### 4.2.6 Vytvoření nových proměnných

Jak bylo v úvodu této kapitoly řečeno, bylo potřeba vymyslet nový přístup, který by umožnil nahrát dvourozměrná data do programu VE Console. Takovýmto nově zvoleným přístupem bylo vytvoření nových proměnných, které by zároveň s rozdělenými souřadnicemi X a Y vstupovaly do programu samostatně. Ke každému bodu vstupní křivky byly tedy vypočteny polární souřadnice  $[r, \varphi]$ :

$$r_i = \sqrt{x_i^2 + y_i^2}, \quad \varphi_i = \arctg\left(\frac{y_i}{x_i}\right),$$

kde  $x_i$  a  $y_i$  jsou souřadnice  $i$ -tého bodu. A euklidovská vzdálenost  $m$  mezi sousedními body:

$$m_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}, \quad i = 2, 3, \dots, N$$

Počátek soustavy byl stejně jak pro kartézské, tak i polární souřadnice v levém horním rohu obrázku.

Poslední součástí skriptu tedy bylo vypočítat tyto proměnné a následně je zapsat do pěti jednotlivých nových textových souborů - *poloha X*, *poloha Y*, *poloha r*, *poloha  $\varphi$*  a *euklidovská vzdálenost mezi sousedními body*. Pro pozdější práci se skriptem, který pracuje s výstupy z programu VE Console, je výstupem tohoto skriptu ještě šestý textový soubor, ve kterém jsou zapsány souřadnice X a Y součastně. Vytvoření těchto souborů je provedeno do uživateli libovolně zvoleného adresáře, který lze vybrat v dialogovém okně, které je zobrazeno po tom, co jsou vygenerovány všechny grafy a obrázky představené v tomto skriptu.

Tou úplně poslední, avšak nepovinnou součástí skriptu je vytvoření dalšího textového souboru, který slouží jako vstupní data do skriptu generujícího videozáznam, který je popsán v následující kapitole. Obsahuje hodnoty (*čas*, *X*, *Y*).

#### 4.2.7 Vytvoření videozáznamu

Pro názornou ukázkou toho, kam všude a jak dlouho se uživatel na překládanou mapu díval, byl vytvořen další skript (*Příloha I\_2*), který generuje videozáznam využívající zmíněnou metodiku tvorby HeatMap (kap. 4.2.4). Kde každý frame obsahuje HeatMapu sestavenou nad dvaceti body vstupních dat, což odpovídá jedné třetině sekundy záznamu. Jelikož není potřeba pracovat se všemi body, bylo postupováno následujícím způsobem, kdy první frame je vytvořen pro prvních dvacet bodů, druhý frame pro 3. - 23. bod, třetí frame pro 5. - 25. bod atd. Skrze vysokou výpočetní náročnost skriptu trvá vygenerování záznamu poměrně dlouhou dobu. U standardně vybaveného PC v řádu několika hodin. Videozáznam je ukládán ve formátu mp4. Ukázkový videozáznam byl vytvořen pro respondenta P19\_C (*Příloha I\_8*).



## 5 IMPLEMENTACE VOTING EXPERTS ALGORITMU S DTW POST-PROCESEM

Již několikrát bylo v této práci zmíněno, že pro segmentaci dat bylo využito programu VE Console, který kombinuje algoritmy Voting Experts a Borcení časové osy. Avšak pro jeho správnou implementaci je potřeba nastavit celou řadu parametrů, které jsou více popsány v Kocyan a kol. (2012). Vzhledem k tomu, že autor testoval nastavení těchto parametrů pro zcela odlišný typ vstupních dat, kterými byl textový řetězec, bylo nutné provést zcela nové testování.

### 5.1 Testování nastavení parametrů programu VE Console

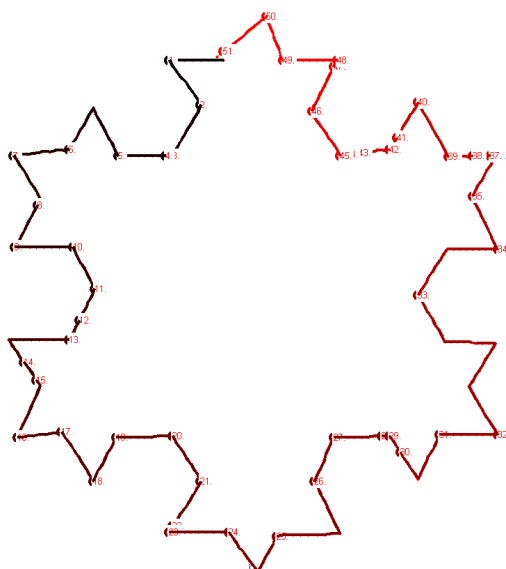
Pro otestování vhodného nastavení parametrů u programu VE Console byly nakresleny vlastní křivky s takovými charakteristickými rysy, na kterých by následně mohlo být zřetelně posouzeno, zda byla křivka smysluplně rozdělena. K nakreslení křivek byl využit editor vektorové grafiky Inkscape. Nakreslený záznam bylo potřeba uložit s relativními souřadnicemi ve formátu Plain SVG (*Scalable Vector Graphics*), který má syntaxi zápisu založenou na XML (*Extensible Markup Language*), k jeho otevření tak postačí běžný textový editor.

Aby bylo možné dále pracovat se souřadnicemi bodů takto nově nakreslené křivky, stejným způsobem jako s daty z eye-tracking systému, byl opět v programu MATLAB vytvořen jednoduchý skript, který souřadnice převede do požadovaného formátu. Jelikož jsou souřadnice v SVG souboru uloženy v jednom řádku s pevně danou strukturou zápisu, bylo jen potřeba nakopírovat tento řádek do proměnné ve vytvořeném skriptu. Ten souřadnice zapsal do textového souboru v klasické sloupcové struktuře (sloupec X, sloupec Y). Jelikož nebylo potřeba tyto postupy mnohokrát opakovat, vypočtení hodnot polárních souřadnic a euklidovské vzdálenosti již nebylo do skriptu implementována a byly vypočteny pouze v Microsoft Office Excelu. Takto vytvořená zkušební data následně vstupovala do programu VE Console, ve kterém bylo testováno nastavení parametrů.

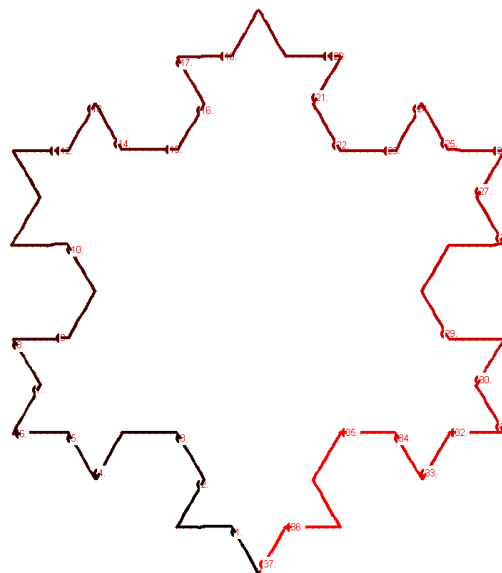
Během testování bylo zjištěno, že největší pozornost je potřeba věnovat parametrům - *hloubka stromu*, *délka plovoucího okna s prahem* a *počet interací post-procesu*. Nastavení ostatní parametrů nemělo na výslednou segmentaci žádný větší vliv, proto byly tyto parametry ponechány s původními hodnotami. Prahovou hodnotu post-procesu zmíněnou v kapitole 3.4 se nebylo v tomto případě vůbec potřeba zabývat, jelikož její implementace je v práci zařazena až později (kap. 5.3). Nejvhodnějším nastavením bylo tedy empiricky stanoveno pro parametry:

- Hloubka stromu: 30
- Délka plovoucího okna s prahem: 30; 1
- Počet interací post-procesu: 4

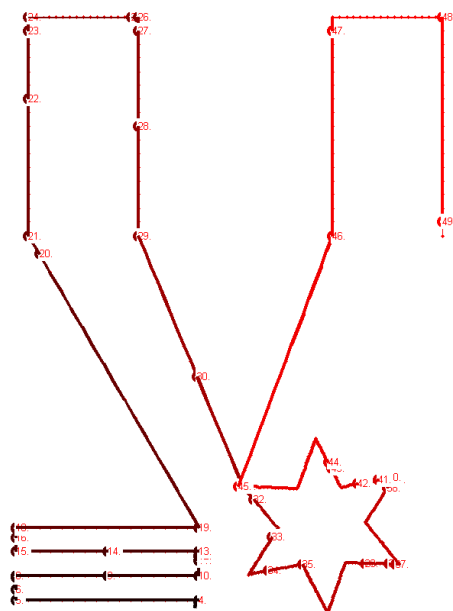
Rozdíl mezi vhodně a nevhodně zvolenými parametry je patrný na obrázcích (Obr. 19 - Obr. 22). Zatímco při špatném nastavení dochází k segmentaci předkládané křivky ve značně nelogických místech, tak u vhodně zvolených parametrů lze pozorovat segmentaci křivky především v lomových bodech. Jelikož smyslem segmentace je nalezení podobných epizod v datech, lze u těchto příkladů konstatovat, že algoritmus vyhodnotil rovné části křivky za podobné, tudíž v případech kde docházelo k zalomení křivky, tak v



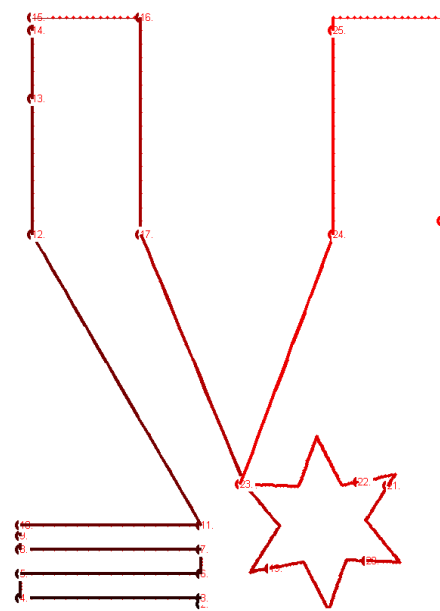
Obr. 19: Ukázka nevhodného nastavení parametrů na testovacích datech.



Obr. 20: Ukázka vhodně zvolených parametrů na testovacích datech.



Obr. 21: Ukázka nevhodného nastavení parametrů na testovacích datech.



Obr. 22: Ukázka vhodně zvolených parametrů na testovacích datech.

## 5.2 Nasazení programu VE Console

Výše uvedené nastavení programu VE Console bylo aplikováno pro data všech dvaceti uživatelů, přičemž do programu jednotlivě vstupovalo postupně vždy pět souborů ( $X$ ,  $Y$ ,  $r$ ,  $\varphi$  a *euklidovská vzdálenost sousedních bodů*). Pro každý soubor bylo vždy potřeba ve zdrojovém kódu aplikace přepsat aktuální umístění a název vstupu i výstupu pro každý jednotlivý textový soubor.

Výstupem z programu je opět pro každého uživatele pět textových souborů ( $X$ ,  $Y$ ,  $r$ ,  $\varphi$  a *euklidovská vzdálenost sousedních bodů*), kde v každém jsou uloženy informace o počtu přiřazených hlasů. Pro každou hodnotu je zde zobrazen počet hlasů od entropy a frequency experta, počet hlasů dosažených v každém kroku post-procesu, součet všech hlasů a doporučení kde by mělo dojít k segmentaci křivky, na obrázku znázorněno svislou čarou (Obr. 23).

Zmíněné doporučení o lokalitě, kde by mělo dojít k řezu křivky je ovšem vypočteno dle prahové hodnoty post-procesu, definované v kapitole 3.4. Avšak jak již bylo výše zmíněno, tento parametr nebyl v rámci této práce nastavován, jelikož k jeho zvolení dojde až posléze. Tyto doporučení tedy nemají žádný vliv pro pozdější segmentaci křivky. Stěžejní informací jsou v tomto případě počty hlasů, které budou v následujícím kroku vyhodnoceny a na jejich základě dojde k řezům.

RAW Input	43	43	43	43	43	43	45	46	47	49	51
EntropyExpert	0	0	0	0	1	4	0	3	1	1	1
FrequencyExpert	0	0	0	1	1	3	0	2	1	0	0
PostProcess 0	0	0	0	0	0	0	0	0	0	0	0
PostProcess 1	0	0	0	0	0	0	0	0	0	0	0
PostProcess 2	0	0	0	0	0	0	0	0	0	0	0
PostProcess 3	0	0	0	0	0	0	0	0	0	0	0
Total votes	0	0	0	1	2	7	0	5	2	1	1
Cuts											

Obr. 23: Část výstupních dat z programu VE Console pro proměnnou  $\varphi$ , uživatele P16\_C.

1. řádek - vstupní hodnoty; 2. a 3. řádek - počet hlasů od entropy a frequency experta; 4. až 7. řádek - počet hlasů pro každou interaci postprocesu; 8. řádek - součet všech hlasů; 9. řádek - doporučení k provedení řezu.

## 5.3 Sčítání hlasů

Pro vyhodnocení hlasů získaných v předešlém kroku byl vytvořen další skript v programu MATLAB (*Příloha I\_3*), který nejprve načte skrze dialogové okno vybranou cílovou složku s pěti výstupními soubory z programu VE Console a sečte dohromady jejich hodnoty s celkovým počtem hlasů (Total votes). Tímto sečtením všech hlasů u daného uživatele, jsou vysokými hodnotami indikovány lokality potencionálních řezů.

Aby bylo možné přesně určit ve kterých lokalitách k řezu dojde, je potřeba stanovit prahovou hodnotu, kterou je nutno překročit. Kocyan a kol. (2012) ji implementovali přímo do programu VE Console a k její definici využili vztah uveden zde v kapitole 3.4. V případě této práce bylo stanovení prahové hodnoty empiricky testováno v rámci

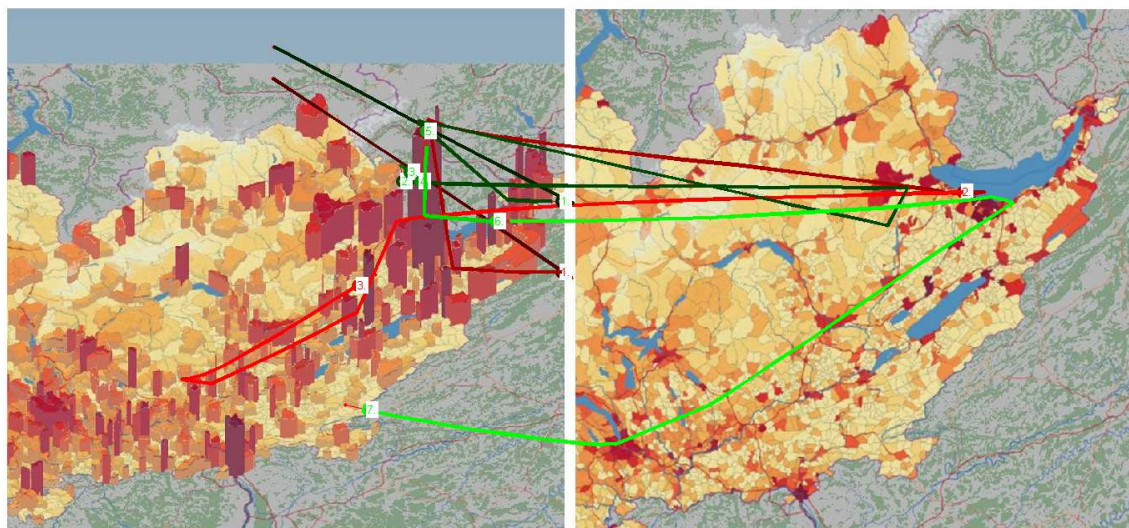
kapitoly 5.1 a její implementace byla zahrnuta do tohoto skriptu. Stanovenou fixní prahovou hodnotou je 13. Aby tedy došlo k řezu křivky, musí konečný součet všech hlasů překročit hodnotu 13, bude-li tato podmínka splněna, dojde k řezu. U testovaných dvaceti uživatelů bylo dle této prahové hodnoty vytvořeno 320 segmentů, jenž byly pro pozdější porovnávání v rámci skriptu uloženy do textových souborů (každému uživateli jeden soubor). Přehled těchto segmentů je uveden v příloze (*Příloha 1\_4*).

## 6 VIZUALIZACE A PROVNÁVÁNÍ SEGMENTŮ

Jelikož jedním z cílů práce bylo vyhledat podobné segmenty, svědčící o podobném postupu uživatelů při řešení zadaného úkolu, bylo pro porovnání výsledných segmentů použito dvou přístupů. První metodou bylo pouze informativní, vizuální porovnání vytvořených řezů v datech, mezi dvěma uživateli. Druhým zvoleným přístupem bylo porovnávání všech jednotlivých segmentů mezi sebou navzájem. K tomu bylo využito tzv. histogramů, jenž vyjadřovaly četnost bodů každého segmentu, příslušných k danému rozdělení prostoru.

### 6.1 Vizuální porovnání

Po té, co byla ve skriptu uvedeném v kapitole 5.3 zadána prahová hodnota a byly načteny všechny výstupní soubory z programu VE Console, je ještě potřeba zvolit opět skrze dialogové okno příslušný textový soubor, jenž současně obsahuje X a Y souřadnice vstupních dat (zmíněno v kap. 4.2.6) a také podkladovou mapu. V případě vykreslení dvou uživateli postačí pouze odkomentovat část skriptu a při jeho spuštění je vyžádáno nahrání výstupních dat z VE Console a textového souboru se souřadnicemi pro tohoto druhého uživatele. Prahovou hodnotu, ani výběr podkladu již podruhé není vyžadován. Jakmile jsou potřebná data nahrána, je vykreslen obrázek s výslednými segmenty (Obr. 24). Pro každého vybraného uživatele je zvolena odlišná barva, škálována od nejtmaší po nejsvětější, v závislosti na čase při pozorování předkládaného stimulu. Aby bylo ještě lépe rozlišitelné pořadí řezů, je vždy na konci vytvořeného segmentu příslušné číselné označení. Tímto přístupem lze vizuálně porovnat nejen dva uživatele mezi sebou, avšak v případě jejich většího počtu se mapa stává nepřehlednou.



Obr. 24: Vizuální porovnání vytvořených segmentů pro dva respondenty navzájem.

Červená škála - respondent P\_19\_C (3 segmenty); zelená škála - respondent P\_09\_C (7 segmentů); odstín škály vyjadřuje časovou složku, od tmavé po světlou.

## 6.2 Porovnávání pomocí histogramů

Pro vzájemné porovnání všech vytvořených 320 segmentů u všech dvaceti uživatelů navzájem, byla vymyšlena metodika, založená na srovnávání složitosti těchto jednotlivých segmentů. Nejprve byly pro každý segment vypočteny souřadnice jeho těžiště  $[x_T, y_T]$ , dle vztahu:

$$x_T = \frac{1}{N} \sum_{i=1}^N x_i,$$

$$y_T = \frac{1}{N} \sum_{i=1}^N y_i,$$

kde  $N$  je počet bodů daného segmentu a  $[x_i, y_i]$  souřadnice  $i$ -tého bodu tohoto segmentu.

Poté se vytvořil šestnáctidílný prostor se středem v tomto těžišti a spočetlo se kolik bodů zkoumaného segmentu spadá do každého dílu takto vytvořeného prostoru. K tomuto byl nejprve vypočten úhel mezi nulovou osou prostoru, která byla určena svisle nahoru, a spojnicí daného bodu křivky s těžištěm:

$$\varphi_i = \arctg \left( \frac{y_i - y_T}{x_i - x_T} \right) .$$

Následně byl získán index dílu prostoru  $c_i$ , do kterého spadá  $i$ -tý bod křivky:

$$c_i = \left\lfloor \frac{\varphi_i}{\frac{2\pi}{16}} \right\rfloor + 1, \quad i = 1, 2, \dots, N,$$

kde  $\lfloor \cdot \rfloor$  značí celou část čísla.

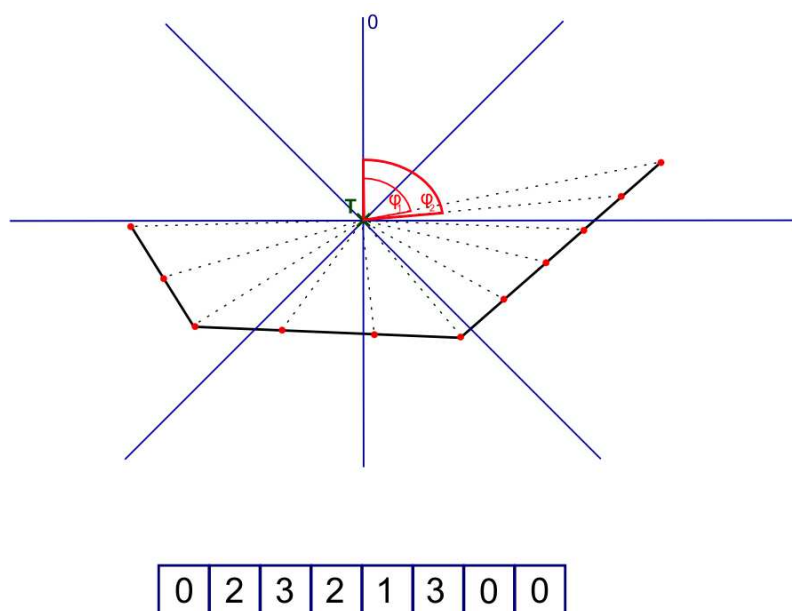
Pro zjištění počtu bodů v  $j$ -tém dílu prostoru bylo využito vztahu:

$$p_j = \sum_{i=1}^N \delta_{jc_i}, \quad j = 1, 2, \dots, 16,$$

kde pro Kroneckerovo delta  $\delta_{jc_i}$  platí:

$$\delta_{jc_i} = \begin{cases} 1 & \text{je-li } c_i = j \\ 0 & \text{je-li } c_i \neq j \end{cases} .$$

Zapsáním těchto vypočtených hodnot do šestnácti prvkového pole byl vytvořen, tzv. histogram četností bodů, příslušných k danému rozdělení prostoru. Tento histogram byl vypočten pro všech 320 segmentů. Výše uvedený princip je zobrazen na (Obr. 25), kde pouze místo šestnáctidílného prostoru je pro větší přehlednost použit prostor osmidílný.



Obr. 25: Princip vytvoření histogramu četností bodů, příslušných k danému rozdělení prostoru, nad vybraným segmentem.

Ke konečnému porovnání podobnosti všech segmentů mezi sebou, bylo užito vzorce:

$$\cos \varphi = \frac{H_1 \cdot H_2}{|H_1| |H_2|},$$

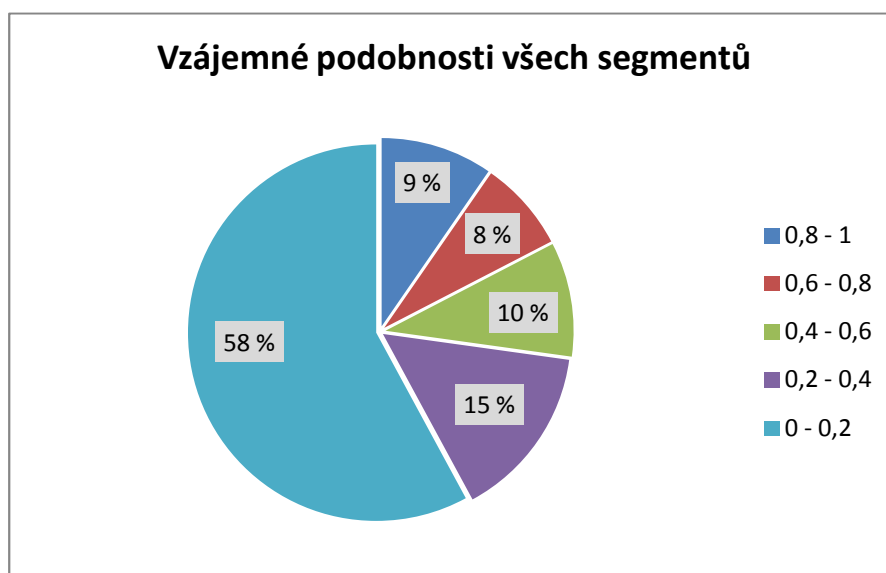
kde histogram jednoho segmentu je považován za vektor  $H_1$  a histogram segmentu druhého, za vektor  $H_2$ . Výsledkem jsou potom hodnoty v intervalu  $\langle 0, 1 \rangle$ , kde v případě, že  $\cos \varphi = 1$  jsou si dva segmenty zcela podobné a v případě kdy  $\cos \varphi = 0$  není nalezena mezi segmenty podobnost vůbec žádná.

Přehled všech histogramů (*Příloha 1\_5*), segmentů (*Příloha 1\_4*) a hodnot podobností (*Příloha 1\_6* a *Příloha 1\_7*) byl vytvořen pomocí sázecího systému LaTeX.

## 7 VÝSLEDKY

Využitím výše představené metodiky, která porovnávala všech 320 segmentů vzájemně mezi sebou, pomocí histogramů četností bodů, příslušných k danému rozdělení prostoru, bylo získáno konečných 79 806 hodnot. Jelikož byl porovnáván každý segment s každým, dochází zde však ke zdvojení každé hodnoty, kdy segment č. 1 je porovnán se segmentem č. 2 a rovněž segment č. 2 je porovnán se segmentem č. 1. Vzhledem k tomuto je tedy možno konečný počet porovnávaných zjednotit na 39 903. Ovšem i takto vysoký počet dosažených výsledků je velmi těžko interpretovatelný.

Jedním z výstupů porovnávaných hodnot je přehled všech podobností (*Příloha 1\_6*), z něhož je sestaven následující graf, který vypovídá o vzájemné podobnosti mezi všemi segmenty. Jelikož výsledky podobnosti nabývají hodnot od 0 do 1, kde 1 znamená absolutní shodu, lze z grafu snadno vyčíst míru podobností (Graf 1).



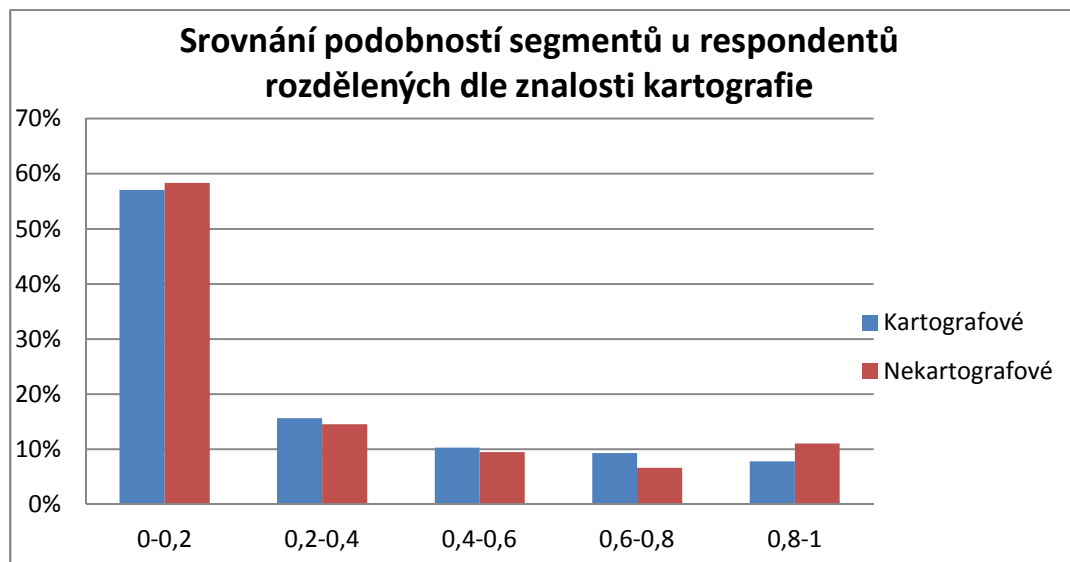
Graf 1: Vzájemné podobnosti všech segmentů

Z tohoto grafu je patrné, že pouze 9 % všech segmentů nabývá nejvyšších hodnot podobnosti, avšak za velmi podobné lze považovat i následujících 8 % segmentů, které odpovídají 0,6 až 0,8 míře podobnosti. Oproti tomu, více než 50 % segmentů nedosahuje ani míry podobnosti 0,2 z čehož lze vyvodit, že v porovnávaných datech se příliš podobností nevyskytuje. Tyto údaje ovšem není možno brát za zcela objektivní, jelikož zde není brána v potaz například prostorová poloha, nebo délka křivek, čímž může být srovnáván velmi krátký segment definovaný pouze dvěma body se segmentem o mnoho větší délce. V konečném důsledku tak krátký segment příliš nevypovídá o charakteristickém přístupu čtení podkladové mapy, stejně jako segment delší. Avšak ve zmíněném porovnávání je mu přiřazena stejná váha. Do grafu nebyly naopak započteny všechny první segmenty, jelikož se jedná o křivky vytvořené dolní propustí při úpravě dat. Jak bylo upozorněno v kapitole 4.2.2 jedná se vždy o rovnou křivku, která byla ve velké většině případů identifikována jako první segment každého uživatele. Tyto



segmenty jsou tedy zcela logicky vyhodnoceny jako naprosto identické a proto nebyly zahrnuty do výpočtu.

Jelikož použítá vstupní data obsahovaly informace o tom, zda respondent absolvoval alespoň jeden semestr kartografie, či nikoliv, byla vypočtena podobnost nalezených segmentů i pro tyto dvě skupiny odděleně a následně vynesena do grafu (Graf 2).



Graf 2: Srovnání podobností segmentů dle znalostí kartografie

Z takto zobrazených dat je možno vyčíst, že neexistuje žádný větší rozdíl v podobnosti čtení předkládané mapy mezi kartografy a nekartografy. Pro zcela objektivní srovnání je ovšem nutno připomenout, že obě skupiny nejsou zastoupeny přesným počtem respondentů. Bylo porovnáváno jedenáct nekartografů s devíti kartografy.

U této konkrétní úlohy, kdy byla respondentům současně v rámci jednoho obrázku zobrazena 2D a 3D mapa a položen úkol, aby označili místo s největším výskytem čerpacích stanic, lze konstatovat, že v použitých experimentálních datech nebylo nalezeno příliš mnoho podobných segmentů, které by svědčily o podobném postupu řešení úlohy respondenty. A zároveň neexistuje žádný větší rozdíl při čtení mapy v závislosti na tom, zda respondent měl, nebo neměl alespoň základní kartografické vzdělání.

Více vypovídajícím výstupem o porovnání podobností všech jednotlivých segmentů navzájem je dokument, který graficky porovnává vždy dva segmenty vedle sebe, včetně vypsání jejich hodnoty podobnosti (*Příloha I\_7*). Uveden je vždy první segment daného uživatele a k němu jsou postupně zobrazeny všechny ostatní segmenty, seřazené sestupně od nejpodobnějšího k tomu nejméně podobnému. Následuje druhý segment daného uživatele se všemi ostatními segmenty seřazenými dle podobnosti atd. Jak již ovšem z principu věci vyplývá, tento přehled grafických vizualizací je vzhledem ke své obsáhlosti značně nepřehledný. PDF soubor (*Portable Document Format*) obsahuje 12 681 stran. Zde je pro názornost uvedeno několik porovnání:

Segment $S_1$	Segment $S_2$	Podobnost $S_1$ a $S_2$
P32_C_Seg004	P23_N_Seg003	0
P32_C_Seg004	P23_N_Seg009	0
P32_C_Seg004	P06_C_Seg007	0
P32_C_Seg004	P42_N_Seg011	0
P32_C_Seg004	P42_N_Seg012	0

Obr. 26: Část porovnání segmentu P32\_C\_Seg004 s ostatními nejméně podobnými segmenty.

Segment $S_1$	Segment $S_2$	Podobnost $S_1$ a $S_2$
P12_C_Seg004	P12_C_Seg032	0,9608
P12_C_Seg004	P30_N_Seg007	0,9428
P12_C_Seg004	P42_N_Seg005	0,9129
P12_C_Seg004	P23_N_Seg011	0,8819
P12_C_Seg004	P12_C_Seg037	0,866

Obr. 27: Část porovnání segmentu P12\_C\_Seg004 s ostatními málo podobnými segmenty.

Segment $S_1$	Segment $S_2$	Podobnost $S_1$ a $S_2$
P36_N_Seg020	P24_N_Seg002	0,728
P36_N_Seg020	P22_C_Seg014	0,7248
P36_N_Seg020	P34_N_Seg013	0,7244
P36_N_Seg020	P09_C_Seg007	0,7225

Obr. 28: Část porovnání segmentu P36\_N\_Seg020 s ostatními segmenty.

Kromě dvou výše zmíněných PDF dokumentů je dále k dispozici ještě zejména pro dílčí informativní účely dokument obsahující přehled všech vytvořených histogramů (*Příloha 1\_5*) a dokument s grafickým přehledem všech vytvořených segmentů, polohou jeho těžiště a vypsáním histogramem (Obr. 29) (*Příloha 1\_4*).

## 8 Uživatel P19\_C

### 8.1 Segment P19\_C\_Seg001

Těžiště: (707, 222; 587)

Histogram četnosti bodů podle daného rozdělení prostoru: (0; 0; 0; 0; 0; 5; 0; 0; 0; 0; 0; 0; 0; 4; 0; 0)

Schématické zobrazení:

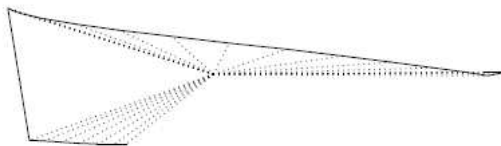


### 8.2 Segment P19\_C\_Seg002

Těžiště: (973, 333; 614, 292)

Histogram četnosti bodů podle daného rozdělení prostoru: (0; 1; 0; 8; 1; 0; 0; 0; 0; 0; 5; 2; 5; 2; 0; 0)

Schématické zobrazení:



Obr. 29: Část přílohy 1\_4, první dva segmenty uživatele P19\_C včetně jejich histogramů a vypočteného těžiště.

Vzhledem k tomu, že primárním cílem práce nebylo charakterizovat, nebo kategorizovat jednotlivé uživatele mezi sebou, dle vypočtených podobností segmentů, ale především implementovat algoritmy pracující s časovými řadami na data z eye-tracking systému a následně s jejich pomocí vyhledat segmenty svědčící o podobném postupu řešení úkolů respondenty, bylo rozhodnuto, že výše uvedená forma výstupů bude pro účely této práce zcela dostačující.

Z dosažených výsledků lze konstatovat, že s využitím kombinace algoritmů Voting Experts a Borcení časovou osou, je možno v datech naměřených eye-tracking systémem nalézt do jisté míry podobné opakující se segmenty, které mohou vypovídat o podobném přístupu uživatelů při čtení předkládaných map. V experimentálních datech pro tuto magisterskou práci bylo u testované úlohy nalezeno přibližně 17% velmi podobných segmentů.

Za výsledky této práce lze rovněž považovat vytvořené skripty v programu MATLAB (*Příloha 1\_1, Příloha 1\_2, Příloha 1\_3*), které jsou napsány a vhodně okomentovány tak, aby byly snadno využitelné pro jakákoliv Raw data z eye-tracking systému. Vzhledem k tomu, že program VE Console, jenž je součástí představené metodiky segmentace dat, byl využit pouze pro účely této práce a není nikterak veřejně poskytnut, lze skripty kromě segmentace využít minimálně pro úpravu a rozdělení vstupních dat na fixace a skákdy, tvorbu HeatMap a vygenerování videozáznamu.

## 8 DISKUZE

Stěžejním cílem a hlavní problematikou této diplomové práce bylo pomocí algoritmů pracujících s časovými řadami nalézt podobné segmenty v datech naměřených technikou eye-tracking. Jelikož bylo pro implementaci takovýchto algoritmů, konkrétně kombinací Voting Experts a Borcení časovou osou, rozhodnuto využít programu VE Console, který však byl primárně vytvořen k práci se zcela odlišným typem vstupních dat. Bylo jedním z hlavních problémů práce vymyslet přístup, který by umožnil aplikaci programu nad dvourozměrnými daty z eye-tracking systému, namísto dat jednorozměrných, pro které byl program vytvořen.

Po té, co se ukázalo použití Mortonova rozkladu pro transformaci dvourozměrných dat na data jednorozměrná za ne zcela ideální pro účely této práce, bylo rozhodnuto nahrát hodnoty X a Y vstupních dat do programu VE Console samostatně. Na základě pouze těchto dvou proměnných však nebylo následně možné určit lokality potencionálních řezů křivky s dostatečnou přesností. Proto byly vytvořeny ještě proměnné s polárními souřadnicemi a euklidovskou vzdáleností dvou sousedních bodů, které dodaly každému bodu k porovnání s ostatními body mnohem více informací. Tyto proměnné následně se souřadnicemi X a Y vstupovaly jednotlivě do zmíněného programu a jelikož při pozdějším testování přesnosti řezů docházelo k uspokojivým výsledkům, již nebylo zkoušeno přidání dalších proměnných, což by ovšem mohlo vést k lehce odlišným výsledkům.

Kromě vymyšlení přístupu implementace dvourozměrných dat do programu VE Console, bylo vzhledem k tomu, že zdrojový kód byl dodán pouze se základními komentáři, tedy bez jakékoliv dokumentace, vůbec problematické samotné pochopení principu fungování všech volitelných parametrů v tomto programu. Testování nastavení parametrů bylo proto prováděno pouze pro parametry Voting Experts algoritmu, které nejvíce ovlivňovaly výsledky segmentace. Z parametrů u DTW posprocesu byl testován pouze počet interací, ostatní parametry při jakékoliv změně nevykazovaly u výsledků větší rozdílnost. Nelze však vyloučit, že i některý z parametrů, kterému nebyla věnována větší pozornost by mohl mít vliv na odlišný výsledek segmentace.

Zřejmě nejslabším místem práce bylo nastavení prahové hodnoty, určující minimální počet všech sečtených hlasů, který byl potřeba k provedení řezu. Zatímco autor programu VE Console tuto prahovou hodnotu počítal flexibilně pro každý bod vstupní křivky, dle vzorce uvedeného v kapitole 3.4, vzhledem k postupnému nahrávání jednotlivých pěti proměnných do programu, nebylo možné tohoto definování prahu využít. Ten by totiž určil každé z pěti proměnných odlišné lokality řezu, které by následně nebylo možné smysluplně spojit dohromady. Proto byla tato prahová hodnota v práci určena fixně jedním číslem, které bylo určeno na základě testování na datech vytvořených v grafickém vektorovém editoru Inkscape.

Zde ovšem dochází k onomu zmíněnému slabému místu práce. Při tvorbě řady testovacích dat bylo zjištěno, že pokud je vytvořena křivka definovaná velkým počtem bodů, jsou výsledky segmentace značně odlišné od totožné křivky, avšak definované menším počtem bodů, myšleno v řádu stovek bodů. U vytvořených testovacích dat (Obr. 19 a Obr. 21) je proto použito kolem 200 bodů křivky, což je přibližná průměrná hodnota bodů u všech vstupních dat. Existují zde však výjimky, kde například vstupní data do VE Console od uživatele P\_38\_N obsahují pouze 45 bodů křivky, zatímco u uživatele P\_23\_N je křivka definována 265 body. Stanovením totožné prahové hodnoty pro oba uživatele má potom za následek, že u P\_38\_N byly vytvořeny pouze dva segmenty, zatímco u uživatele P\_23\_N bylo vytvořeno 49 segmentů. Jedná se ovšem o extrémní případy v použitých datech.

Jak bylo již dříve několikrát zmíněno, stěžejním cílem práce bylo především vymyslet metodiku zpracovávající data z eye-tracking systému, pomocí algoritmů pracujících s časovými řadami. Z tohoto důvodu nebylo do práce zahrnuto obsáhlejší porovnání segmentů v závislosti na jednotlivých uživateli, čímž se zabývá jiný výzkum na Katedře geoinformatiky UP v Olomouci. Ve výsledku jsou tak porovnávány všechny segmenty vzájemně mezi sebou, zatím co k porovnávání přístupu mezi jednotlivými uživateli by bylo vhodnější srovnat vždy jeden segment daného uživatele pouze se segmenty ostatních uživatelů. Další možným přístupem k porovnání vzájemné podobnosti je srovnávat segmenty s důrazem na jejich prostorové rozmístění, nebo časovou složku.

Touto problematikou, tedy porovnáváním jednotlivých segmentů mezi sebou v závislosti na nejrůznějších charakteristikách uživatelů dalších aspektů by se mohl v budoucnu zabývat nějaký další navazující výzkum.

## 9 ZÁVĚR

Hlavním cílem této magisterské práce bylo vyhledat pomocí algoritmů pracujících s časovými řadami podobné segmenty v experimentálních datech, pořízených metodou eye-tracking, které by svědčily o podobném postupu řešení úkolů respondenty. Práce je rozdělena do dvou základních částí, kdy v té první je představena metoda eye-tracking, která snímá pohyb lidských očí při vnímání vizuálního vjemu a algoritmy Voting Experts a Borcení časovou osou, které byly využity pro nalezení podobných segmentů. Druhá, hlavní část práce se věnuje praktickému řešení této problematiky.

Jedním z dílčích cílů diplomové práce bylo definovat vhodné úlohy, nad kterými by probíhalo testování pomocí eye-tracking systému. Jelikož již dříve na Katedře geoinformatiky Univerzity Palackého v Olomouci probíhalo testování zkoumající rozdílnost vnímání mezi 2D a 3D mapami, bylo rozhodnuto pro práci využít části těchto naměřených dat.

Jako zvolená implementace algoritmů Voting Experts a Borcení časovou osou byl použit program VE Console, vytvořený Ing. Tomášem Kocyanem. Vzhledem k tomu, že tento program vyžaduje vstupní data v jednorozměrném formátu, bylo potřeba vymyslet metodiku, která by umožnila aplikaci programu i nad dvojrozměrnými daty z eye-tracking systému. K tomuto byly vytvořeny v programu MATLAB skripty, které data předzpracují do požadované podoby a následně provedou segmentaci vstupních dat na smysluplné epizody.

Všechny takto vytvořené segmenty byly mezi sebou navzájem porovnány a byla vypočtena jejich vzájemná podobnost. V datech použitých pro tuto práci bylo vytvořenou metodikou nalezeno přibližně 17 % velmi podobných segmentů. Přičemž bylo také zjištěno, že neexistuje žádný větší rozdíl při čtení mapy mezi respondenty, kteří absolvovali alespoň jeden semestr kartografie a ostatními respondenty.

# POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

## Tištěné zdroje:

BIEDERT, Ralf, Georg BUSCHER a Andreas DENGEL. The eyeBook – Using Eye Tracking to Enhance the Reading Experience. *Informatik-Spektrum* [online]. Springer-Verlag, 2010, roč. 3, č. 33, s. 10 [cit. 2013-04-02]. Dostupné z: <http://link.springer.com>

BRODERSEN, Lars, Hans ANDERSEN a Steen WEBER. *Applying eye-movement tracking for the study of map perception and map design* [online]. Kort & Matrikelstyrelsen, 2001. ISBN 87-7866-340-7 [cit. 2013-04-04]. Dostupné z: <http://people.land.aau.dk/~lars/documents>

COHEN, Paul a Niall ADAMS. An Algorithm for Segmenting Categorical Time Series into Meaningful Episodes. In: *Advances in Intelligent Data Analysis: 4th International Conference, IDA 2001 Cascais, Portugal, September 13–15, 2001 Proceedings* [online]. Springer Berlin Heidelberg, 2001, s. 198-207. ISBN 978-3-540-44816-7 [cit. 2013-04-06]. Dostupné z: <http://link.springer.com>

COHEN, Paul, Niall ADAMS a Brent HEERINGA. Voting Experts: An Unsupervised Algorithm for Segmenting Sequences [online]. IOS Press Amsterdam, 2006, s. 607-625. [cit. 2013-04-06]. Dostupné z: <http://citeseerx.ist.psu.edu>

DĚDKOVÁ, Pavla. *3D vizualizace zaniklé obce a její hodnocení z hlediska uživatelské kognice* [online]. Olomouc, 2012 [cit. 2013-04-03]. Dostupné z: <http://geoinformatics.upol.cz>. Bakalářská práce. Univerzita Palackého v Olomouci.

DUCHOWSKI, Andrew. *Eye Tracking Methodology: Theory and Practice* [online]. Springer-Verlag London, 2007. ISBN 978-1-84628-609-4 [cit. 2013-04-04]. Dostupné z: <http://link.springer.com>

GIENKO, Gennady a Eugene LEVIN. *Eye-tracking and augmented photogrammetric technologies*. Baltimore, Maryland: ASPRS 2005 Annual Conference [online]. 2005. [cit. 2013-04-03]. Dostupné z: <ftp://ftp.ecn.purdue.edu>

HOLMQVIST, Kenneth, Marcus NYSTROM a kol. *Eye Tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011. ISBN 978-0-19969708-3.



KOCYAN, Tomáš, Jan MARTINOVIČ, Jiří DVORSKÝ a Václav SNÁŠEL. Czech Text Segmentation Using Voting Experts and Its Comparison with Menzerath-Altmann law. In: *Computer Information Systems – Analysis and Technologies: 10th International Conference, CISIM 2011, Kolkata, India, December 14-16, 2011. Proceedings* [online]. Springer Berlin Heidelberg, 2011, s. 152-160. ISBN 978-3-642-27245-5 [cit. 2013-04-06]. Dostupné z: <http://link.springer.com>

KOCYAN, Tomáš, Jan MARTINOVIČ, Štěpán KUCHAR a Jiří DVORSKÝ. Unsupervised Algorithm for Post-Processing of Roughly Segmented Categorical Time Series. In: *Dateso 2012* [online]. 2012, s. 81-92. Vol. 837. ISBN 978-80-7378-171-2 [cit. 2013-04-07]. Dostupné z: <http://ceur-ws.org>

LI, Xia, Arzu ÇÖLTEKIN a Menno-Jan KRAAK. Visual Exploration of Eye Movement Data Using the Space-Time-Cube. In: *Geographic Information Science: th International Conference, GIScience 2010, Zurich, Switzerland, September 14-17, 2010. Proceedings* [online]. Springer Berlin Heidelberg, 2010, s. 295-309. ISBN 978-3-642-15300-6 [cit. 2013-04-04]. Dostupné z: <http://link.springer.com>

MILLER, Matthew a Alexander STOYTCHEV. Hierarchical Voting Experts: An Unsupervised Algorithm for Hierarchical Sequence Segmentation. *Proceedings of the 7th IEEE International Conference on Development and Learning (ICDL)* [online]. 2008 [cit. 2013-04-06]. Dostupné z: <http://www.ece.iastate.edu>

MÜLLER, Meinard. Dynamic Time Warping. In: *Information Retrieval for Music and Motion* [online]. Springer Berlin Heidelberg, 2007, s. 69-84. ISBN 978-3-540-74048-3 [cit. 2013-04-09]. Dostupné z: <http://link.springer.com>

POPELKA, Stanislav, Alžběta BRYCHTOVÁ a Jan BRUS. Advanced Map Optimalization Based on Eye-Tracking. In: *InTech: Cartography - A Tool for Spatial Analysis* [online]. 2012 [cit. 2013-04-06]. Ed.: Carlos Bateiera. ISBN 978-953-51-0689-0. Dostupné z: <http://www.intechopen.com>

SENIN, Pavel. *Dynamic Time Warping Algorithm Review* [online]. Honolulu, 2008. No. CSDL-08-04 [cit. 2013-04-09]. Dostupné z: <https://csdl-techreports.googlecode.com/svn/trunk/techreports/2008/08-04/08-04.pdf>

TOBII TECHNOLOGY. *Tobii Eye Tracking An introduction to eye tracking and Tobii Eye Trackers* [online]. 2010, 12 s [cit. 2013-04-03]. Dostupné z: <http://www.tobii.com/en/eye-tracking-research/global/library/white-papers/tobii-eye-tracking-white-paper>

## **Internetové zdroje:**

Converting images into time series for data mining. *My Experiments in Truth* [online]. 2011 [cit. 2013-04-12]. Dostupné z: <http://izbicki.me/blog/converting-images-into-time-series-for-data-mining>

Dual Purkinje Eyetrackers. *KU Leuven* [online]. 2011 [cit. 2013-04-04]. Dostupné z: <http://ppw.kuleuven.be/english/lep/resources/purkinje>

Eye Tracking. *Consumer Insights Group* [online]. 2011 [cit. 2013-04-04]. Dostupné z: <http://www.cigresearch.com/what-we-do/eye-tracking>

Mortonův rozklad. In: *Wikipedie: Otevřená encyklopedie* [online]. 2009 [cit. 2013-04-15]. Dostupné z: [http://cs.wikipedia.org/wiki/Morton%C5%AFv\\_rozklad](http://cs.wikipedia.org/wiki/Morton%C5%AFv_rozklad)

## **SUMMARY**

The main aim of the thesis was searching for repeating segments in experimental data which would imply similar ways of solving the problems by different respondents. For this purpose the algorithms working with time series were used, namely the Voting Experts algorithm and the Dynamic Time Warping algorithm. The data was obtained by the eye-tracking method.

One of the minor aims of the thesis was to choose data in a way that would enable further analysis. In the past the Department of Geoinformatics at the Palacký University in Olomouc did some research concerning the differences in the perception between 2D and 3D maps. Some of this data was also used in this thesis.

The program that enabled the implementation of the Voting Experts algorithm as well as the Dynamic Time Warping algorithm was created by Ing. Tomáš Kocyan. Since the program requires the input data to be in 1D format, it was also necessary to create the method that would enable the application of the program over the 2D data obtained from eye-tracking system. For this purpose the scripts were created in the MATLAB, which first convert the data into a suitable form and then make the segmentation of input data. All of such segments were then mutually compared and their similarity henceforth calculated.

# **PŘÍLOHY**

# SEZNAM PŘÍLOH

## Volné přílohy

### Příloha 1 DVD

- Vytvořené skripty (*Příloha 1\_1 LoadVE.m, Příloha 1\_2 Video.m, Příloha1\_3 OutVE.m*)
- Porovnání segmentů (*Příloha 1\_4 Segmenty.pdf, Příloha 1\_5 Histogramy.pdf, Příloha 1\_6 Podobnosti.pdf, Příloha 1\_7 Porovnani.pdf*)
- Ukázkový videozáznam (*Příloha 1\_8 Video.mp4*)
- Text magisterské práce
- Webové stránky o diplomové práci