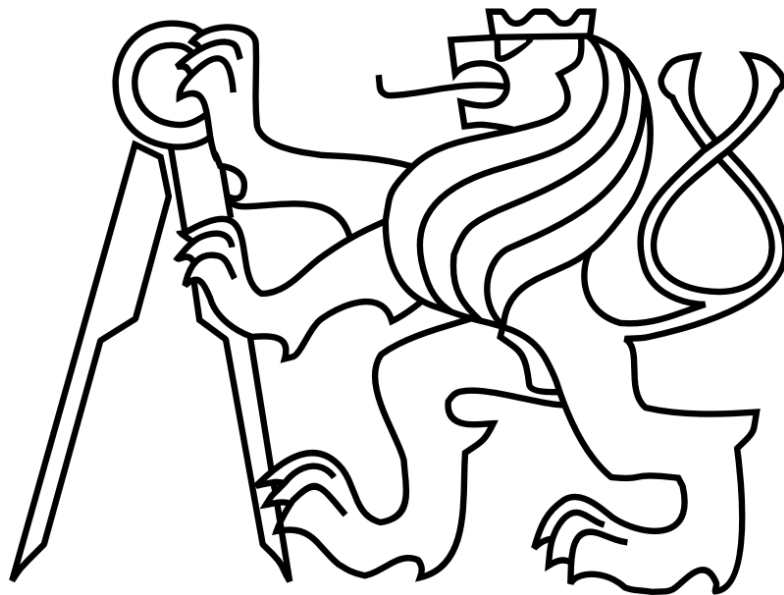


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Diplomová práce



Bc. Michal Zeman

**Aplikace pro vizualizaci a hodnocení
videookulografických dat**

2016

Biomedicínské inženýrství

Vedoucí práce: Ing. Jaromír Doležal, Ph.D

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Michal Z e m a n

Studijní program: Biomedicínské inženýrství a informatika (magisterský)

Obor: Biomedicínské inženýrství

Název tématu: Aplikace pro vizualizaci a hodnocení videookulografických dat

Pokyny pro vypracování:

1. Seznamte se s možnostmi využití sledování očních pohybů pro diagnostiku dyslexie a zpracujte v prostředí Matlab vybraný okulometrický test.
2. Navrhněte modulární architekturu a grafické rozhraní aplikace.
3. Implementujte modul pro načítání a prohlížení dat.
4. Implementujte funkce pro zpracování naměřených dat dle existujícího toolboxu pro Matlab.
5. Implementujte výpočty příznaků pro vybrané okulometrické testy.

Aplikace musí umožňovat prohlížení původních a filtrovaných dat, segmentaci videookulografických dat a zobrazovat parametry specifické k daným okulometrickým testům. Aplikaci implementujte v jazyce C#.

Seznam odborné literatury:

- [1] Jošt, J. (2009). Oční pohyby, čtení a dyslexie. Praha : Fortuna, 176 s., ISBN 978-80-7373-055-0.
- [2] Kowler, E. (2011). Eye movements: The past 25 years. Vision Research, 51, 1457-1483.
- [3] Jošt, J. (2014). Vyhodnocení okulomotorického testu, výzkumná zpráva, 23 s.
- [4] Andreassi, J. L. (2007). Psychophysiology, Human behavior & Physiological responses. Psychological Press, New York. ISBN 978-0-8058-4951-6.

Vedoucí diplomové práce: Ing. Jaromír Doležal, Ph.D.

Platnost zadání: do konce zimního semestru 2016/2017

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 30. 9. 2015

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne.....

.....

Poděkování

Touto cestou bych rád poděkoval vedoucímu mé práce, panu Ing. Jaromíru Doležalovi, Ph. D. za konzultace, čas a morální podporu, které mi v souvislosti s vytvářením mé diplomové práce věnoval. Dále bych chtěl poděkovat všem vyučujícím, kteří mi během mého studia oboru Biomedicínské inženýrství na Fakultě elektrotechnické ČVUT v Praze předali cenné vědomosti a zkušenosti.

Velký vděk patří také mé rodině za podporu během celého studia.

Anotace

Tato práce se zabývá zpracováním očních pohybů a jejich využitím pro diagnostiku dyslexie. Pro zpracování úlohy Pavlidis byl vytvořen nástroj s grafickým uživatelským rozhraním v prostředí Matlab. Tento nástroj byl několikrát použit při studiu očních pohybů a byl integrován do existujícího toolboxu pro Matlab, vyvíjeného na ČVUT (TEMSA toolbox). Ve spolupráci se společností Medicton Group s.r.o. byla vytvořena aplikace pro vizualizaci a hodnocení videookulografických záznamů v jazyce C#. Aplikace umožňuje prohlížení naměřených dat systémem I4Tracking a jejich zpracování. Aplikaci je možné rozšiřovat pomocí samostatných modulů řešících jednotlivé úlohy. V této práci byly vytvořeny moduly pro zpracování úloh Fixační stabilita a Antisakády.

Klíčová slova

Oční pohyby, sledování očních pohybů, videookulografie, dyslexie, Matlab, C#

Annotation

This diploma thesis is concerned with processing of eye tracking data and dyslexia diagnosis. A graphic user interface tool was created for evaluation of the Pavlidis task. This tool was integrated into the existing toolbox for Matlab (TEMSA toolbox) developed in the CTU and it was used in several studies. An application for visualization and evaluation of videoculography data in C# was created in cooperation with Medicton Group s.r.o. company. The application allows visualization and processing of data recorded by the I4Tracking eye tracking system. The application can be extended by independent modules, each dealing with specific task. Modules for evaluation of Fixation stability and Antisaccades tasks were created.

Key words

Eye movements, eye tracking, videoculography, dyslexia, Matlab, C#

Obsah

1. Úvod	1
2. Současný stav poznání	2
2.1. Klasifikace očních pohybů	2
2.2. Dyslexie	3
2.3. Videookulografické trackery	4
2.4. Oční trackery vyvíjené ve světě	6
2.5. I4Tracking	8
2.6. TEMSA toolbox pro Matlab	10
3. Teoretická část	11
3.1. Testy dyslexie	11
3.1.1. Úloha Pavlidis	11
3.1.2. Úloha Fixační stabilita	13
3.1.3. Úloha Antisakády	14
3.2. Aplikace	14
3.2.1. Popis dat z I4Tracking	15
3.2.2. Programovací jazyk C# a WPF	17
3.2.3. Výběr knihovny pro kreslení grafů	19
3.2.3.1. Knihovny s placenou licenci	19
3.2.3.2. Knihovny s volnou licenci	19
3.2.3.3. Knihovna Oxyplot	19
4. Cíle práce	21
5. Praktická část	23
5.1. Nástroj Matlab	23
5.1.1. Grafické uživatelské rozhraní nástroje	23
5.1.2. Popis zpracování úlohy	24
5.1.3. Výsledky zpracování	28
5.2. Aplikace	29
5.2.1. Grafické uživatelské rozhraní aplikace	29
5.2.1.1. Úvodní stránka	30
5.2.1.2. Prohlížení záznamu v XY grafu	30

5.2.1.3.	Prohlížení záznamu v rozděleném grafu	33
5.2.1.4.	Zobrazení úhlové rychlosti a velikosti pupily	34
5.2.2.	Modularita aplikace	36
5.2.2.1.	Vytvoření nového modulu	36
5.2.3.	Modul pro zpracování úlohy Fixační stabilita	37
5.2.4.	Modul pro zpracování úlohy Antisakády	38
5.2.5.	Možnosti dalšího vývoje aplikace I4Viewer	39
6.	Závěr	40

Seznam tabulek

Tabulka 1 - Klasifikace očních pohybů, tabulka převzata z [3]	2
Tabulka 2 - Struktura extractors (výsledek zpracování úlohy Pavlidis)	27

Seznam obrázků

Obrázek 1 - Schéma očního trackeru (obrázek převzatý z [6])	5
Obrázek 2 - Odraz světla od lidského oka (Purkyněho obrazy, obrázek převzatý z [6])	6
Obrázek 3 - Výrobky SMI (obrázky převzaty z [8])	6
Obrázek 4 - Výrobky firmy Tobii Pro (obrázky převzaty z [9])	7
Obrázek 5 - Tracker od společnosti The Eye Tribe (obrázek převzat z [10])	7
Obrázek 6 - Tracker od společnosti SR Research (obrázek převzatý z [11])	8
Obrázek 7 - Tracker umístěný na brýlích (starší verze)	9
Obrázek 8 - Tracker umístěný pod monitorem (novější verze)	9
Obrázek 9 - Terčíky pro školní děti	12
Obrázek 10 - Průběh očních pohybů po horizontální ose v úloze Pavlidis (obrázek převzatý z [5])	12
Obrázek 11 - Úloha fixační stability (obrázek převzatý z [17])	13
Obrázek 12 - Antisakadická úloha (obrázek převzatý z [5])	14
Obrázek 13 - Schéma XML souboru rec.xml	17
Obrázek 14 - Příklad grafu pomocí Oxyplot knihovny	20
Obrázek 15 - Modul Pavlidis (GUI hlavní okno - první krok)	24
Obrázek 16 - Modul Pavlidis (GUI hlavní okno - druhý krok)	25
Obrázek 17 - Špatný průchod úlohou Pavlidis	26
Obrázek 18 - Modul Pavlidis (GUI hlavní okno – třetí krok)	26
Obrázek 19 - Špatný průchod úlohou Pavlidis (ideální průběh)	27
Obrázek 20 - Zpracování textu modulem Pavlidis	28
Obrázek 21 - Porovnání průběhů v úloze Pavlidis u dítěte s diagnózou dyslexie (11 let, nahoře) a zdravého dítěte (10 let, dole), obrázek převzatý z [5]	29
Obrázek 22 - Aplikace I4Viewer: zobrazení XY grafem	31
Obrázek 23 - Aplikace I4Viewer: zobrazení XY grafem (časové omezení výběru)	32
Obrázek 24 - Aplikace I4Viewer: zobrazení XY grafem (zobrazení fixací a sakád)	33
Obrázek 25 - Aplikace I4Viewer: zobrazení v rozděleném grafu	34
Obrázek 26 - Aplikace I4Viewer: zobrazení úhlové rychlosti	35
Obrázek 27 - Aplikace I4Viewer: zobrazení plochy pupil	35
Obrázek 28 - Modul pro zpracování úlohy Fixační stabilita	38
Obrázek 29 - Modul pro zpracování úlohy Antisakády	39

Kapitola 1

Úvod

Videookulografické metody záznamu očních pohybů se v posledních letech stávají velmi dobrým diagnostickým prostředkem. Jedním z mnoha využití studia očních pohybů je i diagnóza specifických vývojových poruch učení, zejména dyslexie. Tato vrozená porucha mozku je pro mnoho lidí na celém světě velkou překážkou v každodenním životě. Dyslexie se často projeví až ve škole na základě špatného prospěchu a problémům se čtením. Videookulografické metody záznamu očních pohybů nemusí být nutně spojeny s textovými stimuly a je možné je využít v diagnóze dyslexie i u předškolních dětí, které ještě neumí číst. U dětí, kde by se dyslexie poznala ještě dříve, než začnou číst a psát, se může dostat odborné péče a předejít tak nepříznivým pochodům (kognitivním, emocionálním, socializačním), které provázejí dyslektiky po celou dobu studia i později. Velkou výhodou těchto metod je, že vyšetření pomocí nich je neinvazivní a nijak nepříjemné [1].

Tato práce se zabývá videookulografickými metodami zpracování očních pohybů a jejich použití v diagnóze dyslexie. V rámci této práce je vyvinut nástroj pro zpracování videookulometrického testu Pavlidis a samostatná aplikace pro vizualizaci a hodnocení videookulografických dat. Tato aplikace (s pracovním názvem I4Viewer) je vyvíjena ve spolupráci se společností Medicton Group s.r.o. a v budoucnosti by se měla stát součástí kompletního balíku software a hardware pro diagnostiku očních pohybů. Metody použité v této aplikaci jsou založeny na základě existujícího toolboxu pro Matlab (TEMSA toolbox), vyvíjeného na ČVUT.

Kapitola 2

Současný stav poznání

Výzkum očních pohybů probíhá již více než 50 let. Velké pokroky ve výzkumu očních pohybů se dějí však hlavně v posledních dvaceti letech. To je především díky inovacím v oblasti digitálních videokamer, které téměř nahradily ostatní způsoby záznamu očních pohybů. Existuje velké množství studií, které potvrzují souvislost očních pohybů s některými velmi vážnými nemocemi.

2.1. Klasifikace očních pohybů

Pohyby lidského oka můžeme klasifikovat do několika typů. Nejčastější typy očních pohybů a pro ně typické hodnoty trvání, amplitudy a rychlosti jsou uvedeny v Tabulka 1. Je pravděpodobné, že existují další typy očních pohybů, které zatím nebyly výzkumníky objevené [2].

Typ	Trvání [ms]	Amplituda	Rychlost
Sakáda	30 - 80	4 - 20°	30 - 500°/s
Fixace	200 - 300	-	-
Plynulé oční pohyby	-	-	10 - 30°/s
Mikrosakáda	10 - 30	10 - 40'	15 - 50°/s
Tremor	-	< 1'	< 20'/s
Drift	200 - 1000	1 - 60'	6 - 25'/s

Tabulka 1 - Klasifikace očních pohybů, tabulka převzata z [3]

Sakády jsou rychlé, simultánní pohyby obou očí ve stejném směru od jedné fixace k další fixaci. Dají se snadno identifikovat právě podle rychlosti těchto očních pohybů. Fixace nastává, když oko zastaví pohyb na určitý moment a spíše se nehýbe. Plynulé oční jsou, pokud oko sleduje nějaký pomalu se pohybující stimul. Tyto pohyby řídí jiná část mozku než sakády. Při fixacích se objevují další tři typy velmi malých pohybů. Mikrosakády jsou velmi krátké pohyby očí, které vrací pohled do středu fixací (často z důvodu driftu). Tremor je velmi malý třes pohledu o frekvenci přibližně 90Hz, jehož úloha není stále důkladně prozkoumána. Drift je pohyb unášející pohled ze středu fixace [3].

Pro detekci mikrosakád, tremoru a driftu je zapotřebí tracker s velmi vysokou přesností a vzorkovací frekvencí. Pokud v úloze postačí detekce fixací a sakád, je možné použít tracker s rozumně velkou snímkovací frekvencí (používají se trackery s frekvencí od 50Hz do 250Hz) a metoda prahování úhlové rychlosti:

$$\begin{aligned} \text{úhlová rychlost očního pohybu} &\geq \text{práh} \dots \text{ Sakáda} \\ \text{úhlová rychlost očního pohybu} &< \text{práh} \dots \text{ Fixace} \end{aligned}$$

2.2. Dyslexie

Mezinárodní norma klasifikace nemocí ICD-10 (International Classification of Diseases 10th Revision) klasifikuje dyslexii pod kódem F 81.0. Hlavním rysem je specifické a výrazné narušení vývoje dovednosti ve čtení, které nelze přičítat mentálnímu věku, problémům zrakové ostrosti nebo neadekvátnímu způsobu vyučování. Může být postiženo chápání čteného textu, schopnost poznávat čtená slova, čtení nahlas a výkon při úkolech, které vyžadují čtení. Ke specifické poruše čtení jsou často přidruženy potíže se psáním, které často přetrvávají do adolescence, i když ve čtení dochází k určitému pokroku. V anamnéze dětí se specifikovanou poruchou čtení se často vyskytují specifické vývojové poruchy řeči a mluvy a v běžné řeči lze mnohdy odhalit drobné potíže [4] [1].

Aby se dalo mluvit o dyslexii, musí být čtenářský výkon podstatně nižší než jeho očekávaná úroveň, která je dána chronologickým věkem, naměřenou inteligencí (IQ) a přiměřenou vzdělávací nabídkou. Zavádí se pojem diskrepance, to je rozdíl IQ – čtenářský výkon. Pro diagnostiku dyslexie se uvádí, že diskrepance je minimálně v hodnotě dvou standardních odchylek. Podrobná diagnostická kritéria a charakteristiku dyslexie uvádí například Diagnostický a statistický manuál Americké psychiatrické asociace DSM-IV-TR pod kódem č. 315.00 [1].

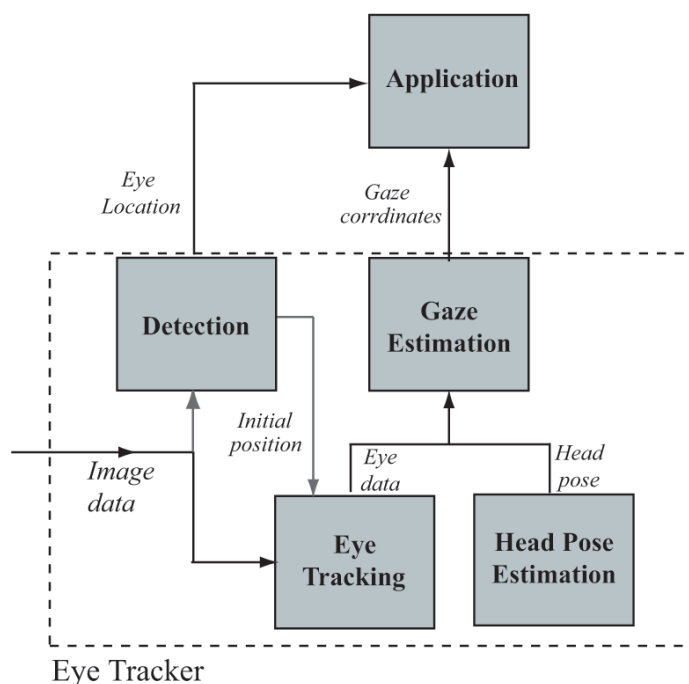
Postižený jedinec své problémy bez cizí pomoci nezvládne a porucha závažným způsobem zasahuje do jeho života, počátku školního a posléze i pracovního a významně ho

omezuje v uplatnění jeho nadání a tvořivosti. Lidé, kteří tuto základní dovednost plně nezvládnou, se tak dostávají do handicapovaného postavení se všemi nepříznivými následky, ať už společenskými (ztrácíme kus jejich tvůrčího potenciálu), či osobními (pokles osobního štěstí a spokojenosti). Z tohoto pohledu je proto nanejvýš záslužné porozumět podmínkám zdravého čtenářského vývoje a jeho nejvýznamnější odchylce - dyslexii, provést včasnou diagnózu a následně i intervenci. Včasnou diagnózou a včasnou nápravou se rozumí období před nástupem dítěte do školy, tedy ještě dříve než se nekontrolovaně rozeběhnou všechny nepříznivé pochody (kognitivní, emocionální, socializační), které provázejí dyslektika ve věku školním i později [5].

2.3. Videookulografické trackery

Základem každého videookulografického trackeru je jedna či více videokamer nasměřovaných na oko. Používají se od nejjednodušších kamer typu webkamera, či kamera v mobilním telefonu, po vysokorychlostní kamery nahrávající v infračerveném (IR) pásmu a vysokém rozlišení. Přímý vliv na přesnost vyhodnocení záznamů má rozlišení kamery a velikost oka v záběru. Používají se kamery, umístěné buď na hlavě, nebo statické (např. pod monitorem). V aplikacích, kde chceme detekovat sakády a mikrosakády je potřeba vyšší vzorkovací frekvence kamery (viz 2.1). Cílem očního trackeru je z nasnímaných obrázků interpretovat směr pohledu oka.

Aby z nasnímaných obrázků bylo možné získat směr pohledu, je zapotřebí nejdříve detekovat oko v nasnímaných obrázcích a porovnat jeho pozici s referenčními pozicemi v závislosti na poloze hlavy (viz schéma na Obrázek 1).

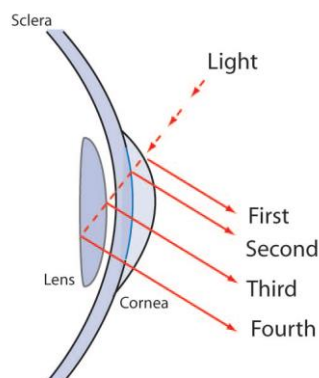


Obrázek 1 - Schéma očního trackeru (obrázek převzatý z [6])

Techniky detekce oka z obrázku se zakládají na specifických geometrických a fotometrických vlastnostech lidského oka, dají se klasifikovat jako detekce na základě tvaru, rysů, podobnosti, či jejich kombinací [6].

Pro získání referenčních pozic oka je nutné provést kalibraci trackeru. V předem určených pozicích se zobrazují body, na které se oko fixuje. Používají se nejčastěji dvou pěti, či devíti bodové kalibrace. Koordináty pohledu se poté počítají pomocí polynomiálních metod, či použitím umělé neuronové sítě, nebo regresí podpůrných vektorů (support vector regression). V úvahu se při výpočtech bere i pozice hlavy. Na přesnost kalibrace má významný podíl doba trvání zobrazení bodů, také počet a rozložení zobrazovaných bodů [7].

Pohyby hlavou při snímání jak statickou kamerou, tak kamerou umístěnou na hlavě, je nutné při výpočtech kompenzovat. Toto zvládají pouze některé trackery, u ostatních je nutné nehýbat hlavou, či dokonce hlavu pevně fixovat. Kompenzace se velmi často provádí snímáním odrazu od rohovky (corneal reflection). Posvítíme-li zdrojem světla (velmi často se používá infračervený zdroj světla, není totiž vidět a neruší při experimentu) do oka, část světla projde jednotlivými strukturami oka a část se od nich odrazí. Vznikají tzv. Purkyněho obrazy (viz Obrázek 2), první obraz je právě odraz od rohovky, často nazýván jako odlesk (anglicky glint). Máme-li jeden či více stacionárních zdrojů světla, můžeme právě díky těmto odrazům vypočítat pohyby hlavy [6].



Obrázek 2 - Odraz světla od lidského oka (Purkyněho obrazy, obrázek převzatý z [6])

2.4. Oční trackery vyvíjené ve světě

V současné době je na trhu několik výrobců vyvíjejících zařízení pro sledování očních pohybů. Oficiální podporu v ČR zatím nemá žádná společnost vyvíjející trackery očních pohybů kromě I4Tracking (viz 2.5), ale v ČR jsou firmy, které používají trackery zahraničních společností a nabízejí služby měření očních pohybů.

Německá společnost SMI (SensoMotoric Instruments) vyrábí několik druhů očních trackerů (Obrázek 3). Mezi nejzajímavější patří SMI RED500, což je tracker pro snímání očních pohybů na obrazovce počítače. Je umístěn pod monitorem a nahrává oční pohyby frekvencí 500Hz, sleduje obě oči a umožňuje volný pohyb hlavy. Zajímavý je také tracker umístěný do zařízení Oculus Rift, umožňující další možnosti pro vývojáře aplikací pro virtuální realitu. SMI vyrábějí i bezdrátové brýle se zabudovaným trackerem (frekvence 60Hz), ty jsou vhodné například pro marketingové účely. K trackerům poskytují vlastní vyhodnocovací software, či případně SDK pro vývojáře dalších aplikací [8].



Obrázek 3 - Výrobky SMI (obrázky převzaty z [8])

Velmi podobné výrobky jako má firma SMI, nabízí i švédská společnost Tobii Pro (Obrázek 4). Má také statické oční trackery pod monitor s nahrávacími frekvencemi od 30Hz do 300Hz, stejně tak tracker do Oculus Riftu a brýle se zabudovaným trackerem (frekvence

2. SOUČASNÝ STAV POZNÁNÍ

50Hz nebo 100Hz). Nabízí vlastní vyhodnocovací software i SDK pro vývojáře. Stejně jako firma SMI na svých stránkách nikde neuvádí ceny trackerů a poskytovaných služeb. Měření pomocí kamery Tobii X120 umožňuje v České republice například firma QuickINSIGHT s.r.o. [9].



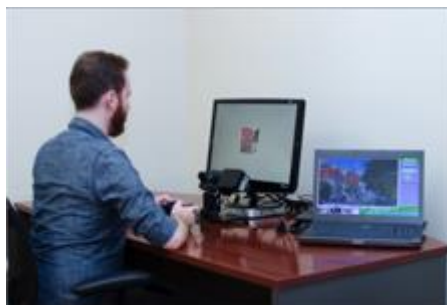
Obrázek 4 - Výrobky firmy Tobii Pro (obrázky převzaty z [9])

Zajímavým výrobkem je oční tracker od dánské firmy The Eye Tribe (Obrázek 5). Nejzajímavější je na něm hlavně cena – 99 USD včetně software a SDK pro vývojáře. Tracker pracuje na frekvenci 30Hz nebo 60Hz a je možné jej připojit i k zařízení s Android nebo Windows Mobile. Snímá pohyby obou očí a umožňuje pohyby hlavy. Tracker také získal ocenění z prestižního veletrhu CES 2015 za nejlepší inovace [10].



Obrázek 5 - Tracker od společnosti The Eye Tribe (obrázek převzat z [10])

Kanadská společnost SR Research nabízí oční tracker s vzorkovací frekvencí 1000Hz v módu sledování obou očí, či frekvencí 2000Hz v módu sledování jednoho oka (Obrázek 6). To je v současné době tracker s nejvyšší snímkovací frekvencí na trhu. Jedná se o statický tracker umístěný pod monitorem. S trackerem poskytuje vlastní vyhodnocovací systém a SKD pro vývojáře. Bohužel na jejich webových stránkách není uvedena cena [11].



Obrázek 6 - Tracker od společnosti SR Research (obrázek převzatý z [11])

Za zmínku také stojí software od litevské firmy Neurotechnology nazývaný se SentiGaze SDK, který promění kterýkoliv počítač s webkamerou na oční tracker. Cena tohoto software je 890 Euro a na webových stránkách je 30-ti denní trial verze [12].

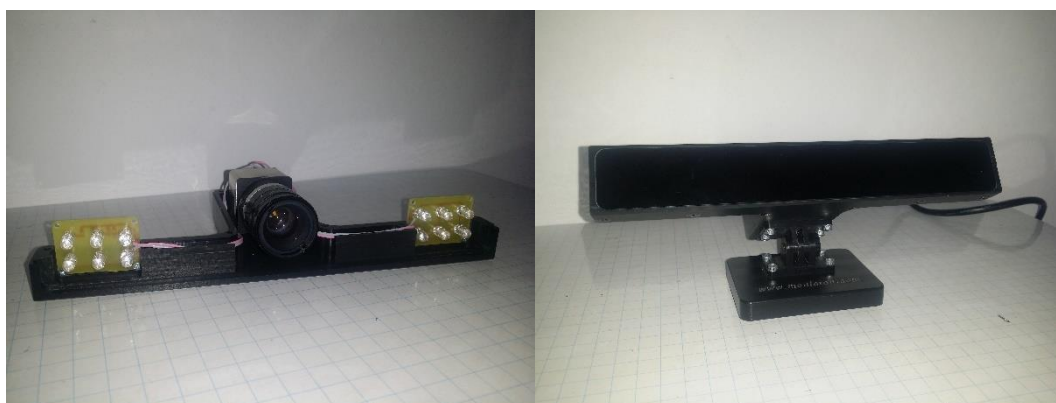
Vzrůstající zájem o sledování očních pohybů také dokazuje zájem „velkých hráčů“ v podobě Samsungu a Google. Samsung používá ve svých modelech řady S funkci Smart Stay, která detekuje, zda uživatel pozoruje displej telefonu a zabraňuje usnutí displeje [13]. Google si nechal patentovat systém sledování očních pohybů umístěný na zařízení na hlavě (jako je např. Google Glass) sledující oční reakce na vnější podněty (jako jsou např. reklamy). To jestli svůj patent dovede do reality není jisté, každopádně to však dokazuje vzrůstající zájem o sledování očních pohybů [14] [15].

2.5. I4Tracking

Program I4Tracking vyvíjený společností Medicton Group s.r.o. je software určený k záznamu očních pohybů na monitoru počítače. Pomocí dodávaného hardware snímá oční pupilu videookulografickou metodou, tedy pomocí vysokorychlostní infračervené kamery (více než 150 snímků za vteřinu) umístěné na brýlích – starší verze trackeru (Obrázek 7) nebo monitoru – novější verze (Obrázek 8). Tracker je přizpůsoben pro snímání očních pohybů na obrazovce monitoru.



Obrázek 7 - Tracker umístěný na brýlích (starší verze)



Obrázek 8 - Tracker umístěný pod monitorem (novější verze)

Tracker používá 5t-i, či 9-ti bodovou kalibraci a umožňuje zobrazovat statické (fotografie), či dynamické (video) stimuly a nahrávat reakce na tyto stimuly. U starší verze trackeru je velmi vhodné omezit pohyby hlavy, ty vnášejí do nahraných dat nežádoucí artefakty. U některých jedinců (hyperaktivní děti, dyslektici, atd.) se doporučuje fixovat hlavu, aby se co možná nejvíce zamezilo těmto artefaktům. Nová verze trackeru řeší vznik těchto artefaktů kompenzací pohybů hlavy za pomoci sledování odrazu infračervených LED diod od rohovky oka (viz 2.3).

Nasnímané obrázky se ukládají jako surová a nezkomprimovaná data (to je z důvodu zachování informací) do souboru „raw_images.dat“. Po ukončení nahrávání dat se spustí vyhodnocování dat, které z nahraných obrázků vyextrahuje pozice očních pohybů na monitoru a mnoho dalších parametrů (více o snímaných parametrech v 3.2.1). Vše se uloží ve formátu XML do souboru „rec.xml“. Záznamy o parametrech experimentu se také uloží v XML formátu do souboru „exp.xml“.

2.6. TEMSA toolbox pro Matlab

TEMSA toolbox byl vytvořen pro zpracování nahraných dat z očního trackeru na ČVUT jako sbírka funkcí pro Matlab. V současné době obsahuje velké množství metod pro zpracování, filtraci, úpravu a klasifikaci signálů očních pohybů. Tento toolbox beru jako mé know-how při vytváření aplikace pro vizualizaci a hodnocení videookulografických dat I4Viewer.

Data z programu I4Tracking (viz 3.2.1) jsou nejdříve přetransformována do formátu TEMSA, který je typu Matlab Struct a lépe se s ním pracuje v prostředí Matlab. Tato data je dále možné pomocí připravených funkcí modifikovat (filtrace, extrakce artefaktů, atd.), dopočítat další data (rychlost pohybu, zrychlení, atd.), či získat příznaky pro předpřipravené úlohy. Část metod se zaměřuje na zpracování úloh nasnímaných v projektu TAČR, který se zabývá analýzou dyslexie u dětí školního a předškolního věku a nalezení signifikantních rozdílů mezi skupinou dětí s dyslexií a kontrolní skupinou pro možnost predikce dyslexie u předškolních dětí.

Kapitola 3

Teoretická část

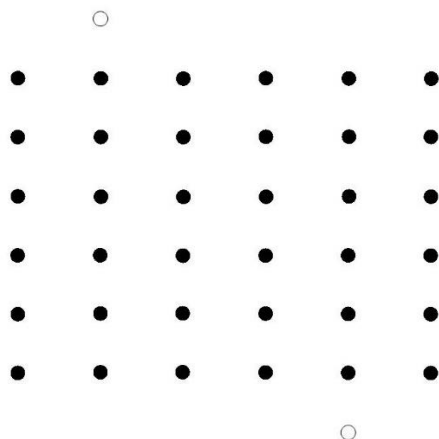
3.1. Testy dyslexie

Pro prostudování problematiky zpracování očních pohybů a získání lepší znalosti naměřených dat jsem se seznámil se zpracováním okulometrického testu – úloha Pavlidis (viz 3.1.1). Tuto úlohu použil poprvé George Th. Pavlidis v roce 1985 [16]. Je zejména vhodná pro předškolní a školní děti, protože nevyžaduje schopnost čtení. Pro tuto úlohu jsem navrhl modul v prostředí Matlab sloužící k hromadnému zpracování záznamů této úlohy (viz 5.1).

Abych prokázal funkčnost aplikace I4Viewer navrhl jsem moduly pro analýzu úlohy Fixační stabilita (viz 3.1.2) a úlohy Antisakády (viz 3.1.3), které již předtím byl zpracovány pomocí TEMSA toolboxu pro Matlab (viz 5.2.3 a 5.2.4).

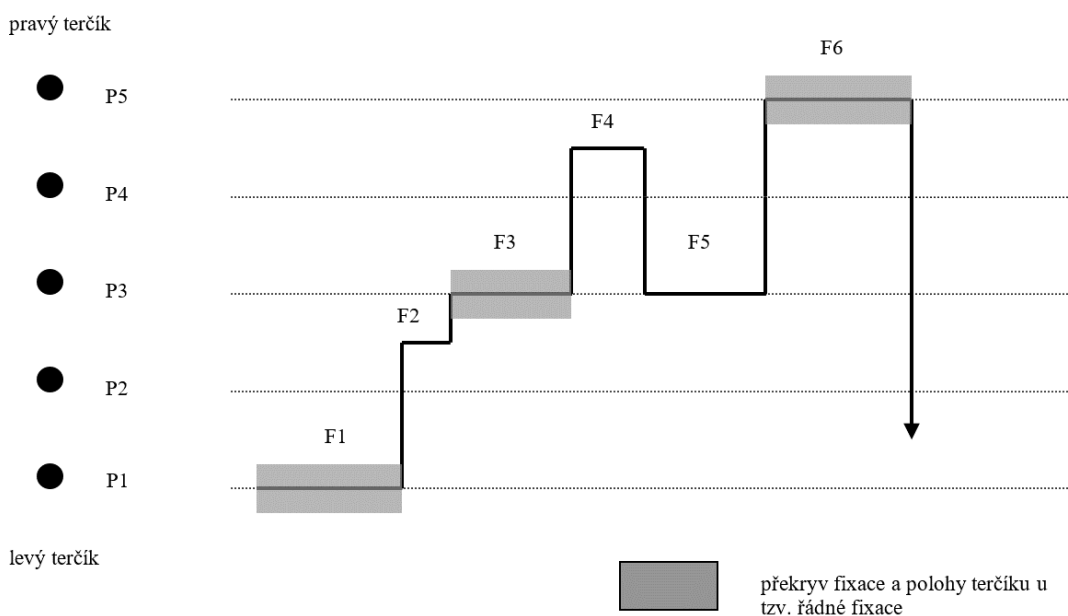
3.1.1. Úloha Pavlidis

Na obrazovce naskočí 6 řádek terčků po šesti terčících v každé řádce (viz Obrázek 9). Dítěti řekneme: sleduj každý puntík zleva doprava a shora dolů. Na úlohu ukazujeme prstem, skáče z jednoho terčiku na druhý a slovně doprovázíme (hop a hop...), až dojdeme na konec řádky a přeskočíme prstem na začátek druhé řádky a začneme znovu. Dítě musí pochopit směr sledování. Nesmí si ovšem pomáhat prstem - k tomu mají některé děti sklón a my to dítěti hned v instrukci zakážeme [17].



Obrázek 9 - Terčíky pro školní děti

V této úloze je vyšetřována poloha v horizontální ose (osa X) a ve vertikální ose (osa Y) v závislosti na čase. Osa Y znázorňuje řádky a slouží spíše pro kontrolu, zda byla úloha řádně provedena. Pokud jsou vidět změny na ose Y jindy než na konci řádku je pravděpodobné, že dítě přestalo řádně plnit úlohu a dále zpracovávat data této úlohy nemá smysl. Hlavní význam ve vyšetření této úlohy má osa X, na které je vidět jak se dítě postupně dívalo na jednotlivé terčíky. Možný průběh je možné vidět na následujícím obrázku (Obrázek 10).



Obrázek 10 - Průběh očních pohybů po horizontální ose v úloze Pavlidis (obrázek převzatý z [5])

Černé tečky značí řadu terčů (v reálu uspořádanou vodorovně). P1 až P5 značí polohy terčů a úhlové vzdálenosti mezi nimi. F1 až F6 značí fixace; délka jejich čar značí dobu trvání. Překryv fixace a polohy terčů (šedé plochy) značí tzv. řádnou (regulérní) fixaci. Ve fixacích F2 a F4 se dítě dívá mimo terč, v F5 se vrací do již jednou fixovaného terčů [5].

Při zpracování této úlohy je nutné spočítat počty fixací na každém řádku (ideálně pět regulérních fixací), zjistit jak dlouhé byly všechny fixace (ideálně stejně dlouhé) a spočítat odchylku od ideálního průběhu (průběh s pěti stejně dlouhými regulérními fixacemi). Také se ukázalo, že pro statistické vyhodnocení je důležitý počet „zpětných“ fixací [18], tedy fixací na bod, na který již dítě jednou fixovalo (fixace F5).

3.1.2. Úloha Fixační stabilita

Dítě fixuje terč, který naskočí na obrazovce. Dítěti řekneme: dívej se na tuhle tečku a nespouštěj z ní oči. Nejprve varianta uniformní: tečka je černá a pozadí bílé (či světle šedé). Doba úlohy: 10 s. Poté varianta optokinetická, stimulující optokinetický nystagmus a znesnadňující fixaci, viz obrázek 1. Doba úlohy opět 10 s. Přejít z jedné varianty do druhé bude plynulý: s nástupem jedenácté vteřiny naskočí na obrazovce během jedné - dvou vteřin svislé světlé a tmavě šedé pruhy, které se začnou ihned při svém zjevení pohybovat ve směru zprava doleva. Rychlost pohybu $5^\circ/s$. Po skončení naskočí prázdná obrazovka [17].

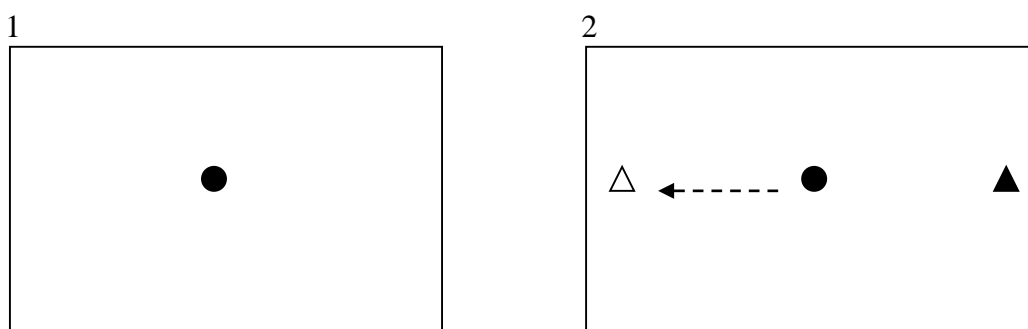


Obrázek 11 - Úloha fixační stability (obrázek převzatý z [17])

V ideálním případě dítě celou dobu fixuje na uvedený bod. U této úlohy je potřebné zjistit, zda se dítě odklonilo od fixovaného bodu, nerozlišuje se směr pohledu. Počítá se počet odchylek, celková doba odchylek, statistický rozptyl a odchylka od ideálního průběhu. Pro ideální průběh platí: souhrn odchylek reálného průběhu očních pohybů v čase a prostoru se rovná nule [5].

3.1.3. Úloha Antisakády

Úloha předkládá vyšetřované osobě také vizuální podnět na jedné straně vizuálního pole, avšak požaduje, aby se vyšetřovaná osoba podívala na stranu opačnou. Vyšetřovaná osoba fixuje nejprve bod ve středu obrazovky a poté, jakmile se objeví černý trojúhelník, tj. sakadický podnět, vykoná sakadický pohyb (je vyznačen šipkou) na opačnou stranu zhruba do místa, které leží stejně daleko od fixačního bodu jako černý trojúhelník; toto místo je symbolicky označeno bílým trojúhelníkem, který se samozřejmě na obrazovce neobjeví [5].



Obrázek 12 - Antisakadická úloha (obrázek převzatý z [5])

Sakadická reakce na opačnou stranu se nazývá antisakadická či antisakáda, reakce směrem k vizuálnímu sakadickému podnětu se nazývá prosakadická či prosakáda. Prosakáda v antisakadické úloze je hodnocena jako chyba. Lidé si většinou chybu, tj. prosakadickou reakci v anti-úloze, neuvědomují. Průměrný reakční čas antisakád je delší než u prosakád. Sakadický podnět totiž vyvolává reflexní odezvu, tj. náš mozek začne automaticky připravovat prosakádu, a ta musí být naší vůlí potlačena. Teprve pak je možné sakádu přesměrovat na opačnou stranu. Antisakády jsou proto časově náročnější. Antisakády obsahují voluntární komponentu, zatímco prosakády jsou kontrolovány reflexně. Oba druhy sakád jsou řízeny z odlišných míst mozku [5]. Vyšetřuje se směr antisakadického pohybu v horizontální ose, počet chyb a počet opravených chyb.

3.2. Aplikace

Tato kapitola se zabývá teoretickou částí spojenou s vývojem aplikace pro vizualizaci a hodnocení videookulografických dat. Je zde popsána struktura přijímaných dat (výstup z aplikace I4Tracking) a programovací jazyk C# s grafickým subsystémem WPF (Windows Presentation Foundation). Jelikož velkou část aplikace tvoří grafy a bylo mi povoleno použití knihoven pro kreslení grafů, zabývá se tato kapitola i výběrem knihovny pro kreslení grafů.

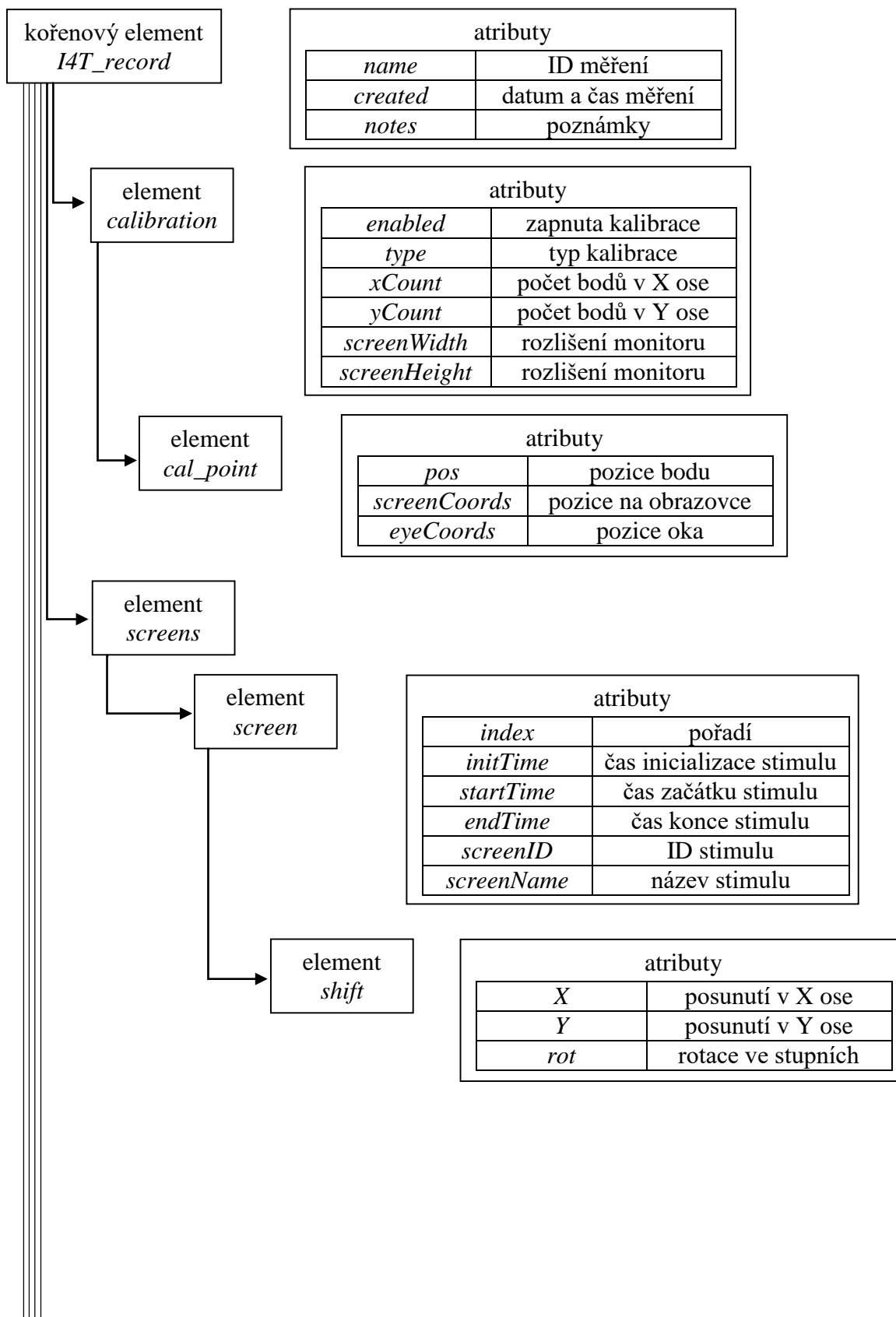
3.2.1. Popis dat z I4Tracking

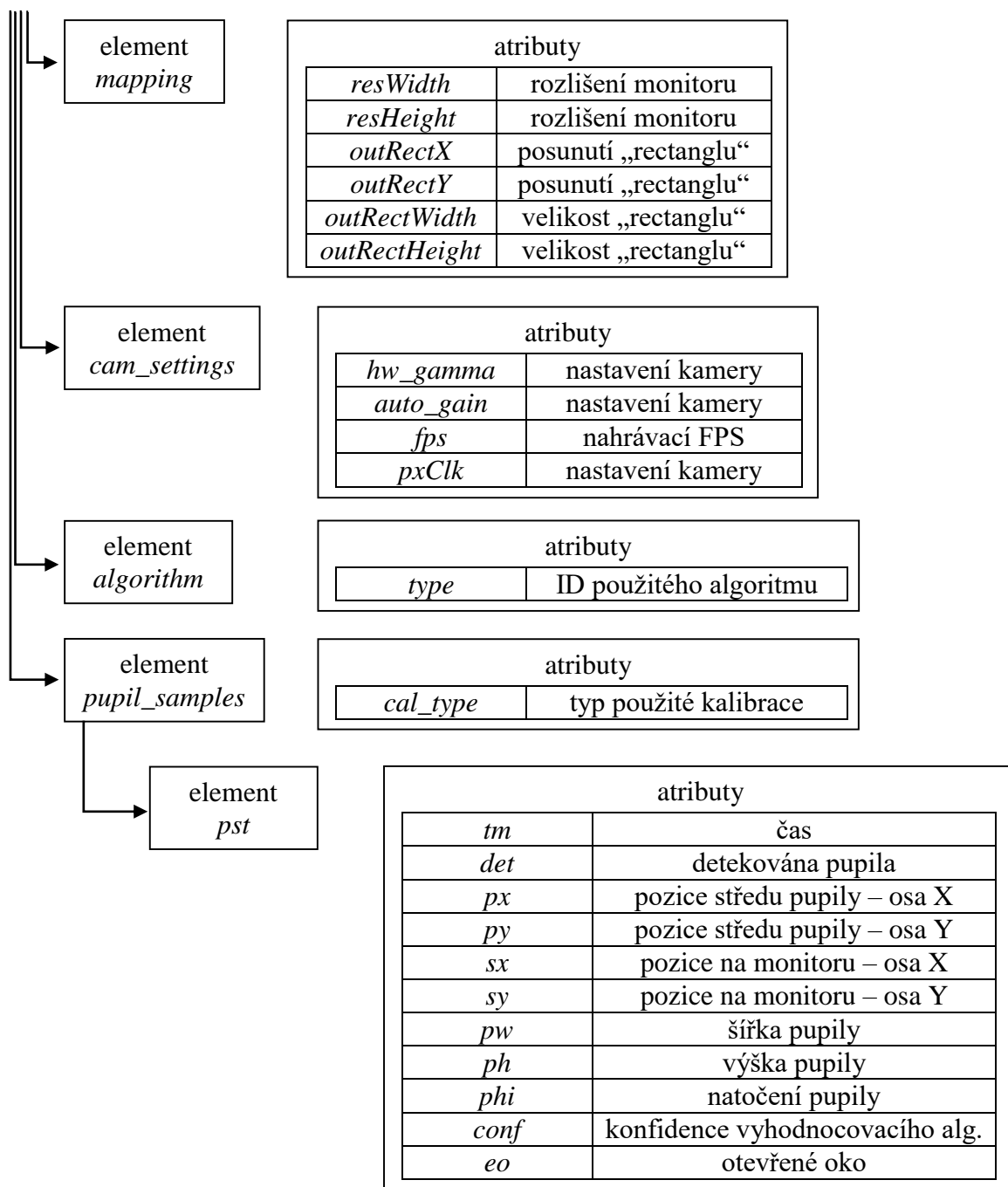
Program I4Tracking, který vyvíjí společnost Medicton Group s.r.o., je jak pro TEMSA toolbox, tak pro aplikaci I4Viewer zdrojem naměřených dat. Před začátkem snímání očních pohybů je potřeba v aplikaci I4Tracking vytvořit experiment a vložit do něj stimuly. Stimuly mohou být jak statické (jeden či více obrázků na monitoru) tak dynamické (video). Poté je možné za pomoci poskytnutého hardware snímat oční pohyby.

Nasnímané obrázky oka jsou aplikací zpracovány a zpracovaná data jsou uložena v XML formátu v souborech „rec.xml“ a „exp.xml“. Zatímco data v souboru „rec.xml“ patří pouze k jednomu náměru, data v souboru „exp.xml“ jsou společná pro celý experiment. Nasnímané obrázky jsou zároveň uloženy v surovém nezkomprimovaném formátu v souboru „raw_images.dat“.

Pro zobrazení nasnímaných dat jsou nejdůležitější hodnoty z elementů *pst* ze souboru „rec.xml“ (viz Obrázek 13). Obsahují pozice očních pohybů na monitoru v přesně zaznamenaných časových okamžicích. Také obsahují údaje o velikosti pupily v každém snímku. Elementy *pst* nejsou rozdělené podle jednotlivých částí experimentu. Pokud je tedy experiment složen z více stimulů, tak je nutné přiřadit jednotlivé elementy *pst* k odpovídajícímu stimulu. To je možné podle elementu *screen*, ve kterém je zaznamenán přesný čas, kdy začal a skončil každý stimul experimentu.

Údaje o stimulech jsou uloženy v souboru „exp.xml“, slouží především pokud chceme zobrazit stimuly jako pozadí pod naměřené oční pohyby. V elementu *image* je atribut *file_name*, který označuje cestu k souboru se stimulem. Pokud je v jednom stimulu více obrázků, je důležitý element *image_rectangle*, ve kterém je uložena pozice a velikost každého obrázku. V ostatních elementech jsou uloženy další atributy experimentu, jako například hudba doprovázející stimul, barva pozadí, předvolby zpracování experimentu a další.





Obrázek 13 - Schéma XML souboru rec.xml

3.2.2. Programovací jazyk C# a WPF

C# je univerzální, typově bezpečný, objektově univerzální programovací jazyk vyvíjený společností Microsoft. Jako hlavní cíl si bere programátorskou produktivitu. Jazyk velmi dobře balancuje mezi jednoduchostí, výmluvností a výkonem. Hlavním architektem

tohoto jazyka je Anders Hejlsberg (tvůrce Turbo Pascalu a architekt Delphi). Jazyk C# je platformně neutrální, ale byl psán tak, aby velmi dobře spolupracoval s Windows .NET Frameworkem [19].

Jazyk C# podporuje objektově orientované paradigmaty, jako je enkapsulace, dědičnost a polymorfismus. Enkapsulace znamená, že okolo objektů se vytvářejí hranice, které přesně oddělují jeho vnější chování (public) od vnitřních implementačních detailů (private). Dědičností se v objektově orientovaném programování míní způsob jak stanovit vztahy mezi objekty. Objekty jsou definované třídami a mohou dědit atributy a chování od již existujících tříd, takovým třídám se říká rodičovské třídy. Polymorfismus je vlastnost, která umožňuje jednomu objektu volat jednu metodu s různými parametry, objektům odvozených z různých tříd volat tutéž metodu se stejným významem v kontextu jejich třídy, přetěžování operátorů neboli provedení rozdílné operace v závislosti na typu operandů.

Správa paměti je v jazyce C# automatická. Spoléhá se na Common Language Runtime (CLR) který zároveň s programem spouští garbage collector („sběrač odpadků“), který uvolňuje paměť za objekty, které již nejsou v programu referencované. Díky tomuto nemusí programátor řešit alokaci a dealokaci paměti pro objekty ve svém programu, také to řeší velkou část problémů s ukazateli na objekt, který již v paměti neexistuje. Z jazyka C# nezmizely pointery (ukazatele), jako jsou například v jazycích C a C++, ale spíše se nepoužívají (pouze ve velmi vzácných případech kdy je dbáno na velkou rychlost kódu) [19].

Jazyk C# se primárně používá pro psaní kódu pro programy, které běží na platformě Windows. O funkčnost kódu se v tomto případě stará .NET Framework. .NET Framework obsahuje Microsoft CLR a také velký počet knihoven. Kód se kompiluje do spustitelných souborů (přípona .exe), či knihoven (přípona .dll). Avšak kód v jazyce C# může být spuštěn i na jiné platformě. Jazyk C# může běžet na serveru a vytvářet HTML kód, toto je případ ASP.NET v současnosti velmi používaný pro tvorbu webových stránek. Nebo například Projekt Mono [20] je open source implementace kompilera a CLR pro platformy Linux, Solaris, Mac OS X a Windows.

Windows Presentation Foundation (WPF) je grafický subsystém pro vytváření uživatelských rozhraní v aplikacích pro Windows vyvíjený společností Microsoft. Je součástí .NET Frameworku od verze 3.0 a pro vykreslování využívá DirectX. Pro vytváření grafických prvků se používá jazyk XAML (jazyk založený na XML struktuře), to kvůli oddělení funkčnosti a vzhledu aplikace. Cílem WPF je sjednotit poutavé uživatelské rozhraní, 2D a 3D grafiku, vektorovou a rastrovou grafiku, animace, vázání dat a audio a

video. WPF může být napojeno na kterýkoliv jazyk běžící na platformě .NET (tedy i na jazyk C#), který obstarává interní logiku aplikace. WPF je přímým nástupcem Windows Forms. Uživatelské rozhraní pro Microsoft Visual Studio 2010 a novější jsou například napsané v WPF.

3.2.3. Výběr knihovny pro kreslení grafů

Velkou část aplikace tvoří zobrazení dat. Z povahy dat je často nejlepší zobrazení pomocí grafů. Velkou pozornost jsem proto věnoval výběru knihovny pro kreslení grafů. Důležitým kritériem výběru bylo, aby knihovna měla stále aktivní vývoj a podporu. Uvedu několik příkladů dostupných knihoven a zhodnotím jejich klady a zápory pro použití na tomto projektu.

3.2.3.1. Knihovny s placenou licenci

Za zmínku stojí dvě knihovny pro kreslení grafů v C# WPF a to jsou LightningChart a SciChart. Jsou to profesionální knihovny s velkým množstvím funkcí a možností. Obě podporují hardwarovou akceleraci pomocí GPU a chlubí se velmi rychlým vykreslováním grafů. Mají placenou licenci, v tuto chvíli za 764 USD za 1 licenci na 1 rok v případě SciChart a za 2470 USD za 1 licenci na 1 rok v případě LightningChart.

Po dohodě s vedoucím práce jsme usoudili, že se aplikaci pokusíme napsat za pomoci knihovny s bezplatnou licenci, v případě, že by nevyhovovala, tak bychom začali uvažovat o použití knihovny s placenou licenci.

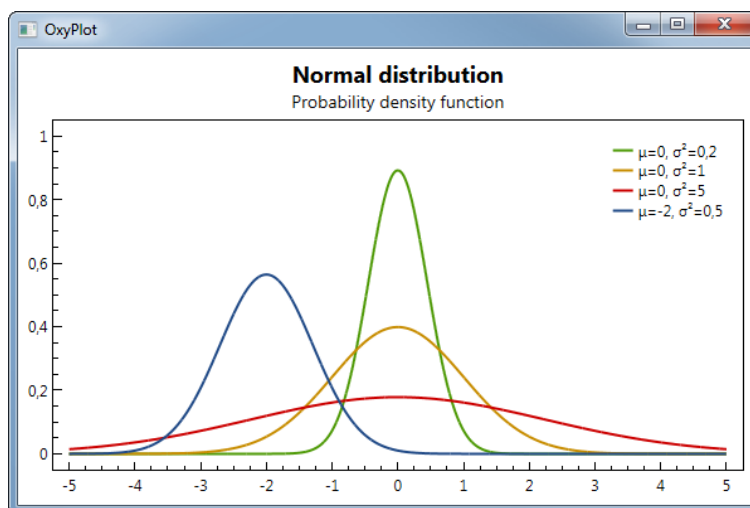
3.2.3.2. Knihovny s volnou licenci

Knihoven s volnou licenci není mnoho a velká část z nich neodpovídá nastaveným požadavkům. Samotné WPF nabízí možnost kreslení základních grafů, neumí však velkou část požadovaných funkcí (např. zoom). Další knihovny mají licenci zdarma pouze pro nekomerční účely, nebo mají poslední verzi starší než dva roky (Dynamic Data Display, Sparrow toolkit, atd.). Jediná knihovna s bezplatnou licenci odpovídající podmínkám výběru se ukázala být Oxyplot.

3.2.3.3. Knihovna Oxyplot

Oxyplot je multiplatformní knihovna pro .NET. Umožňuje kreslení grafů v WPF, Windows 8, Windows Phone, Windows Phone Silverlight, Windows Forms, Silverlight, GTK#, Xwt, Xamarin.iOS, Xamarin.Android, Xamarin.Forms a Xamarin.Mac. To je

obrovskou výhodou, pokud by se měla aplikace v budoucnu přesunout na nějaké jiné zařízení (telefon/tablet, Mac, web). Knihovna je pod MIT licenci, to znamená pro naše použití zcela zdarma. Vývojový tým je velmi aktivní (aktualizace knihovny minimálně jednou týdně) a reaguje na dotazy na svém fóru. Protože je použit na grafy anti-aliasing, tak čáry grafů vypadají opravdu pěkně (Obrázek 14). Přehledná dokumentace, webový prohlížeč, dobré ohlasy uživatelů a vysoké hodnocení na nuget.org.



Obrázek 14 - Příklad grafu pomocí Oxyplot knihovny

Za nevýhody považují hlavně rychlost kreslení bodů, kdy při velkém počtu bodů je celkem trhané posouvání grafem a zoomování. Zatím není implementována akcelerace vykreslování pomocí GPU, ale pracuje se na implementaci.

Počet bodů, které bude aplikace vykreslovat je celkem velký, například pro 10-ti minutový záznam pořízen snímkovací frekvencí 200 snímků za sekundu se jedná o 120 000 bodů. Po zvážení všech výhod a nevýhod byla vybrána knihovna Oxyplot. Pokud by se v budoucnu ukázala jako nevyhovující, nebude příliš složité ji vyměnit za nějakou jinou.

Kapitola 4

Cíle práce

Cílem této diplomové práce je seznámit se s možnostmi využití očních pohybů pro diagnostiku dyslexie a vytvořit aplikace pro umožnění zobrazení a zpracování naměřených záznamů očních pohybů.

Prvním cílem je seznámit se s okulometrickým testem Pavlidis a vytvořit v prostředí Matlab nástroj pro hromadné zpracování tohoto testu.

Druhým cílem je vytvoření aplikace v jazyce C# pro zobrazení naměřených očních pohybů a jejich hodnocení ve spolupráci s firmou Medicton Group s.r.o. zastupovanou Ing. Tomášem Jursou (oponent práce). Požadavky na aplikaci jsou následující:

- umožnit zobrazení videookulografických dat nahraných programem I4Tracker bez nutnosti vlastnit program Matlab a programovací dovednosti
- vytvořit přehledné grafické uživatelské rozhraní ve Windows Presentation Foundation (WPF)
- vytvořit modulární architekturu programu v jazyce C# a popsat postup tvorby nových modulů
- implementovat do aplikace některé funkce z TEMSA toolboxu pro zpracování naměřených dat v jazyce C#
- umožnit další vývoj aplikace

Aplikace by se v budoucnu měla stát součástí kompletního balíku software a hardware pro Eye Tracking (sledování očních pohybů) vyvíjeným firmou Medicton Group

4. CÍLE PRÁCE

s.r.o. Program by měl umožňovat načítání samostatných modulů, předávaných jako DLL knihovny. Tyto moduly budou rozšiřovat možnosti aplikace a implementovat nové funkce do aplikace. Především je myšleno použití modulů pro zpracování úloh dyslexie (jako jsou metody v TEMSA toolboxu). Důležité je zachovat velkou univerzálnost pro neomezování vývojářů, kteří budou pokračovat ve vývoji této aplikace, v kreativě a funkčnosti modulů. Je možné, že některým zákazníkům budou poskytnuté pouze některé moduly. Je tedy nutné, aby aplikace samostatně kontrolovala, zda jsou DLL knihovny přítomny a zobrazovala pouze poskytnuté moduly.

Třetím cílem je vytvoření modulů aplikace pro zpracování okulometrických testů Fixační stabilita a Antisákady.

Kapitola 5

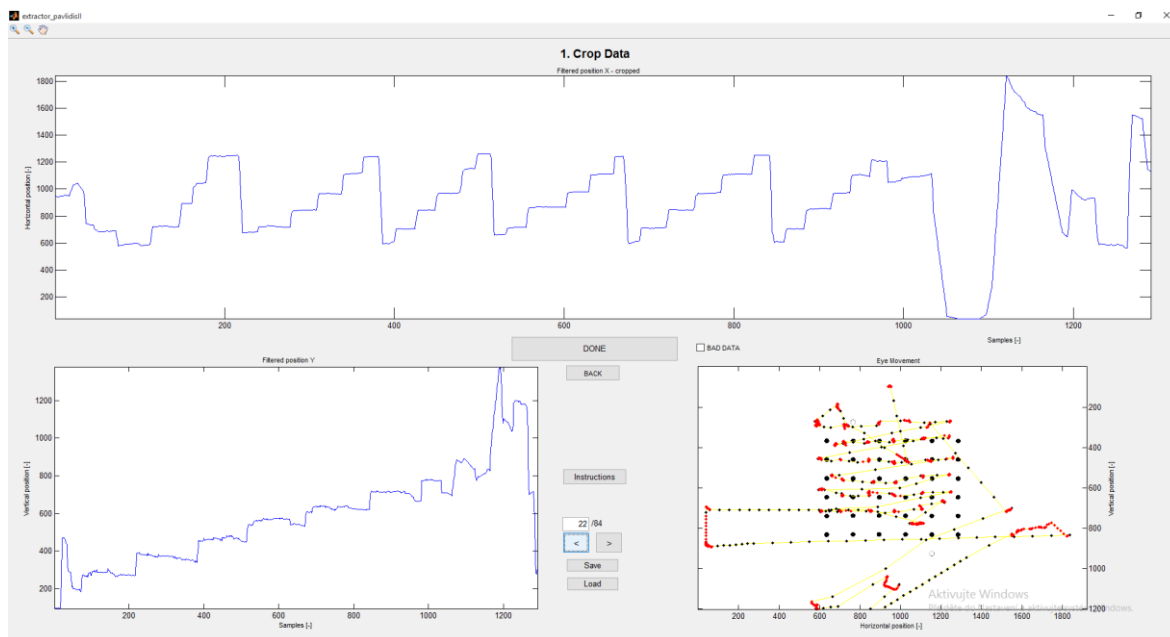
Praktická část

5.1. Nástroj Matlab

Před vytvořením samotné aplikace pro vizualizaci a hodnocení videookulografických dat jsem vypracoval v rámci této diplomové práce nástroj do TEMSA toolboxu pro zpracování očních pohybů v úloze Pavlidis. Požadavkem pro tento nástroj bylo umožnit hromadné zpracování naměřených očních pohybů a pro každý zpracovaný náměr poskytnout požadovaná data (viz 3.1.1) ve vhodném formátu k dalšímu zpracování.

5.1.1. Grafické uživatelské rozhraní nástroje

V prostředí Matlab jsem vytvořil nástroj s grafickým uživatelským rozhraním (GUI) pro zjednodušení zpracování naměřených záznamů této úlohy. Nástroj umožňuje načítání jednoho či více měření v TEMSA formátu a za pomoci přehledného GUI zpracování těchto záznamů (viz Obrázek 15). Jsou zobrazeny 3 grafy, ve kterých je nejlépe vidět průběh úlohy: horní – pozice oka na monitoru v X souřadnici, levý – pozice oka na monitoru v Y souřadnici, pravý – pozice oka na monitoru v XY souřadnicích. V grafech je možné zoomovat a posouvat je, lze ukládat a nahrávat již označená data. U dat ze starší verze trackeru se občas stávalo, že data zobrazená v pravém spodním grafu nepasují na obrázek v pozadí. To je způsobeno mírným pohybem hlavy po provedení kalibrace. Nasnímaná data to trochu znehodnocuje, avšak je možné je později upravit pomocí TEMSA toolboxu. Byl přidán příznak „DISTORTED“ pro označení právě takto posunutých nebo natočených dat.



Obrázek 15 - Modul Pavlidis (GUI hlavní okno - první krok)

5.1.2. Popis zpracování úlohy

Zpracování úlohy probíhá ve třech krocích:

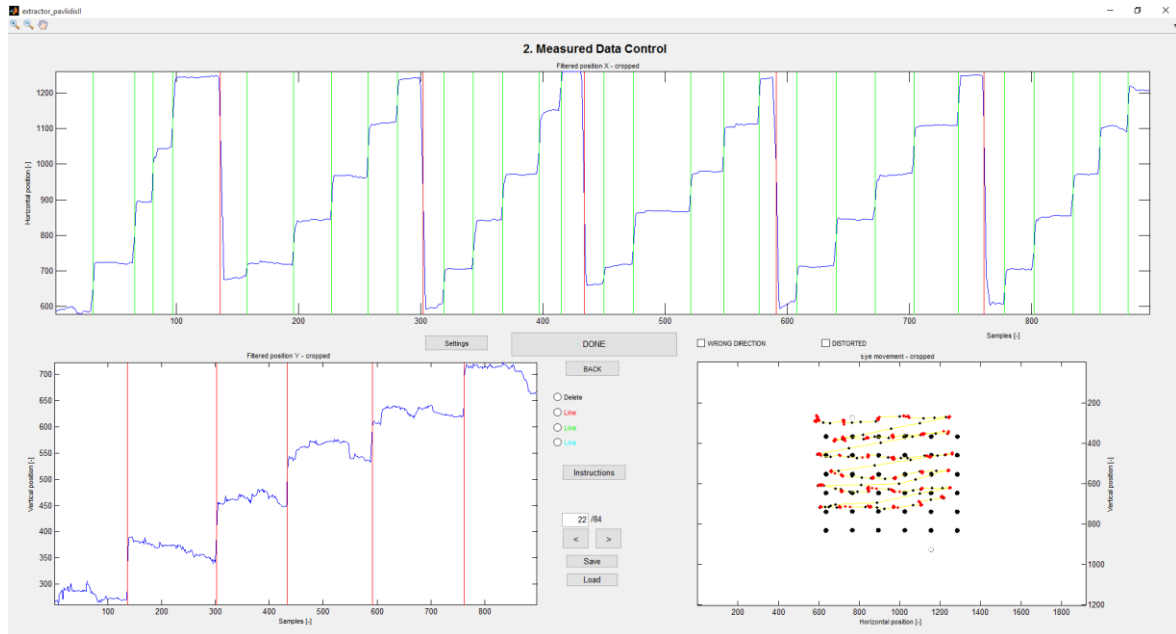
1. Oříznutí dat
2. Kontrola naměřených dat
3. Analýza dat

První krok (oříznutí dat) je nutné provést ručně, GUI v tomto kroku vypadá, jako je již zobrazeno na Obrázek 15. Tato úloha nemá přesně nastavený časový začátek a konec úlohy ani rychlost průběhu - to si řídí testovaná osoba a dává obsluhu najevo, když dokončil úlohu. Proto je v každém náměru chvíle než testovaná osoba začne provádět úlohu a chvíle po zakončení úlohy než obsluha přeruší nahrávání. Kliknutím myši do horního nebo levého grafu se vloží svislá čára označující začátek/konec úlohy, opětovným klikáním se upravuje pozice vložené nejbližší čáry. Ideálně by mělo být označeno šest průběhů, ale je možné, že dítě po několika řádcích ztratilo pozornost a k zpracování nejsou další průběhy použitelné, je tedy možné označit průběhů méně. Pokud se náměr jeví jako nemožný zpracovat, je zde možnost data označit jako špatná (pomocí checkboxu „BAD DATA“) a tím je vyřadit z dalšího zpracování.

V druhém kroku jsou vidět již oříznutá data a automaticky označené sakády pomocí vlastního algoritmu (Obrázek 16). Zelené čáry značí dopředné sakády, světle modré čáry

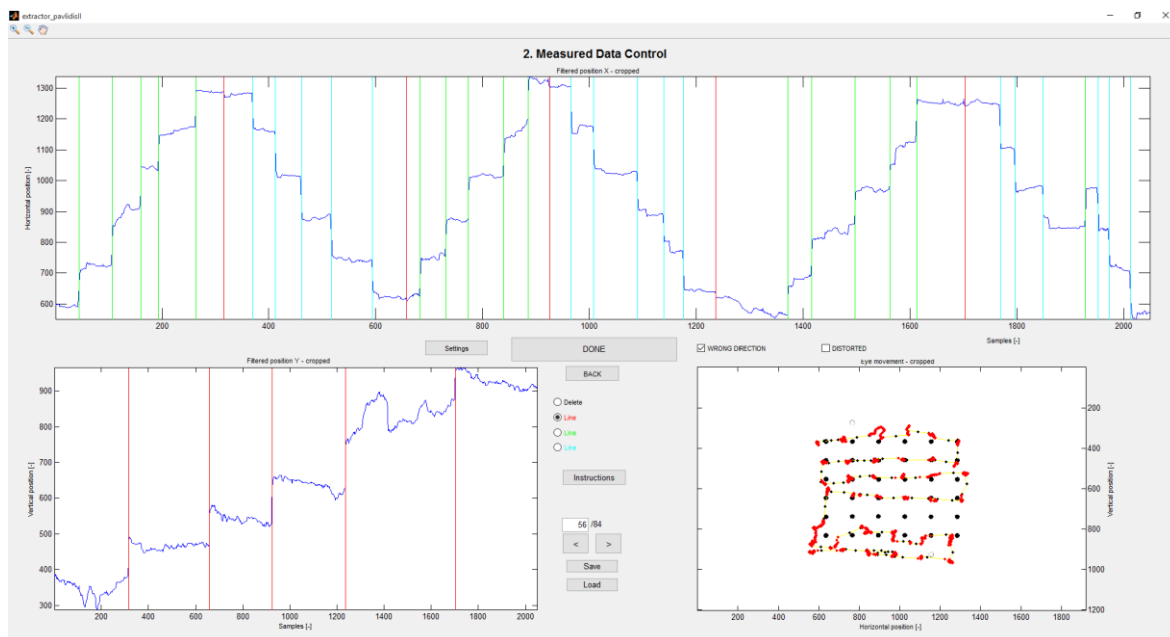
5. PRAKTICKÁ ČÁST

značí zpětné sakády, červené čáry značí přechody na další řádky. Je potřeba zkontrolovat zda algoritmus správně vyhodnotil data a případně upravit konstanty detekce sakád v menu „Settings“, či ručně přidat/odebrat čáry značící sakády. To lze opět pomocí klikání myši v horním či levém grafu.



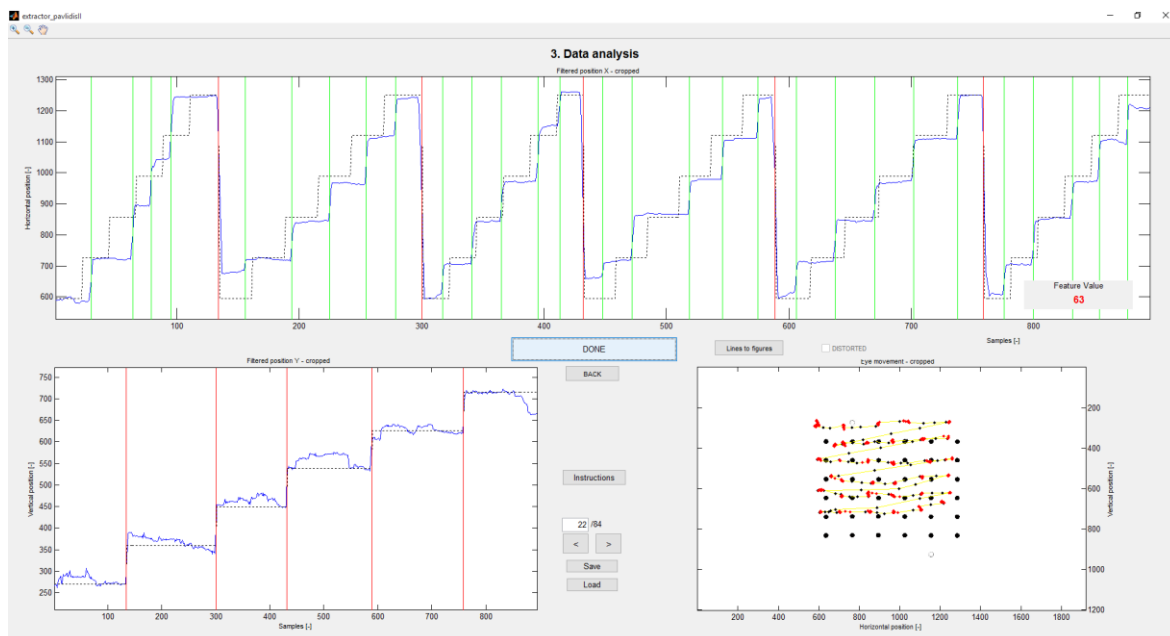
Obrázek 16 - Modul Pavlidis (GUI hlavní okno - druhý krok)

Velmi často se stávalo, že testovaná osoba přesně nepochopila úlohu a po přeskočení na další řádek procházela body zprava doleva a na dalším řádku opět zleva doprava (průběh „jako had“) – viz Obrázek 17. Aby se takováto data nemusela zahodit jako špatná data, je možné pomocí checkboxu „WRONG DIRECTION“ označit špatný průchod úlohou a dále zpracovávat tyto data. Na výsledcích úlohy by takovýto špatný průchod neměl mít vliv, avšak tato data jsou nadále označena hodnotou „WD“ a je na zpracovateli úlohy zda je použije v dalším zpracování.



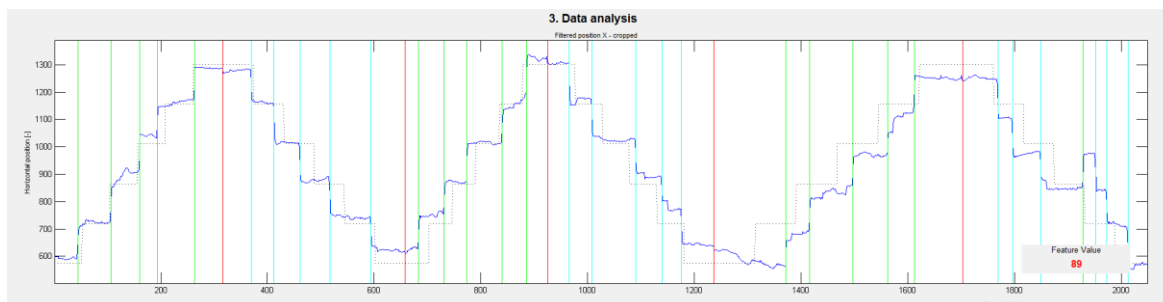
Obrázek 17 - Špatný průchod úlohou Pavlidis

Ve třetím kroku je vygenerován ideální průběh a je spočítána odchylka od tohoto průběhu (označena jako „Feature Value“), hodnota je bezrozměrná a platí, že čím je menší, tím více se data přibližují ideálnímu průběhu. Ideální průběh se zobrazí jako přerušovaná černá čára, odchylka od ideálního průběhu je zobrazena v horním grafu (viz Obrázek 18).



Obrázek 18 - Modul Pavlidis (GUI hlavní okno – třetí krok)

Ideální průběh je vygenerován i v případě špatného průchodu úlohou (viz Obrázek 19), či menším množstvím průběhů.



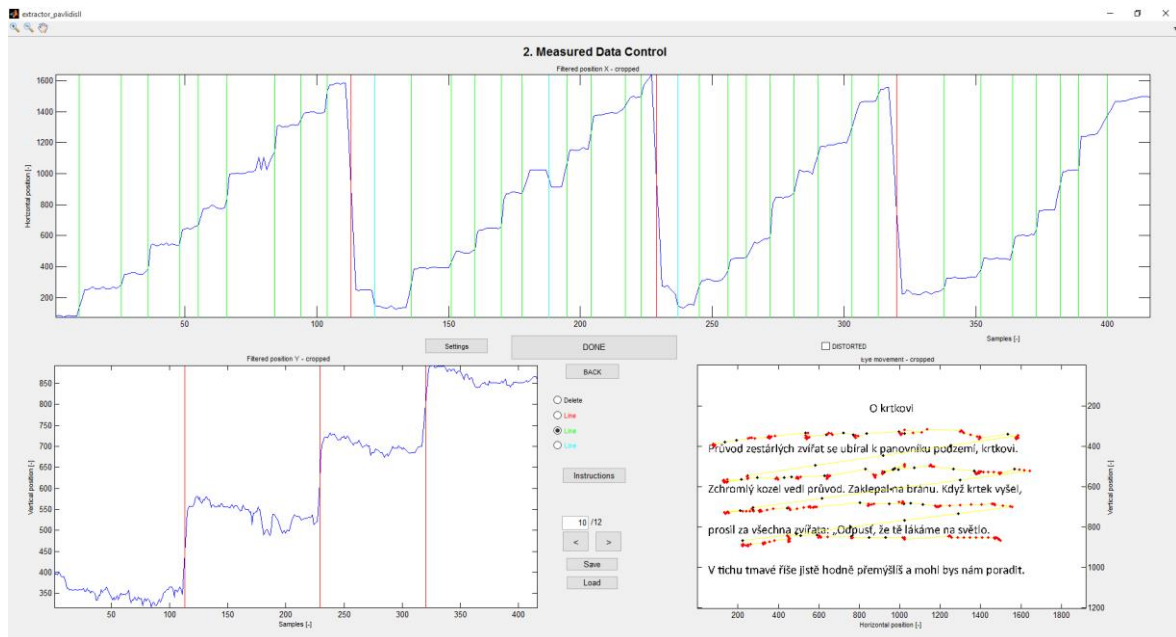
Obrázek 19 - Špatný průchod úlohou Pavlidis (ideální průběh)

Díky přehlednému GUI a návodu k nástroji může tuto úlohu zpracovat téměř kdokoliv. Nástroj automaticky vygeneruje data s příznaky ve formátu vhodném k dalšímu zpracování v prostředí Matlab, například klasifikaci. Formát je stejný jako TEMSA formát, ke každému záznamu je přidána struktura „extractors“ ve které jsou zaznamenány údaje ze zpracování úlohy (viz Tabulka 2).

Název pole	Očekávaná hodnota	Poznámka
badData	hodnota 0/1	boolean – špatná data
WD	hodnota 0/1	boolean – špatný průchod
cropped	pole 2 hodnot	první hodnota značí začátek úlohy, druhá konec úlohy
row_jumps	pole X hodnot	indexy červených čar
column_jumps_forward	pole X hodnot	indexy zelených čar
column_jumps_backward	pole X hodnot	indexy modrých čar
stage	hodnota 1-3	značí krok zpracování, ve kterém se nachází náměr
idealrun_y	pole X hodnot	ideální průběh v ose Y
idealrun_x	pole X hodnot	ideální průběh v ose X
FV	hodnota 0 - ∞	odchylka od ideálního průběhu
Distorted	hodnota 0/1	boolean – zakřivená data

Tabulka 2 - Struktura extractors (výsledek zpracování úlohy Pavlidis)

Později se ukázalo, že je možné tento nástroj také použít ke zpracování úloh se čtením textu, kde je potřeba extrakce velmi podobných příznaků (Obrázek 20). Hodnoty pro ideální průběh jsou v tomto případě nevyužity.

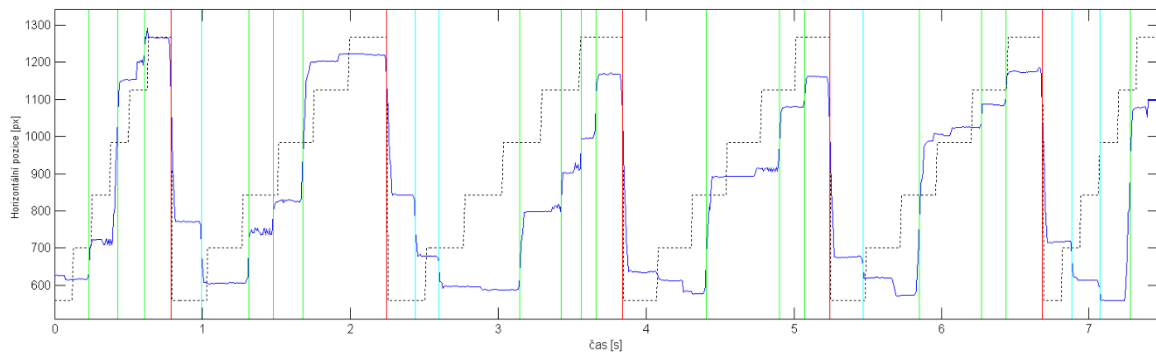


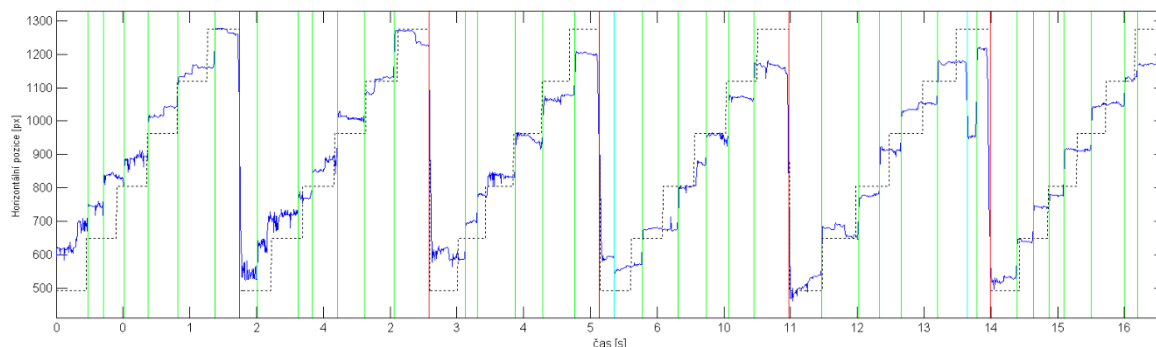
Obrázek 20 - Zpracování textu modulem Pavlidis

5.1.3. Výsledky zpracování

Nástroj je v současné době plně funkční. Jeho funkčnost a použitelnost k zpracování této úlohy demonstruje Návrh certifikované metody [5], konference [21] a diplomová práce [18] ve kterých byl použit pro zpracování a klasifikaci záznamů očních pohybů.

Na dalším obrázku (Obrázek 21) jsou typické naměřené průběhy v ose X pro dítě s diagnózou dyslexie a zdravé dítě. Zelené čáry značí dopředné sakády, světle modré čáry značí zpětné sakády, červené čáry značí přechody na další řádky. Čerchovaná čára značí ideální průchod úlohou. To, zda jsou fixace regulérní, je nutné zjistit dalšími výpočty, nebo to může značit odchylka od ideálního průběhu (vysoká odchylka znamená malé množství regulérních fixací).





Obrázek 21 - Porovnání průběhů v úloze Pavlidis u dítěte s diagnózou dyslexie (11 let, nahoře) a zdravého dítěte (10 let, dole), obrázek převzatý z [5]

Děti s diagnózou dyslexie mají v této úloze často tendenci k narušení správného průchodu úlohou, objevují se časté zpětné sakády a neschopnost provedení regulérních fixací [5].

5.2. Aplikace

Hlavním cílem této diplomové práce bylo vytvořit aplikaci pro vizualizaci a hodnocení naměřených dat programem I4Tracking. Aplikace měla být vytvořena v jazyce C# a grafickém prostředí WPF (Windows Presentation Foundation). Bylo mi umožněno použití jakýchkoliv knihoven s volnou licencí, popřípadě po konzultaci a schválení knihoven s placenou licencí. Aplikace dostala pracovní název I4Viewer.

5.2.1. Grafické uživatelské rozhraní aplikace

Velký důraz byl kladen na přehledné a uživatelsky příjemné GUI (Graphical User Interface - grafické uživatelské rozhraní). Program je koncipován, aby běžel na obrazovce desktopu nebo notebooku s dostatečně velkou úhlopříčkou (alespoň 15 palců). Použitím relativní velikosti pro grafy (automaticky se zmenšují, či zvětšují tak, aby zabírali maximální prostor), které zabírají většinu plochy obrazovky, je zajištěna možnost změny na jiné rozlišení. Doporučené rozlišení aplikace je 1920x1080 px, minimální rozlišení aplikace je přibližně 500x500 px. Tato hodnota by v budoucnu mohla být ještě menší, pokud by to bylo potřeba, provedením grafických změn v oblasti informačních popisků (zmenšení, skrývání, atd.).

Aplikace je navržena, aby pracovala pouze s jedním záznamem očních pohybů. Je však možné otevřít záznamů více – každý v novém okně. Také je možné aplikaci spustit s parametrem cesty k záznamu rec.xml, aplikace jej pak při spuštění rovnou načte. Díky

tomuto je možné aplikaci spouštět například z programu I4Tracking po nahrání záznamu. Toto byl také jeden z požadavků společnosti Medicton Group s.r.o.

Pro zobrazení úlohy se velmi hodí grafy. Jako knihovna pro kreslení grafů byla vybrána knihovna Oxyplot (viz 3.2.3.3). Grafy vypadají velmi pěkně (v porovnání například s grafy z Matlabu), protože je na vykreslené body použit antialiasing. Grafy je možné zoomovat kolečkem myši (v obou osách s kurzorem myši uvnitř grafu, či pouze v jedné ose s kurzorem myši na příslušné ose) a posouvat držením levého tlačítka a pohybem myši. Klik levým tlačítkem myši na libovolný bod grafu vyvolává detailní hodnoty o tomto bodu.

Měření se zobrazuje po jednotlivých částech (úlohách), jak bylo naměřeno. Je možné pomocí tlačítek či otevíracího menu přepínat mezi úlohami. Vzhled grafů je nastavitelný v menu, je možné měnit velikost zobrazených bodů, jejich barvu a barvu čar. Toto nastavení se ukládá a je načteno při dalším spuštění aplikace.

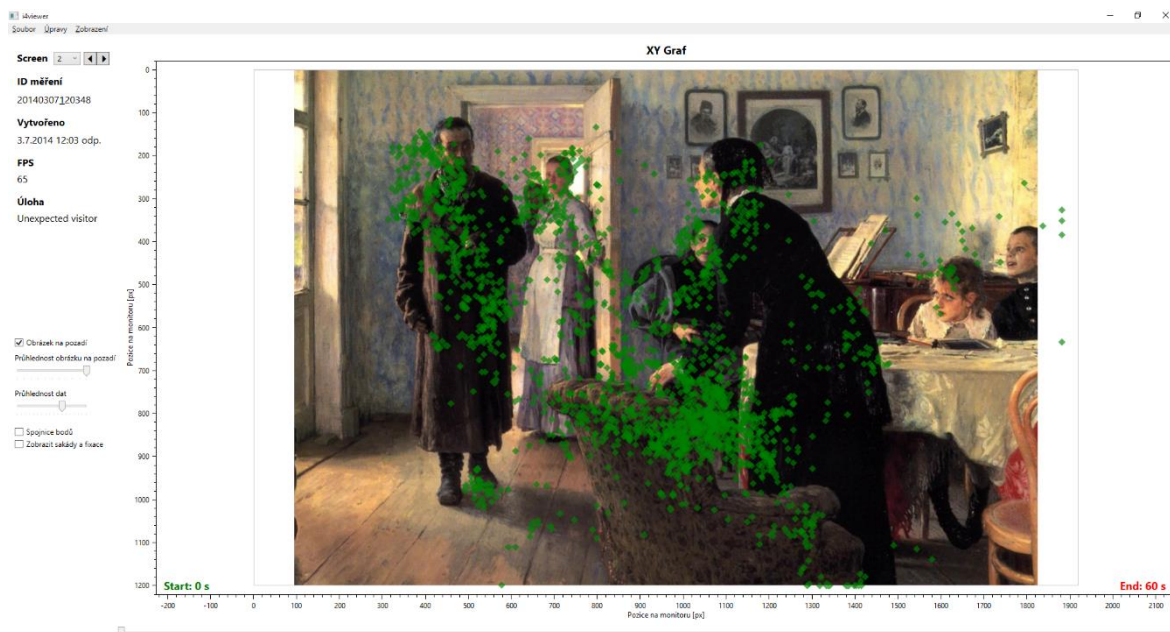
5.2.1.1. Úvodní stránka

Úvodní stránka se zobrazuje po startu aplikace, pokud nebyla ve startovních argumentech předána cesta k souboru, který se má zobrazit (v tom případě je tato stránka přeskočena). Je to stránka, kterou uživatel vidí, když si spustí aplikaci. Tato stránka by mu měla umožnit zobrazit očekávaná data. První dojem je velmi důležitý.

V současné verzi aplikace obsahuje úvodní stránka tlačítko otevřít, kterým je možné vyvolat standardní dialog pro otevření souboru a nalézt na disku záznam měření, a rychlé odkazy na několik ukázkových úloh sloužících k ladění programu.

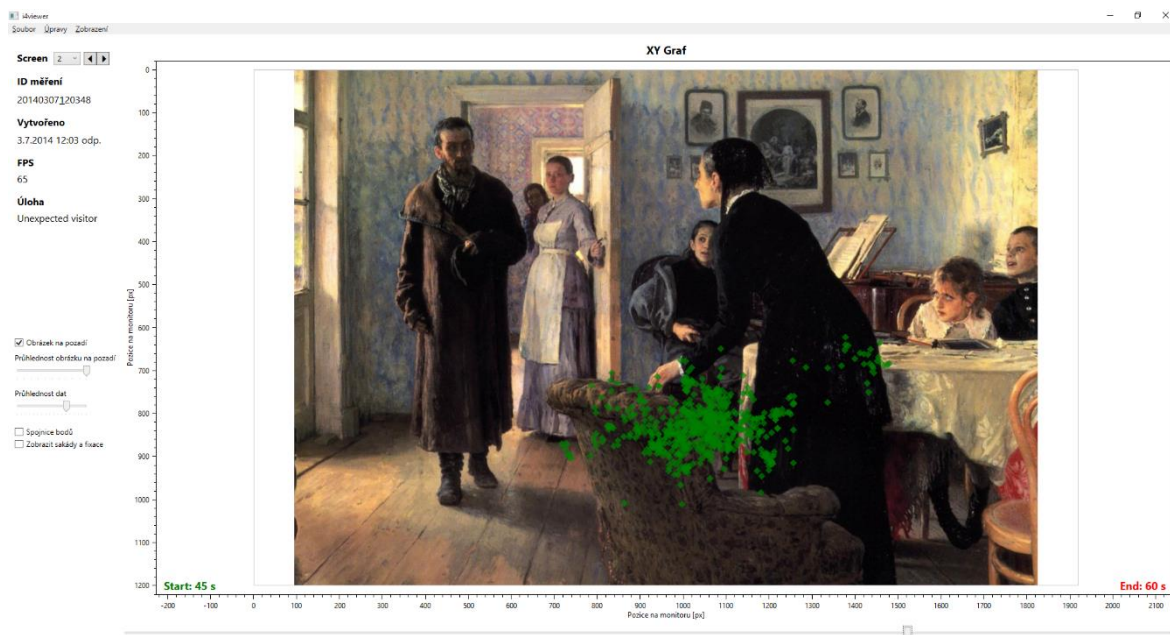
5.2.1.2. Prohlížení záznamu v XY grafu

Nejpřirozenější pohled na naměřená data je v XY souřadnicích. Data jsou zobrazena, jako byla na monitoru při nahrání a lze tak dobře pozorovat, do jakých míst se člověk nejvíce díval. Velmi se hodí pro statické stimuly se zónami, jako je například na následujícím obrázku (Obrázek 22), kde bylo úkolem si po dobu jedné minuty prohlížet obraz „Nečekaný host“ od Ilyi Repina. Tato úloha byla použita při výzkumu parafílie v léčebně Bohnice.



Obrázek 22 - Aplikace I4Viewer: zobrazení XY grafem

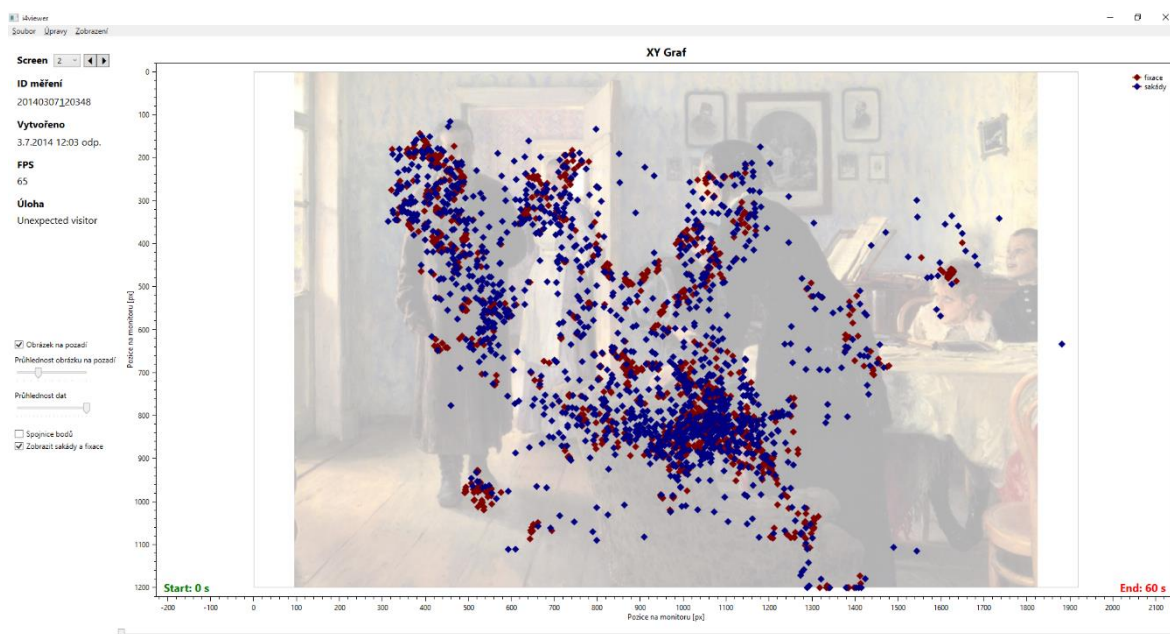
V grafu je vidět, že testovaný si dobře prohlédl postavy na obraze, některé ho zaujaly více, některé méně. Pro zobrazení pouze části snímaného času je ve spodní části zabudován dvojitý posuvník, kterým lze omezit zobrazená data podle vymezeného času. Díky tomuto můžeme pozorovat, že tentýž testovaný si posledních 15 vteřin z minutového záznamu prohlížel „prázdné místo“ v obraze.



Obrázek 23 - Aplikace I4Viewer: zobrazení XY grafem (časové omezení výběru)

Jsou implementovány funkce pro skrytí obrázku na pozadí a změnu průhlednosti obrázku na pozadí i zobrazených dat (pomocí posuvníků v levé části obrazovky). Je možné zobrazit spojnice bodů, zatím však pouze jako neorientované úsečky. Avšak předal jsem tvůrcům knihovny Oxyplot podnět k přidání orientovaných úseček [22]. Také jsem implementoval funkci pro výpočet fixací a sakád. Tuto funkci jsem přepsal do jazyka C# podle již existující funkce v TEMSA toolboxu pro Matlab.

Sakády a fixace se v tomto případě získávají prahováním úhlové rychlosti. Pokud je úhlová rychlost jednotlivých bodů vyšší než nastavený práh, bod se prohlásí za sakádu, pokud je nižší, bod se prohlásí za fixaci. Prah je možné měnit (Úpravy - Nastavit práh pro sakády a fixace) a přímo pozorovat, jak to ovlivňuje určení bodů v grafu. Přepínání zobrazení fixací a sakád je pomocí zaškrtnutí pole (Obrázek 24).

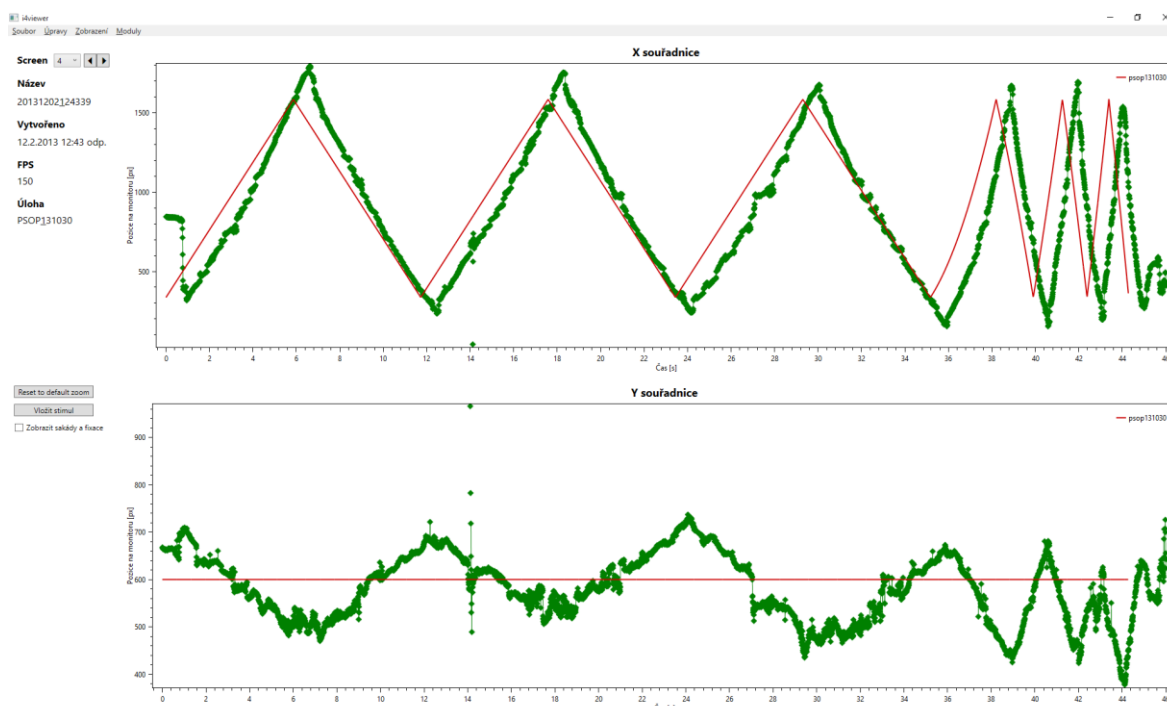


Obrázek 24 - Aplikace I4Viewer: zobrazení XY grafem (zobrazení fixací a sakád)

5.2.1.3. Prohlížení záznamu v rozděleném grafu

Pro některé úlohy je velmi vhodné analyzovat zvláště oční pohyby v horizontální ose (X souřadnice) a vertikální ose (Y souřadnice). Například se jedná o úlohy, kde je úkolem sledovat objevující se / pohybující se bod, úlohy čtení, úloha Pavlidis...

V tomto okně jsou zobrazeny 2 grafy, v horním se vykresluje x-ová souřadnice bodů v závislosti na čase, ve spodním se vykresluje y-ová souřadnice bodů v závislosti na čase (Obrázek 25). Grafy mají „spojenou“ časovou osu, pokud se tedy zoomují data v jednom grafu, v druhém se automaticky provádí stejný zoom na časové ose. Je možné zobrazení fixací a sakád i v těchto grafech.

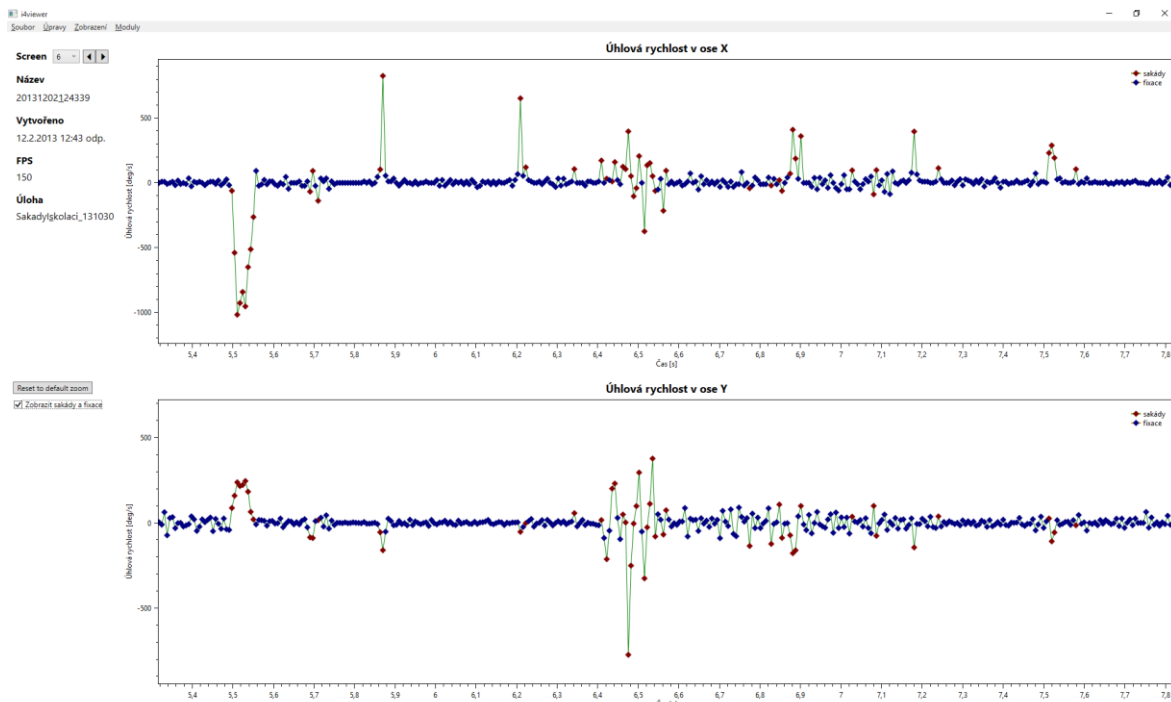


Obrázek 25 - Aplikace I4Viewer: zobrazení v rozděleném grafu

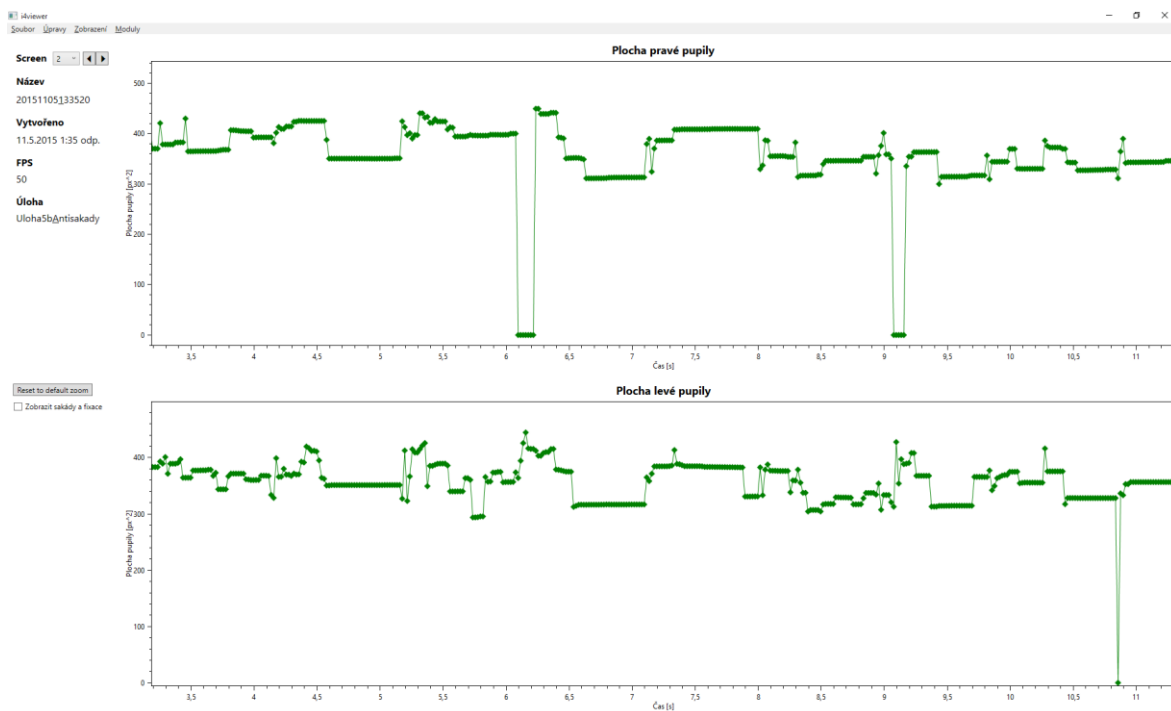
Jsou implementovány funkce pro vložení předem připraveného stimulu v Matlabovém formátu `.mat`. Aplikace automaticky prochází složku, ve které jsou uloženy předpřipravené stimuly a pokud se název stimulu alespoň z 75% podobá názvu úlohy, automaticky jej zobrazí. Podobnost dvou stringů se počítá na základě Levenshteinovy vzdálenosti. Levenshteinova vzdálenost je vzdálenost dvou řetězců (a , b), definovaná jako minimální počet operací vkládání, mazání a substituce takových, aby po jejich provedení byly zadané řetězce totožné. Levenshteinovu vzdálenost zavedl v roce 1965 Vladimír Levenshtein [23]. Stimul je možné vložit i ručně, pomocí klasického otevřít dialogu (open dialog).

5.2.1.4. Zobrazení úhlové rychlosti a velikosti pupily

Poslední z připravených zobrazení nahraných dat jsou zobrazení úhlové rychlosti (Obrázek 26) a zobrazení plochy pupily/y (Obrázek 27). Tato zobrazení se hodí při analýze specifických úloh. Například zobrazení úhlové rychlosti při pozorování pohybujících se stimulů, plocha pupily v úloze s měnícím se okolním osvětlením, či měření emociálních reakcí. V obou zobrazeních lze přepnout na zobrazení sakád a fixací. Úhlová rychlost není přenášena v XML souborech s daty, je počítána na základě rovnic z TEMSA toolboxu, které byli implementovány v jazyce C#.



Obrázek 26 - Aplikace I4Viewer: zobrazení úhlové rychlosti



Obrázek 27 - Aplikace I4Viewer: zobrazení plochy pupily

5.2.2. Modularita aplikace

Pro umožnění maximální univerzálnosti a kreativity při tvorbě modulů byla modularita aplikace navržena tak, aby vývojář byl co možná nejméně vázaný omezeními aplikace. Vývojáři je poskytnuto celé okno aplikace s výjimkou horní lišty s menu. Modul přijímá data experimentu od aplikace, může je tedy jakkoliv upravovat, zobrazovat, počítat statistiky... Je požadována pouze velmi základní struktura modulu (viz 5.2.2.1). Tato struktura pouze nařizuje použití WPF a hlavního prvku modulu jako typ Page (třída System.Windows.Controls.Page). Omezení na WPF není úplně přesné, protože existují postupy jak pomocí WPF zobrazovat grafické prvky, vytvořené v jiných grafických subsystémech (např. Windows Forms). To, zda vývojář bude vytvářet další okna, či stránky aplikace je již zcela na něm a nijak se neomezuje.

Aplikace automaticky načítá zkompilevané moduly ve formátu DLL knihoven a vkládá je do menu aplikace. V této verzi aplikace se všechny moduly načítají ze složky „Modules“ v adresáři aplikace a odkazy na ně se vkládají do menu Moduly. V budoucích verzích je možné vylepšit načítání modulů, například přidáním podmínek načítání modulů, bude-li to vyžadováno.

5.2.2.1. Vytvoření nového modulu

Pro vytvoření nového modulu je možné přepsat si ModuleExample podle svých potřeb, nebo postupovat podle návodu níže. Návod se může zdát jako obtížný, ale nezabere více než minutu. Tento postup je platný pro Visual Studio 2015, pro ostatní verze se může mírně lišit.

1. V Sollution I4Viewer vytvořit nový projekt, typ projektu nastavit jako WPF aplikace a název projektu musí být ve formátu „Module*“ (místo * je možné doplnit jiné znaky).
2. Vymazat z projektu „App.xaml“
3. V souborech „MainWindow.xaml“ a „MainWindow.xaml.cs“ přepsat výskyty Window na Page (pouze doslovné výskyty, složeniny jako např. „MainWindow“ se nepřepisují).
4. Ve vlastnostech projektu zkontrolovat zda má projekt stejný „Target framework“ jako projekt I4Viewer a popřípadě sjednotit, přepnout „Output type“ na „Class Library“
5. Vložit referenci na projekt I4Viewer

6. Je důležité zachovat metodu `MainWindow()` a přidat k ní parametr `MainWindow(Experiments experiment)` – tato metoda se spouští jako hlavní metoda modulu
7. V souboru „`MainWindow.xaml.cs`“ přidat řádek „`using I4Viewer.Experiment;`“ na začátek souboru.
8. Je vhodné vyplnit pole „`Title`“ pro stránku `MainWindow`

Díky tomuto postupu se vytvoří samostatný projekt, který je oddělený od hlavního projektu aplikace a může být vyvíjen nezávisle na aplikaci. Tento projekt je zkompileován jako samostatný program do DLL knihovny a je spustitelný hlavní aplikací. Pokud se použijí knihovny již referencované v projektu `I4Viewer`, není nutné opakovaně přidávat reference i do projektu s modulem. Pokud by byla potřeba přidat reference na nové knihovny, doporučuje se přidat referenci do projektu `I4Viewer` (pro zajištění optimální funkčnosti), referováním na projekt `I4Viewer` se tím pádem automaticky referují i na nově přidanou knihovnu. Po provedení sestavení projektu (`Build`) se projekt zkompileuje do DLL knihovny, pokud se tato knihovna vloží do složky s aplikací „`I4Viewer/Modules`“, aplikace jí automaticky načte. Titulek stránky `MainWindow` se použije jako popis tlačítka v menu.

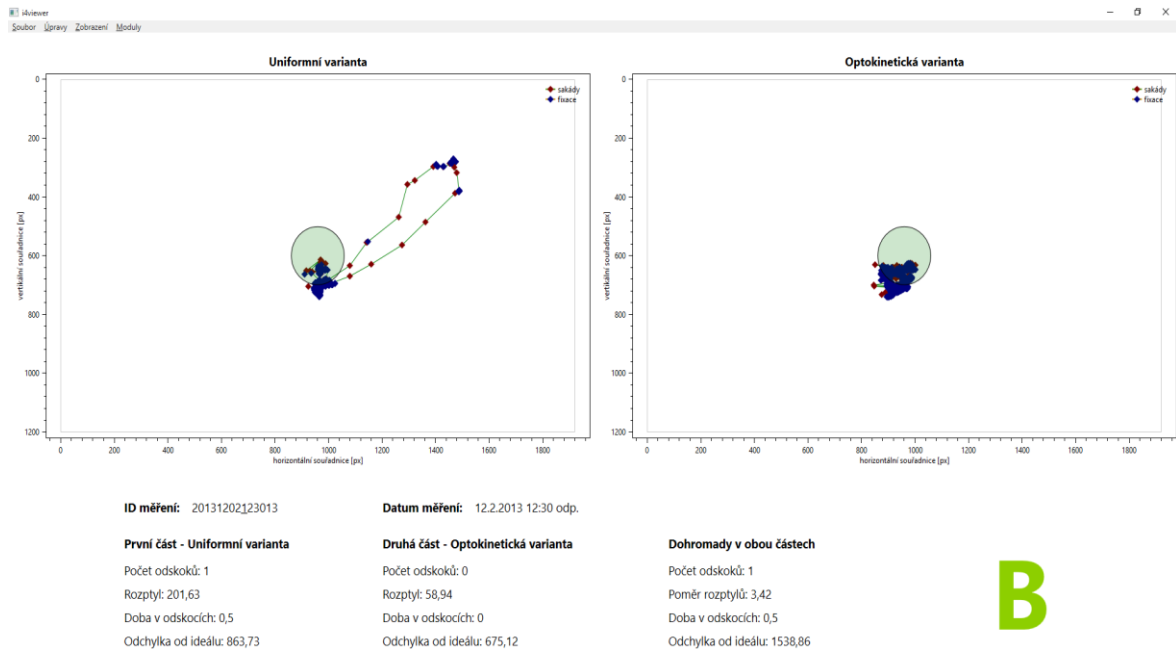
5.2.3. Modul pro zpracování úlohy Fixační stabilita

Jako názorný příklad použití modulů jsem vypracoval modul pro zpracování úlohy Fixační stabilita. Kód, který vypočítává hodnotu příznaků, jsem převzal z `TEMSA` toolboxu pro `Matlab` a přepsal ho do jazyka `C#`. Snažil jsem se ponechat kód velmi podobný jako je v `TEMSA` toolboxu pro snadnější debugování a provádění případných změn.

Modul se automaticky snaží najít úlohu Fixační stabilita v otevřeném záznamu porovnáváním stringů (viz 5.2.1.3). Pokud jej v úloze nenalezne, otevře se okno pro manuální označení úlohy (je možné zpracovat jakoukoliv úlohu, avšak zpracování jiných úloh nemá význam, pokud se zadáním velmi nepodobají úloze fixační stability). Po provedení potřebných výpočtů se zobrazí okno s výsledky (Obrázek 28). Výsledek je zobrazen ve dvou grafech (stejně jako v `TEMSA` toolboxu) a vypočtené hodnoty příznaků se zobrazují pod grafy. V této úloze je výstupem 12 hodnot příznaků (viz 3.1.2), jsou to počty odskoků, statistický rozptyl bodů, celková doba odskoků a odchylka od ideálního průběhu. Příznaky se počítají, jak pro obě části úlohy zvlášť, tak i pro porovnání obou částí.

5. PRAKTICKÁ ČÁST

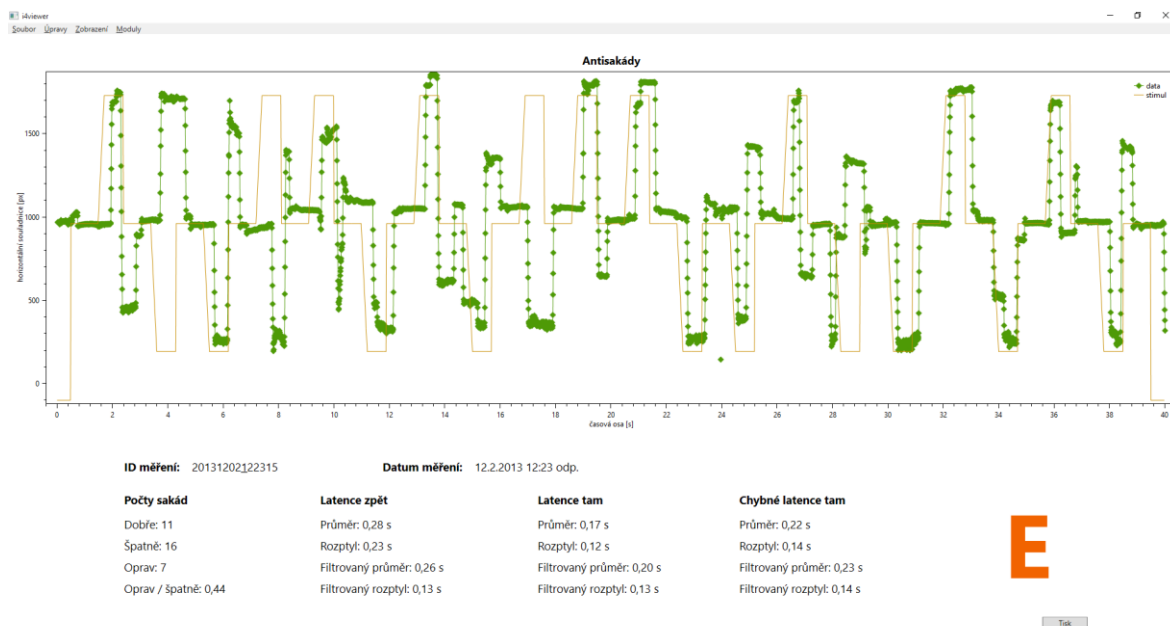
Hodnocení úlohy porovnává vypočtené hodnoty příznaků s referenčními normami a klasifikuje daný náměr známkou. Modul umožňuje tisk celé stránky, jak na připojenou tiskárnu, tak například do PDF.



Obrázek 28 - Modul pro zpracování úlohy Fixační stabilita

5.2.4. Modul pro zpracování úlohy Antisakády

Velmi podobně jako v 5.2.3 jsem vytvořil modul pro zpracování úlohy Antisakády (zadání úlohy viz 3.1.3). Tento modul je vytvořen na základě funkcí z TEMSA toolboxu pro Matlab. Po výpočtu příznaků se zobrazí okno s hodnocením úlohy (viz Obrázek 29). V grafu je vidět horizontální pohyb očí v závislosti na čase a zobrazený stimul. V této úloze je výstupem 16 hodnot příznaků. Jsou to: počty dobře a špatně provedených antisakád, počet oprav, poměr oprav k počtu špatně provedených antisakád, průměrná doba latence a směrodatná odchylka v době všech latencí. Hodnoty příznaků jsou vypsány pod grafem a na základě těchto hodnot je vypočítána výsledná známka.



Obrázek 29 - Modul pro zpracování úlohy Antisakády

5.2.5. Možnosti dalšího vývoje aplikace I4Viewer

Aplikaci je možné použít v této verzi pro prohlížení naměřených dat a zpracování úloh, pro které byly vytvořeny moduly. Jako další vývoj aplikace se nabízí přidání nových modulů zpracujících další úlohy testů dyslexie [5], či jiných úloh založených na sledování očních pohybů. Je možné přidat možnost hromadného zpracování záznamů, exportování záznamů a funkce pro úpravu a zobrazení dat, buď jako samostatný modul, nebo přímo do aplikace.

Pro finální produkt bude vhodné upravit úvodní stránku. Měla by obsahovat odkazy na naposledy otevřené soubory, seznam změn v poslední verzi, úvodní slovo k programu, odkaz na webové stránky s programem a online nápovědou, kontakt na technickou podporu. Také bude potřeba vyřešit licencování aplikace.

Kapitola 6

Závěr

Prvním cílem v praktické části mé práce bylo seznámit se se zpracováním naměřených videookulografických dat programem I4Tracking na úloze Pavlidis. Vytvořil jsem nástroj pro hromadné zpracování této úlohy v prostředí Matlab (viz 5.1). Funkčnost a použitelnost tohoto nástroje byla několikrát ověřena při zpracování naměřených dat ve studiu dyslexie. Tento nástroj byl integrován do TEMSA toolboxu pro Matlab.

Druhým cílem bylo vytvořit aplikaci pro vizualizaci a hodnocení videookulografických dat v jazyku C# za použití WPF (Windows Presentation Foundation). Aplikace byla vytvářena ve spolupráci se společností Medicton Group s.r.o., která vyvíjí systém pro sledování očních pohybů I4Tracking. Vytvořil jsem aplikaci s pracovním názvem I4Viewer. Tato aplikace umožňuje zobrazovat naměřená data programem I4Tracking (viz 5.2.1.2 až 5.2.1.4) v přehledných a uživatelsky nastavitelných grafech. Aplikace implementuje modulární architekturu (viz 5.2.2), dle cílů práce. Popsal jsem postup vytvoření modulů (viz 5.2.2.1).

Třetím cílem bylo vytvoření modulů pro výpočty příznaků pro okulometrické testy: úloha Fixační stabilita a úloha Antisakády dle TEMSA toolboxu. Moduly byly vytvořeny (viz 5.2.3 a 5.2.4), jsou plně funkční a poskytují stejné výsledky jako TEMSA toolbox.

Popsal jsem možnosti dalšího vývoje této aplikace (viz 5.2.5). Tato aplikace by se v budoucnu měla stát součástí kompletního balíku software i hardware pro sledování očních pohybů od firmy Medicton Group s.r.o. Všechny cíle dané zadáním práce a požadavky společnosti Medicton Group s.r.o. byly splněny, výsledná aplikace je plně funkční.

Bibliografie

- [1] J. Jošt, Čtení a dyslexie, 1. ed., Praha: Grada, 2011.
- [2] E. Kowler, "Eye movements: The past 25years," *Vision Research*, vol. 51, no. 13, pp. 1457-1483, 2011.
- [3] K. Holmqvist, Eye tracking: a comprehensive guide to methods and measures, 1st pub. editor, Oxford: Oxford University Press, 2011.
- [4] World Health Organisation, „ICD-10 Version:2016,“ [Online]. Available: <http://apps.who.int/classifications/icd10/browse/2016/en#/F81>. [cit 10. leden 2016].
- [5] J. Jošt, J. Doležal, M. Dobiáš a V. Fabián, „Metodika záznamu očních pohybů a dyslexie, cesta k objektivizaci nálezů v psychologii a speciální pedagogice,“ Praha, 2014.
- [6] D. Hansen a Qiang Ji, „In the Eye of the Beholder: A Survey of Models for Eyes and Gaze,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, sv. 32, č. 3, pp. 478-500, 2010.
- [7] K. Harezlak, P. Kasproski a M. Stasch, „Towards Accurate Eye Tracker Calibration – Methods and Procedures,“ *Procedia Computer Science*, sv. 35, pp. 1073-1081, 2014.
- [8] „SensoMotoric Instruments - Oficiální stránky výrobce,“ [Online]. Available: <http://www.smivision.com/>. [cit. 10. leden 2016].
- [9] „Tobii Pro - Oficiální stránky výrobce,“ [Online]. Available: <http://www.tobii.com/>. [cit. 10. leden 2016].
- [10] „The Eye Tribe - Oficiální stránky výrobce,“ [Online]. Available: <https://theyetribe.com/>. [cit. 10. leden 2016].
- [11] „SR Research - Oficiální stránky výrobce,“ [Online]. Available: <http://www.sr-research.com/>. [cit. 10. leden 2016].
- [12] „Neurotechnology - Oficiální stránky výrobce,“ [Online]. Available: <http://www.neurotechnology.com/sentigaze.html>. [cit. 10. leden 2016].
- [13] G. Mazo, „How to use Smart Stay on the Galaxy S3,“ Android Central, [Online]. Available: <http://www.androidcentral.com/how-use-smart-stay-galaxy-s3>. [cit. 10. leden 2016].
- [14] L. Rollenhagen, „New Patent Hints at 'Pay-Per-Gaze' Advertising for Google Glass,“ Mashable, [Online]. Available: <http://mashable.com/2013/08/15/google-glass-pay-per-gaze/#AWgB1KRKZGqZ>. [cit. 10. leden 2016].
- [15] Google Inc., „Gaze tracking system“. US Patent 13/105,701, 13 květen 2011.
- [16] G. T. Pavlidis, „Eye movements in dyslexia - their diagnostic significance,“ *Journal of Learning Disabilities*, sv. 18, č. 1, pp. 42-49, 1985.

- [17] J. Jošt, *Vyhodnocení okulomotorického testu, výzkumná zpráva*, Praha, 2014, p. 23.
- [18] M. De Rosa, „Diagnose of Dyslexia by Eye Movements,“ *Universita' degli Studi di Napoli Federico II, Diplomová práce*, 2015.
- [19] J. Albahari a B. Albahari, *C# 5.0 in a Nutshell*, O'Reilly Media, 2012.
- [20] „Project Mono,“ [Online]. Available: <http://www.mono-project.com/>. [cit. 10. leden 2016].
- [21] J. Doležal a V. Fabián, „Application of eye tracking in neuroscience,“ v *61. společný sjezd České a Slovenské společnosti pro klinickou neurofyzologii*, 2014.
- [22] „Oxyplot issue: Oriented LineStyle for FunctionSeries,“ [Online]. Available: <https://github.com/oxyplot/oxyplot/issues/670> [cit. 10. leden 2016].
- [23] M. Gilleland, „Levenshtein Distance, in Three Flavors,“ *Pittsburg University*, [Online]. Available: <http://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Fall2006/Assignments/editdistance/Levenshtein%20Distance.htm>. [cit. 8. leden 2016].

Příložené CD

Elektronická verze diplomové práce se souvisejícími soubory jsou uloženy na CD a přiloženy k práci. Struktura uložených údajů na CD:

Složka: Diplomová práce

- Elektronická verze diplomové práce – DP_Michal_Zeman.pdf

Složka: Nástroj Pavlidis

- Složka obsahující nástroj Pavlidis – spouštěcí soubor extractor_pavlidisII.mat

Složka: Aplikace I4Viewer

- Složka obsahující aplikaci I4Viewer – spouštěcí soubor i4viewer.exe
- Ukázkové soubory s očními pohyby