

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

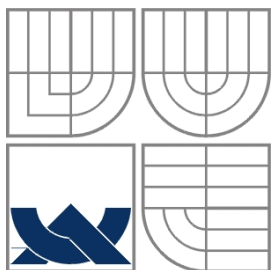
UŽIVATELSKÉ ROZHRANÍ ZALOŽENÉ  
NA ZPRACOVÁNÍ HLOUBKOVÉ MAPY

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

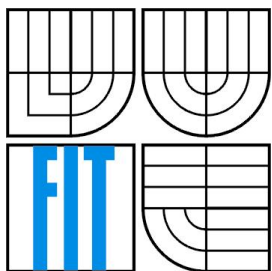
AUTOR PRÁCE  
AUTHOR

BC. PETER KUBICA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# UŽIVATELSKÉ ROZHRAŇÍ ZALOŽENÉ NA ZPRACOVÁNÍ HLOUBKOVÉ MAPY

Depth-Based User Interface

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

BC. PETER KUBICA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. PAVEL ŽÁK

BRNO 2013

## **Abstrakt**

Tradiční uživatelská rozhraní nejsou vždy nejvhodnější variantou ovládní aplikací. Úlohou této práce je seznámit se s problematikou zpracování dat ze senzoru Kinect a analyzovat možnosti ovládní aplikací prostřednictvím hloubkových senzorů. A následně, s využitím získaných znalostí, navrhnout uživatelské rozhraní pro práci s multimediálním obsahem, které pro interakci s uživatelem využívá senzor Kinect.

## **Abstract**

Conventional user interfaces are not always the most appropriate option of application controlling. The objective of this work is to study the issue of Kinect sensor data processing and to analyze the possibilities of application controlling through depth sensors. And consequently, using obtained knowledge, to design a user interface for working with multimedia content, which uses Kinect sensor for interaction with the user.

## **Klíčová slova**

Kinect, hloukový senzor, hloubková mapa, gesto, multimediální aplikace

## **Keywords**

Kinect, depth sensor, depth map, gesture, multimedia application

## **Citace**

Peter Kubica: Uživatelské rozhraní založené na zpracování hloubkové mapy, diplomová práce, Brno, FIT VUT v Brně, 2013

# **Uživatelské rozhraní založené na zpracování hloubkové mapy**

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Žáka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Peter Kubica  
20.5.2013

## **Poděkování**

Ďakujem pánovi Ing. Pavlovi Žákovi za vedenie tejto diplomovej práce a za cenné a inšpiratívne rady poskytnuté pri jej riešení.

© Peter Kubica, 2013

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

1 Úvod.....	2
2 Microsoft Kinect.....	3
2.1 Technológia.....	3
2.2 Knižnice.....	5
2.3 Zhrnutie.....	8
3 Rozhrania založené na gestách rúk.....	9
3.1 Gestá.....	9
3.2 Gestá rúk.....	9
3.3 Gestá rúk snímané farebnou kamerou.....	10
3.4 Gestá rúk a DTW.....	12
3.5 Gestá rúk a HMM.....	13
3.6 Existujúce využitia rozhraní pre gestá rúk.....	19
4 Návrh demonštračnej aplikácie.....	20
4.1 Užívateľské štúdie.....	20
4.2 Návrh konvenčného rozhrania.....	22
4.3 Návrh rozhrania pre gestá rúk.....	23
5 Implementácia konvenčného rozhrania.....	25
5.1 Audio prehrávač.....	25
5.2 Video prehrávač.....	26
5.3 Prehliadač obrázkov.....	26
5.4 Konfiguračné rozhranie.....	27
6 Implementácia rozhrania pre gestá rúk.....	30
6.1 Detekcia rúk bez využitia skeletonizácie.....	30
6.2 Detekcia rúk s využitím skeletonizácie.....	32
6.3 Spracovanie gest.....	34
6.4 Registrácia gest.....	37
6.5 Detekcia gest.....	38
6.6 Ovládanie gestami.....	39
6.7 Tok dát.....	40
7 Testovanie rozhrania pre gestá rúk.....	41
7.1 Parametre detekcie gest.....	41
7.2 Splnenie návrhu rozhrania.....	45
7.3 Užívateľské testovanie.....	47
8 Záver.....	48
Literatúra.....	49
Zoznam príloh.....	52

# 1 Úvod

Tradičné užívateľské rozhrania, ktoré sú založené na interakcii s užívateľom, napríklad pomocou klávesnice, myši, pera, alebo iných konvenčných zariadení, nie sú vždy najvhodnejším variantom ovládania aplikácií. Kvôli ich fyzikálnym, principiálnym, či iným obmedzeniam je ich použiteľnosť limitovaná. Stále viac pozornosti sa dostáva novým prístupom založeným na dotykových obrazovkách, rozpoznávaní ľudského hlasu, či počítačom videní.

Táto práca sa špecializuje na bezdotykové ovládanie gestami rúk. Priame použitie rúk ako zariadenia pre interakciu človeka s počítačom je veľmi atraktívne. Gestá rúk sú neoddeliteľnou súčasťou medzilidskej komunikácie a preto by mohli slúžiť ako prostriedok, do určitej miery prirodzenej, interakcie človeka s počítačom. Ovládanie gestami rúk je v prípade tohto diplomového projektu založené na spracovaní hĺbkovej mapy<sup>1</sup>, teda v podstate na priestorovom počítačovom videní.

Cieľom je navrhnúť a implementovať užívateľské rozhranie pre prácu s multimediálnym obsahom práve pomocou ovládania gestami rúk, založenom na spracovaní hĺbkovej mapy a navrhnúť a implementovať vhodnú demonštračnú aplikáciu, ktorej bude rozhranie súčasťou. Ovládanie gestami rúk pri práci s multimediálnym obsahom sa javí ako zaujímavé, perspektívne a reálne dobre použiteľné. Hĺbkový senzor, ktorý je pre potreby práce využitý sa nazýva Kinect od spoločnosti Microsoft.

Prvým krokom na ceste k realizácii rozhrania je zoznámenie sa s problematikou spracovania dát zo senzoru Kinect (kapitola 2). Ďalej práca analyzuje rozhrania založené na interakcii prostredníctvom gest (kapitola 3). Kapitola 4 je venovaná návrhu multimediálnej aplikácie, ktorá obsahuje rozhranie založené na gestách rúk. Samotnej implementácii konvenčného rozhrania tejto aplikácie sa venuje kapitola 5, zatiaľ čo kapitola 6 rozoberá implementáciu bezdotykového rozhrania tejto aplikácie, ktorého testovaniu sa venuje kapitola 7. Záver obsahuje sumarizáciu výsledkov práce a možné rozšírenia.

Diplomová práca naväzuje na Semestrálny projekt, z ktorého bola v kapitole 2 prevzatá časť teórie o senzore Kinect a v kapitole 3 časť teórie rozhraní založených na gestách rúk.

---

1 Pojem hĺbková mapa bude vysvetlený v nasledujúcej kapitole.

## 2 Microsoft Kinect

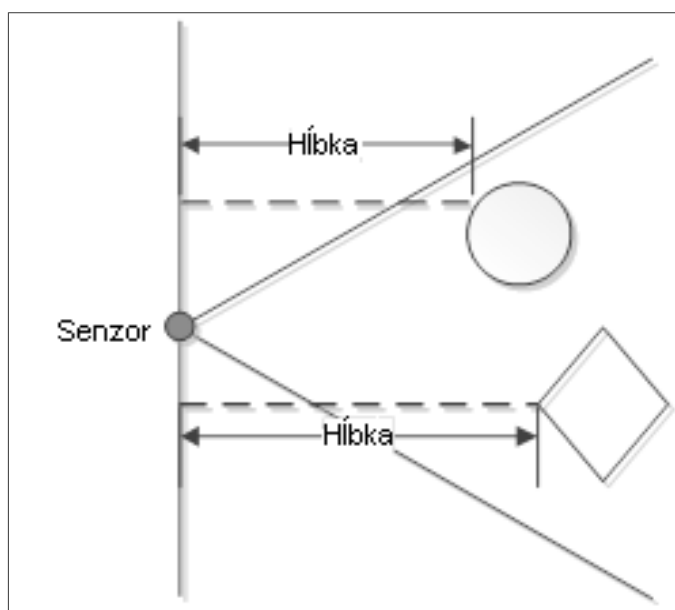
Microsoft Kinect (obrázok 2.1), pôvodne známy pod kódovým označením Projekt Natal, je zariadenie pre 3D snímanie scény pomocou hĺbkového senzora, vyrábané spoločnosťou Microsoft. Zariadenie bolo vyvinuté k hernej konzole Xbox 360 ako doplnok pre ovládanie hier vlastným pohybom tela a hlasovými pokynmi. Konkurenciou v tomto smere ovládania hier sú zariadenia Wii Remote Plus a PlayStation Move pre herné konzoly Wii, resp. PlayStation 3. Zdroje: [1] [9] [10]



Obrázok 2.1: Microsoft Kinect [24]

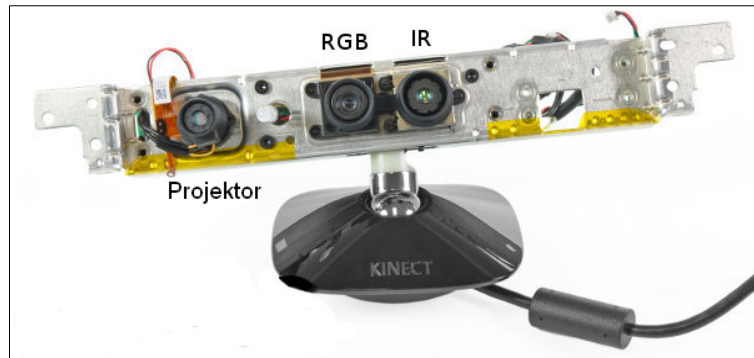
### 2.1 Technológia

Zariadenie obsahuje 3D senzor, RGB kameru a pole mikrofónov na motorizovanom podstavci. Kinect patrí do kategórie tzv. hĺbkových senzorov. To znamená, že detekcia okolia zariadenia je založená na spracovaní hĺbkovej mapy. Hĺbková mapa je 1-kanálový obraz, v ktorom hodnota každého bodu určuje vzdialenosť najbližšieho bodu v priestore na daných súradniciach (obrázok 2.2).



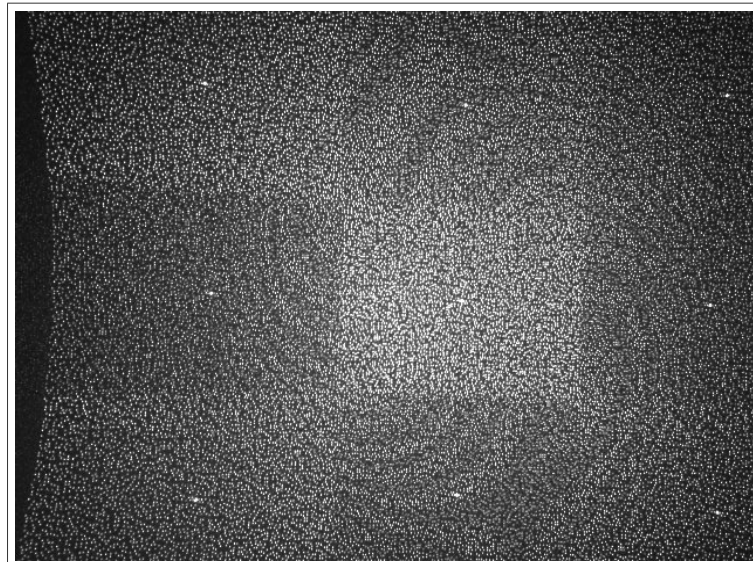
Obrázok 2.2: Význam hĺbky v hĺbkovej mape [24]

Systém pre 3D snímanie použitý v zariadení Kinect sa nazýva „kódovanie svetla“. Ako môžeme vidieť na obrázku 2.3, systém pozostáva z infračerveného laserového žiariča a monochromatického CMOS senzoru, ktoré sú od seba vzdialené približne 7,5 cm.



Obrázok 2.3: Detail zariadenia Kinect [2]

Infračervený žiarič vysiela statický obrazec svetlých a tmavých bodov (obrázok 4). Obrazec je generovaný zo sady difrakčných mriežok so špeciálnym prístupom za účelom znížiť efekt propagácie nultého rádu stredového svetlého bodu.



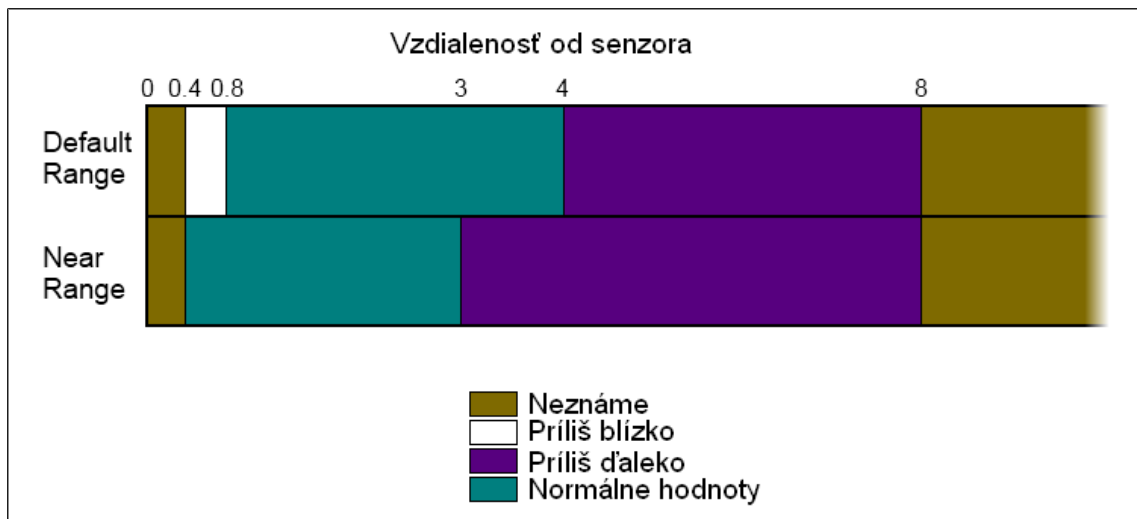
Obrázok 2.4: Infračervený obrazec [2]

Hĺbka je počítaná trianguláciou oproti známemu obrazcu z projektora, ktorému prislúcha určitá hĺbka. Pre každý bod sa použije malé korelačné okno k porovnaniu okolia bodu v známom a zachytenom obraze. Výsledkom tohto porovnania je odchýlka (v bodoch) v hĺbke oproti zapamätanému obrazcu. Táto odchýlka sa nazýva disparita. Hĺbka bodu sa následne vypočíta ako:

$$z = \frac{b \cdot f}{d} \quad (1)$$

kde „z“ je hĺbka (v metroch), „b“ je horizontálna vzdialenosť medzi žiaričom a senzorom (v metroch) a „d“ je disparita (v bodoch). Zariadenie podporuje dva základné režimy snímania hĺbky a to základný režim (*Default Range*) a režim pre malé vzdialenosti (*Near Range*). Oba režimy sú prehľadne znázornené na obrázku 2.5.





Obrázok 2.5: Režimy snímania hĺbky [24]

Presnosť výstupného hĺbkového toku je 11-bitov (2,048 úrovní) a maximálne rozlíšenie je 640x480 pri obnovovacom kmitočte 30 Hz (FPS<sup>2</sup>). Zdroje: [1] [2]

## 2.2 Knižnice

### Kinect for Windows SDK

*Kinect for Windows SDK* [3] je nekomerčný vývojový kit pre prácu so senzorom Kinect vydaný spoločnosťou Microsoft, ktorý obsahuje ovládače kompatibilné so systémom Windows 7. Okrem iného obsahuje bohaté aplikačné programové rozhranie. Poskytuje podporu vývoja softvéru v jazykoch C++, C# a Visual Basic za pomoci nástroja Microsoft Visual Basic 2010. Podporuje priamy prístup k základným výstupným tokom dát z hĺbkového senzoru, RGB kamery i mikrofónového poľa, čo umožňuje budovať nízkoúrovňové programy. Hĺbková mapa je k dispozícii vo frekvencii 30 FPS v nasledovných troch rozlíšeniach:

- 640x480 (základné),
- 320x240,
- 80x60.

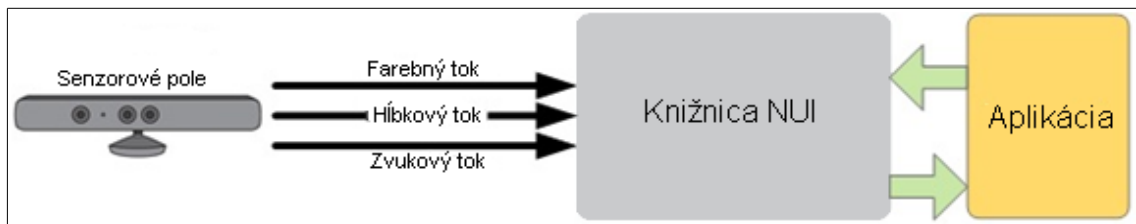
Farebný obraz je poskytovaný v týchto konfiguráciách:

- RGB
  - 1280x960, 12 FPS
  - 640x480, 30 FPS
  - 640x480, 15 FPS
- YUV
  - 640x480, 15 FPS
- Bayer
  - 1280x960, 12 FPS
  - 640x480, 30 FPS

2 Z anglického frames per second – snímkov za sekundu

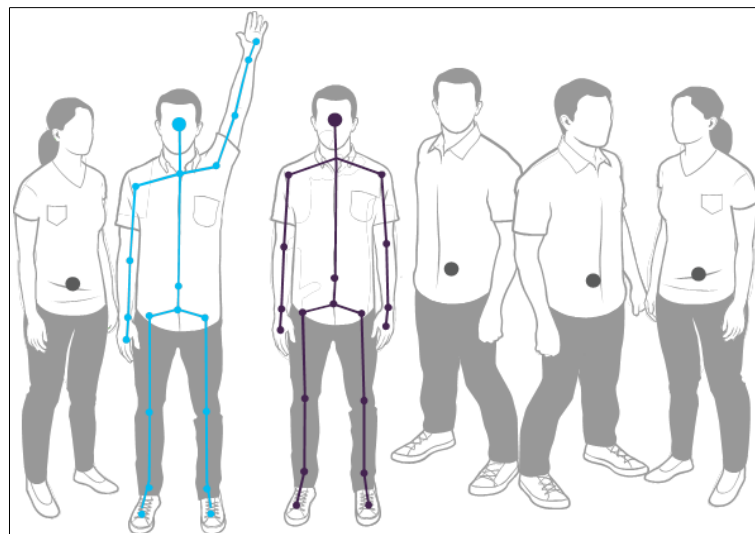
K dispozícii je i obraz v odtieni šedej, ktorý je výstupom infračervenej kamery, a to v konfigurácii: 640x480, 30 FPS.

Ďalej obsahuje knižnicu NUI (Natural User Interface), ktorá využíva základné toky dát a poskytuje pokročilé funkcie. Interakcia medzi HW, SW a aplikáciou je znázornená na obrázku 2.6.



Obrázok 2.6: HW a SW interakcia s aplikáciou [24]

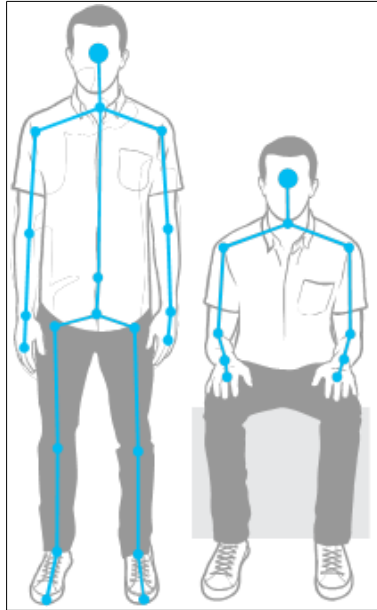
Medzi pokročilé funkcie knižnice NUI patrí podpora skeletonizácie, transformácie súradnicových systémov, či výpočet roviny podlahy. Skeletonizáciu chápeme ako proces extrahovania aktuálnych pozícií a smerov základných rysov ľudskej kostry. Kinect for Windows SDK používa pre skeletonizáciu špeciálny systém, ktorý porovnáva obrazy ľudských tiel s tokom dát vznikajúcim v hĺbkovom senzore. Tento prístup umožňuje veľmi rýchlo detekovať objekty tvaru ľudskeho tela a následne z nich generovať sadu rysov ľudskeho tela. Celý postup je detailne popísaný v článku [25]. Knižnica podporuje súčasné sledovanie maximálne šiestich osôb, pričom maximálne dvoch z nich dokáže skeletonizovať (viď obrázok 2.7).



Obrázok 2.7: Sledovanie a skeletonizácia osôb [24]

Ako je ukázané na obrázku 2.8, skeletonizácia je dostupná v dvoch rozličných módoch, poskytujúcich pozície rôznych rysov ľudskeho tela:

- Základný mód (*Default*): hlava, ramená, stred medzi ramenami, lakty, zápästia, ruky, chrbtica, boky, stred medzi bokmi, kolená, členky, chodidlá.
- Sediaci mód (*Seated*): hlava, ramená, stred medzi ramenami, lakty, zápästia, ruky.

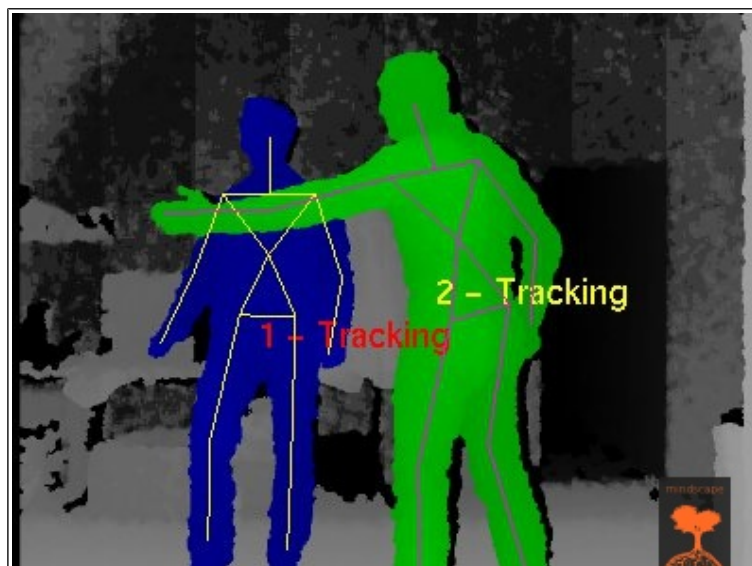


Obrázok 2.8: Režimy skeletonizácie (vľavo základný mód) [24]

## OpenNI + NITE

*OpenNI* je nezisková organizácia vedená priemyslom založená k certifikovaniu a podpore kompatibility a interoperability zariadení pre prirodzenú interakciu s človekom a k nim patriacich aplikácii a middlevéru. Jedným z cieľov organizácie je urýchlenie zavádzania takýchto zariadení na trh. Pod záštitou organizácie bol vyvinutý OpenNI framework, ktorý bude priblížený v ďalšom odseku.

OpenNi framework je viacjazyčný, multiplatformný a open-source framework, ktorý definuje aplikačné užívateľské rozhranie pre tvorbu programov využívajúcich prirodzenú interakciu. Toto aplikačné rozhranie umožňuje komunikáciu priamo so senzormi, tak ako aj s middlevérom, ktorý existuje k senzorum. Takýmto middlevérom pre senzor Kinect je NITE od spoločnosti PrimeSense. Kombinácia týchto dvoch programových vybavení zaručuje prístup k základným tokom dát zo senzorov (RGB, hĺbková mapa i IR), tak ako aj segmentáciu scény podľa detekovaných užívateľov a skeletonizáciu (obrázok 2.9). Knižnica nešpecifikuje parametre základných tokov dát, je potrebné si ich zistiť pomocou funkcie *getSupportedVideoModes*. Zdroje: [4] [5] [6]



Obrázok 2.9: Ukážka skeletonizácie v *OpenNI* + *NITE* [14]

## OpenKinect

*OpenKinect* je otvorená komunita ľudí zaujímavých sa o využívanie senzoru Kinect na osobných počítačoch a iných zariadeniach. Komunita pracuje na open-source knižniciach, ktoré umožňujú používať Kinect na systémoch Windows, Linux a Mac. Primárne sa komunita zameriava na vývoj knižnice *libfreenect* v jazyku C. V súčasnosti sú k dispozícii jazykové obálky pre nasledovné programovacie jazyky: Python, Actionscript, C++, C#, Java JNI, Java JNA Javascript, Common Lisp. Knižnica obsahuje aplikačné programové rozhranie na dvoch úrovniach. Vysokoúrovňové rozhranie poskytuje funkcie pre ovládanie a získavanie stavu LED diódy, ovládanie a získavanie stavu motorčeka, ktorý nastavuje pozíciu hlavného zariadenia na odseku a získavanie RGB a hĺbkového toku dát. Zdroj: [7]

## 2.3 Zhrnutie

Každá z dostupných knižníc disponuje určitými výhodami. Knižnica *OpenKinect* podporuje široké spektrum programovacích jazykov, avšak tento projekt bude postavený na jazyku C++, ktorý podporujú všetky tri knižnice. Nevýhodou knižnice *OpenKinect* je chýbajúca podpora skeletonizácie, zatiaľ čo knižnice *Kinect for Windows SDK* a *OpenNI* touto funkčnosťou disponujú. Výhodou *OpenNI* oproti knižnici od spoločnosti Microsoft je multiplatformnosť, na druhej strane výhodou knižnice pre systémy Windows je výrazne kvalitnejšia dokumentácia.

## 3 Rozhrania založené na gestách rúk

### 3.1 Gestá

Rozpoznávanie gest je oblasťou v informačných technológiách, ktorá má za cieľ interpretáciu ľudských gest matematickými algoritmi. Gestá môžu pochádzať z akéhokoľvek pohybu alebo stavu tela, ale najčastejšie pochádzajú z tváre alebo rúk. Predmetom rozpoznávania sú ľudské emócie vyjadrené gestami tváre a rúk, znaková reč, chôdza, proxemika<sup>3</sup> a ľudské správanie ako celok.

Detekcia gest môže byť chápaná ako cesta počítačov k porozumeniu reči tela, čo buduje bohatší komunikačný most medzi ľuďmi a strojmi ako textové alebo grafické rozhrania. V počítačových rozhraniach rozlišujeme dva typy gest:

- offline gestá – sú spracované po užívateľskej interakcii s objektom (napr. spustenie menu gestom),
- online gestá – slúžia na priamu manipuláciu s objektmi (napr. rotácia objektu).

Podľa dynamickosti môžeme gestá kategorizovať nasledovne:

- statické gestá – gesto chápeme ako statickú pózu tela alebo jeho časti (napr. vzpriamený postoj),
- dynamické gestá – gesto chápeme ako dynamickú zmenu polohy alebo orientácie tela alebo jeho časti (napr. mávanie rukou).

Nasledujúci zoznam obsahuje príklady praktického využitia rozpoznávania gest:

- rozpoznávanie znakovkej reči – preklad znakovkej reči do textovej podoby.
- asistenčná robotika – pomoc pri rehabilitácii s využitím rôznych senzorov.
- priama indikácia ukazovaním na objekty – určenie kontextu vyjadrení alebo inštrukcií.
- ovládanie gestami tváre – pohyb kurzora a zaostrenie na objekty sledovaním pohybu očí.
- virtuálne ovládanie hier – interaktívnejšie a realistickejšie ovládanie hier.
- vzdialené ovládanie – ovládanie televízie a iných zariadení.
- detekcia pocitov a emócií – identifikácia emocionálnych prejavov.

Zdroje: [15] [16]

### 3.2 Gestá rúk

Ďalej sa bude práca zaoberať už len výhradne podmnožinou gest, ktorou sú gestá rúk. Tie boli počas existencie ľudstva vždy jedným z najbežnejších a najprírodzenejších komunikačných prostriedkov. Výskum rozpoznávania gest rukami získal veľa pozornosti vďaka ich použiteľnosti v interaktívnych rozhraniach medzi človekom a počítačom a virtuálnych prostrediach. Kľúčovým problémom interakcie gestami je chápanie ľudských gest počítačom. Gestá rúk môžeme zo statického hľadiska chápať ako stav alebo z dynamického hľadiska ako zmeny stavov nasledujúcich parametrov ruky:

- polohy prstov,
- orientácia ruky,

---

3 druh neverbálnej komunikácie spočívajúci vo vyjadrení vzťahu medzi ľuďmi prostredníctvom vzdialenosti, ktorú ľudia od seba pri komunikácii udržujú

- poloha ruky,
- kombinácia predchádzajúcich parametrov.

Metódy snímania týchto parametrov môžeme rozdeliť na:

- metódy založené na elektronických senzorových rukaviciach,
- metódy založené na počítačovom videní.

Metódy založené na rukaviciach vyžadujú od užívateľa mať na ruke nasadené nešikovné zariadenie spravidla s niekoľkými káblami pripájajúcimi zariadenie k počítaču a umožňujú väčšinou výlučné snímanie polôh prstov. Naopak techniky postavené na počítačovom videní dokážu rozpoznať všetky parametre a sú pohodlnejšie. Tieto metódy môžeme ďalej rozdeliť na:

- metódy využívajúce farebnú kameru,
- metódy využívajúce hĺbkový senzor.

Metódy využívajúce farebnú kameru dokážu snímať všetky parametre ruky (polohu ruky často len v 2D), avšak dosiahnutie univerzálnej použiteľnosti je veľmi obtiažne. Týmto metódam je venovaná podkapitola 3.3. Naproti tomu metódy využívajúce hĺbkový senzor majú pri detekcii polôh prstov a orientácie ruky problémy s presnosťou, avšak spoľahlivo a presne dokážu sledovať polohu ruky v 3D a ich univerzálnosť je vysoká. Keďže táto práca sa špecializuje výhradne na tieto metódy, problematika z hľadiska hĺbkových senzorov je popísaná v kapitole 2 a problematika z algoritmického hľadiska je popísaná v podkapitolách 3.4 a 3.5. Pre dosiahnutie komplexného a zároveň univerzálneho systému je možné metódy vhodne kombinovať. Zdroje: [8] [12]

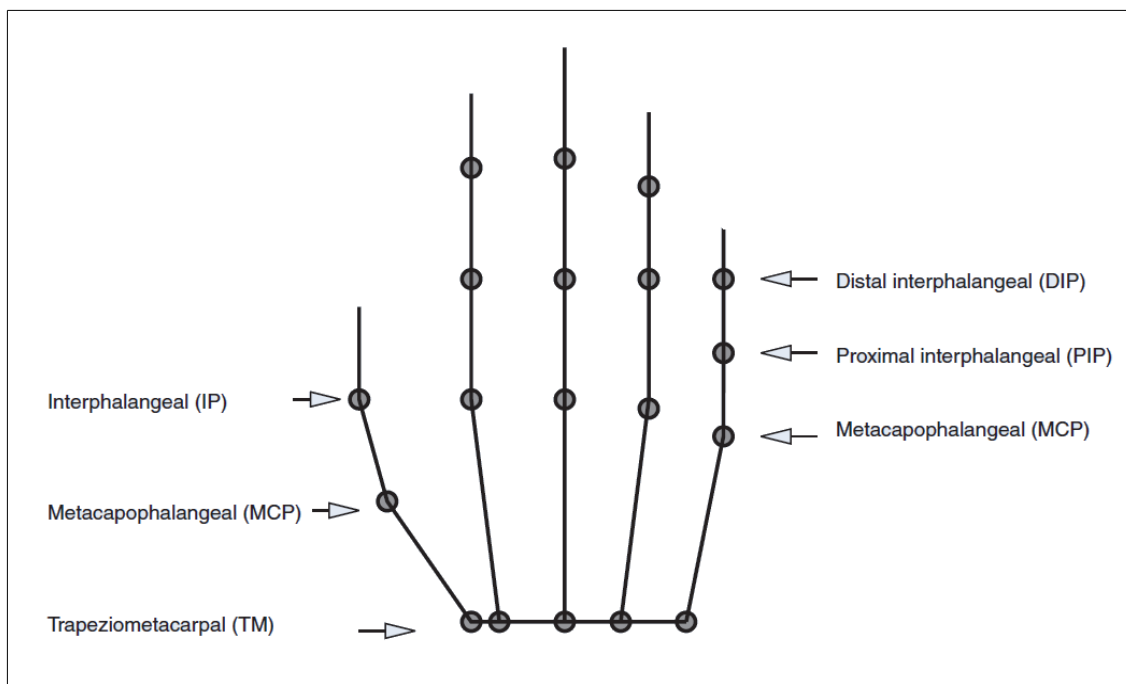
### 3.3 Gestá rúk snímané farebnou kamerou

Ovládanie gestami rúk snímaných bežnou kamerou je veľmi prirodzený spôsob interakcie s počítačom, ktorý musí spĺňať nasledujúce kritéria:

- Systém musí byť invariantný k prostrediu a osvetleniu, v ktorom pracuje.
- Systém musí správne fungovať nezávisle na osobe, ktorá systém využíva.

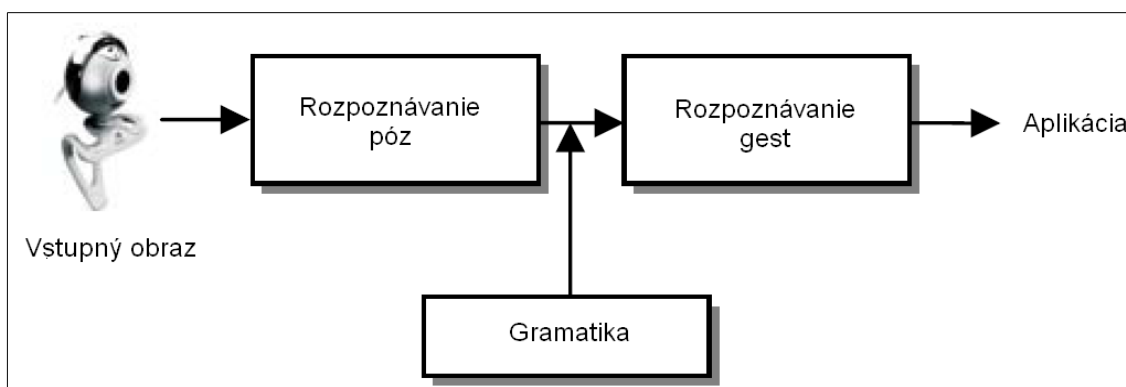
V ďalších odsekoch budú priblížené používané techniky a technológie.

Ľudská ruka má komplexnú anatomickú štruktúru (obrázok 3.1) pozostávajúcu z mnohých kĺbmi prepojených častí, umožňujúcich komplexné vzťahy medzi nimi, čo poskytuje zhruba 27 stupňov voľnosti. Vývoj užívateľských rozhraní vyžaduje znalosti anatómie ľudskej ruky, aby bolo možné určiť, aké polohy a gestá rúk sú praktické a pohodlné. Pritom ako polohu ruky označujeme statickú pozíciu ruky v priestore bez zaangažovanosti pohybu a gestom ruky rozumieme dynamický pohyb vzťahujúci sa na sekvenciu polôh ruky prepojených spojitým pohyb v krátkom časovom slede.



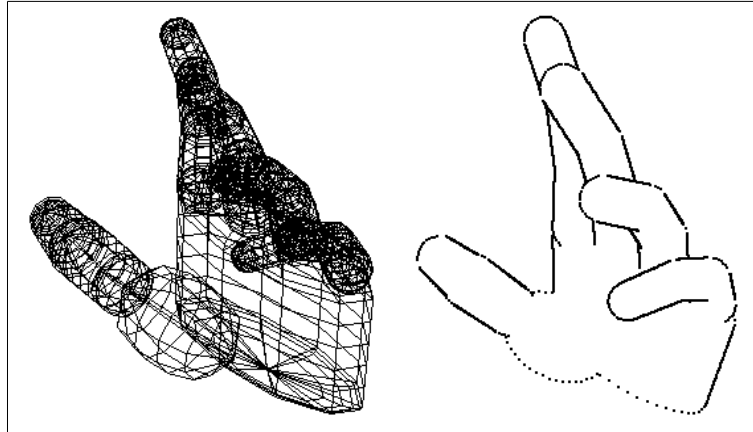
Obrázok 3.1: Štruktúra kostry ruky [13]

V systémoch rozpoznávania gest rukami, založených na počítačovom videní, je pohyb ruky nahrávaný kamerou (viacerými kamerami alebo senzormi). Záznam sa spätne rozloží na jednotlivé snímky, na ktoré môžu byť aplikované rôzne filtre, ktoré odstránia nepotrebné dáta (napríklad zvyšok tela a pozadie) a zvýrazia ruky. Takto izolované ruky sú rozpoznané a je zaznamenaná ich poloha. A keďže gestá nie sú nič iné ako sekvencia polôh ruky prepojená spojitým pohybom, rozpoznávací proces môže byť trénovaný na možnú gramatiku. S týmto prístupom môžu byť gestá rúk špecifikované ako budovanie zo skupiny polôh rúk rôznymi spôsobmi, rovnako ako sú frázy budované slovami. Proces rozpoznávania je znázornený na obrázku 3.2. V nasledujúcich odsekoch budú predstavené vybrané techniky rozpoznávania gest rúk založené na počítačovom videní.



Obrázok 3.2: Proces rozpoznávania gest rukami [12]

Prvou rozoberanou technikou je technika 3D modelu ruky (obrázok 3.3). Táto technika sa spolieha na 3D kinematický model ruky, ktorý berie do úvahy všetky stupne voľnosti. Parametre ruky sú odhadované z porovnania vstupných obrázkov a možných 2D projekcií 3D modelu ruky. Tento prístup je ideálny pre realistickú interakciu vo virtuálnych prostrediach. Jednou možnosťou tejto techniky je anatomicky presný model ruky zložený zo zrezaných kvadrík.



Obrázok 3.3: 3D model (vľavo) a z neho generovaná kontúra (vpravo) [17]

Tento model umožňuje generovanie 2D profilov elegantnými nástrojmi projekčnej geometrie a vytvorenie účinnej metódy pre prekonanie problému sebazatienenia ruky. Výhodou 3D modelového prístupu je, že ponúka bohatý popis, ktorý umožňuje zvládať širokú triedu gest rukami. Avšak, pretože 3D modely ruky sú kľbové deformovateľné objekty s mnohými stupňami voľnosti, k pokrytiu všetkých charakteristických tvarov pod rôznymi pohľadmi je potrebná veľmi rozsiahla databáza obrázkov. Ďalším spoločným problémom 3D modelov je problém extrakcie rysov a nízka schopnosť vysporiadania sa so zvláštnosťami, ktoré vznikajú pri nejednoznačných pohľadoch.

Ďalšia technika, ktorá bude priblížená, je technika založená na vzhľade. Tento prístup používa rysy obrázkov k modelovaniu vizuálneho vzhľadu ruky a následne porovnáva získané parametre rysov s rysmi získanými zo vstupného obrázka. Vzhľadom na prácu v 2D sú varianty tejto techniky časovo vysoko výkonné, avšak často sú invariantné k osvetleniu a k rôznym užívateľom. Vzhľadom k zameraniu tejto práce nie je potrebné túto techniku do detailov rozvádzať. Zdroje: [12] [17]

### 3.4 Gestá rúk a DTW

*Dynamic Time Warping* (DTW) je algoritmus na meranie podobností dvoch sekvencií, ktoré sa môžu líšiť v čase alebo rýchlosti. Časté využitie nachádza v systémoch rozpoznávania hlasu, ale v nasledujúcich odsekoch bude prezentované jeho využitie pre rozpoznávanie gest.

Reyes, Domínguez a Escalera navrhli systém rozpoznávania gest z hĺbkovej mapy založený na DTW. Pôvodný algoritmus DTW [32] bol definovaný, aby spájal časové distorzie medzi dvoma modelmi, hľadaním zarovnávejacej deformačnej cesty medzi dvoma časovými radmi.

$$Q = \{q_1, \dots, q_n\} \quad (2)$$

$$C = \{c_1, \dots, c_m\} \quad (3)$$

Pre toto zarovnanie je navrhnutá matica  $M_{m \times n}$ , ktorá má na pozícii  $(i, j)$  hodnotu vzdialenosti medzi  $c_i$  a  $q_j$ .

$$M_{ij} = |c_i - q_j| \quad (4)$$

Najčastejšie býva použitá euklidovská vzdialenosť. Potom deformačná cesta

$$W = \{w_1, \dots, w_T\}, \quad \max(m, n) \leq T < m + n + 1 \quad (5)$$

je definovaná ako sada „súvislých“ prvkov matice, ktoré definujú mapovanie medzi C a Q. Deformačná cesta je typicky podriadená niekoľkým obmedzeniam:



- Hraničné podmienky:

$$\begin{aligned} w_i &= (1, 1) \\ w_T &= (m, n) \end{aligned} \quad (6)$$

- Kontinuita:

$$\text{Ak } w_{i-1} = (a', b'), \text{ potom } w_i = (a, b), \quad a - a' \leq 1 \wedge b - b' \leq 1 \quad (7)$$

- Monotónnosť (monotónne rozprestrenie bodov W v čase):

$$w_{i-1} = (a', b'), \quad w_i = (a, b), \quad a - a' \leq 1 \wedge b - b' \leq 1 \quad (8)$$

Predmetom záujmu je finálna deformačná cesta, ktorá spĺňa uvedené podmienky a maximalizuje deformačnú cenu:

$$DTW(Q, C) = \min \left\{ \frac{1}{T} \cdot \sqrt{\sum_{i=1}^T w_i} \right\} \quad (9)$$

kde T kompenzuje rôzne dĺžky deformačných ciest. Táto cesta môže byť veľmi efektívne nájdená pomocou dynamického programovania<sup>4</sup> vyhodnotením nasledujúcej rekurzívnej definície kumulovanej vzdialenosti  $\gamma(i, j)$  ako vzdialenosti  $d(i, j)$  v aktuálnej bunke a minimum z kumulovaných vzdialeností susedných buniek:

$$\gamma(i, j) = d(i, j) + \min(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)) \quad (10)$$

Ak berieme do úvahy dynamické gesto, závislé len od sekvencie polôh ruky, je možné pomocou hľadania finálnej deformačnej cesty zistiť podobnosť dvoch gest invariante k rýchlosti vykonania gesta a takto realizovať komplexnú detekciu gest. Zdroj: [31]

### 3.5 Gestá rúk a HMM

Skryté Markovské modely (HMM<sup>5</sup>) sú široko využívané vo vede, technike a mnohých iných oblastiach (rozpoznávanie reči, bioinformatika, počítačové videnie, ekonomika). Sú to štatistické modely pre rozpoznávanie postupností (signálov). Medzi štatistické modely patrí kvôli skutočnosti, že pri ich popise sa berú do úvahy len štatistické vlastnosti signálu. Zásadným predpokladom štatistickým modelov je, že signál sa dá veľmi dobre charakterizovať ako parametrický náhodný proces a parametre stochastického (náhodného) procesu je možné vypočítať (odhadnúť) presne stanoveným postupom.

Markovský proces je náhodný proces, v ktorom sú pravdepodobnosti udalostí v budúcnosti určené poslednými udalosťami v minulosti. To znamená, že vzdialená minulosť nie je dôležitá v porovnaní so znalosťou blízkej minulosti. Táto vlastnosť sa nazýva Markovská vlastnosť.

HMM je vo svojej podstate variant konečného stavového automatu [36], ktorý obsahuje priestor skrytých stavov **Q**, priestor pozorovaní (výstupov) **O**, pravdepodobnosti prechodov **A**, výstupné (emisné) pravdepodobnosti **B** a pravdepodobnosti počiatočného stavu **π**. Aktuálny stav nie je pozorovateľný (preto „skryté“ Markovské modely). Každý stav produkuje výstup s určitou pravdepodobnosťou (**B**). Zvyčajne sú stavy **Q** a výstupy **O** samozrejme a za HMM sa považuje

4 Dynamické programovanie je metóda riešenia komplexných problémov ich delením na jednoduchšie podproblémy. Podrobne v [33].

5 Z anglického Hidden Markov Models

trojica  $(A, B, \Pi)$ . Priestor pozorovaní, (menej používané i stavový priestor) môže byť, nielen diskrétny, ale i spojitý. Táto práca sa špecifikuje výhradne na diskrétny skryté Markovské modely.

Formálna definícia:

- Skryté stavy:

$$Q = \{q_i\}, \quad i = 1, \dots, N \quad (11)$$

- Pravdepodobnosti prechodov:

$$A = \{a_{ij} = P(q_j : t+1 | q_i : t)\} \quad (12)$$

kde:

- $P(a|b)$  je podmienená pravdepodobnosť javu „a“, daná pravdepodobnosťou javu „b“
- $t = 1, \dots, T$  je čas
- $q_i, q_j \in Q$

- Pozorovania (symboly, výstupy):

$$O = \{o_k\}, \quad k = 1, \dots, M \quad (13)$$

- Výstupné pravdepodobnosti:

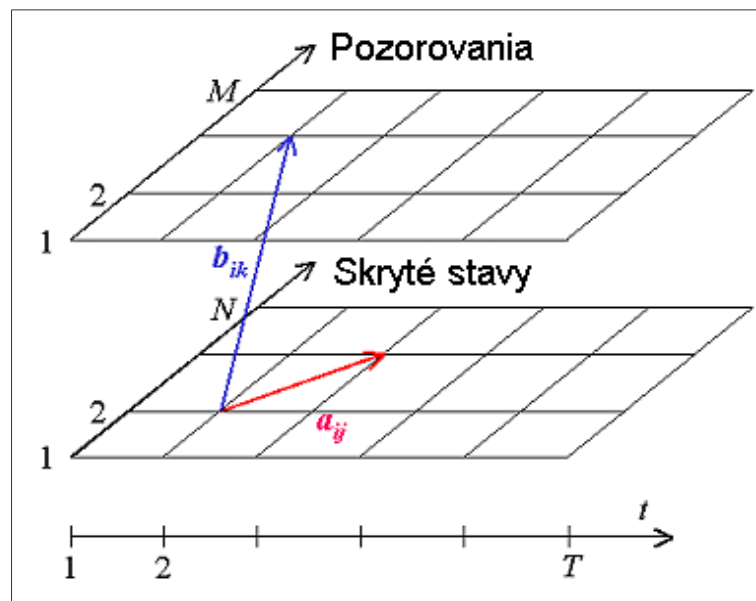
$$B = \{b_{ik} = b_i(o_k) = P(o_k | q_i)\} \quad (14)$$

- Pravdepodobnosti počiatočného stavu:

$$\Pi = \{p_i = P(q_i : t=1)\} \quad (15)$$

Model je charakterizovaný kompletnou sadou parametrov:

$$\lambda = (A, B, \Pi) \quad (16)$$



Obrázok 3.4: Skrytý Markovský model [35]

V Skrytých Markovských modeloch sú tri kanonické problémy:

- Vypočítanie pravdepodobnosti jednotlivých častí výstupnej postupnosti, pre zadané parametre modelu. Tento problém rieši algoritmus Forward-Backward (algoritmus 3.1) [40].
- Nájdenie najpravdepodobnejšej sekvencie (skrytých) stavov, ktoré mohli vygenerovať danú výstupnú postupnosť, pre zadané parametre modelu. Tento problém rieši algoritmus Viterbi (algoritmus 3.2) [40].

III. Nájdenie najpravdepodobnejšej sady prechodových a výstupných pravdepodobností pre danú výstupnú sekvenciu. Tento problém rieši viacero algoritmov a medzi nimi algoritmy Baum-Welch (BW) a Viterbi Path Counting (VPC) (algoritmus 3.3) [39].

### Algoritmus 3.1 - Forward-Backward

**Vstup:**

- $\lambda = (A, B, \Pi)$  – Skrytý Markovský Model so stavovým priestorom veľkosti  $N$ ,
- $O$  – sekvencia pozorovaní dĺžky  $T$ .

**Výstup:**

- $P(O|\lambda)$  – pravdepodobnosť sekvencie pozorovaní pre zadaný model.

**Metóda:**

1. Inicializácia

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (17)$$

2. Indukcia

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}), \quad 1 \leq i \leq T-1, \quad 1 \leq j \leq N \quad (18)$$

3. Terminácia

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (19)$$

### Algoritmus 3.2 - Viterbi

**Vstup:**

- $\lambda = (A, B, \Pi)$  – Skrytý Markovský Model so stavovým priestorom veľkosti  $N$ ,
- $O$  – sekvencie pozorovaní dĺžky.

**Výstup:**

- $q^*$  – najpravdepodobnejšia sekvencia stavov modelu pre zadané pozorovania.

**Metóda:**

1. Inicializácia

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned} \quad (20)$$

2. Rekurzia

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T \\ \psi_t(j) &= \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 1 \leq j \leq N \end{aligned} \quad (21)$$

3. Terminácia

$$\begin{aligned} p^* &= \max_{1 \leq i \leq N} [\delta_T(i)] \\ q_T^* &= \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \end{aligned} \quad (22)$$

4. Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1 \quad (23)$$

### Algoritmus 3.3 - Viterbi Path Counting

#### Vstup:

- $\{O_1, O_2, \dots\}$  – množina trénovacích sekvencií pozorovaní,
- $N$  – veľkosť stavového priestoru modelu,
- počet trénovacích kôl.

#### Výstup:

- $\lambda = (A, B, \Pi)$  – skrytý Markovský Model, ktorý s čo najvyššou pravdepodobnosťou (na zadaný počet kôl) dokáže produkovať vstupné sekvencie pozorovaní.

#### Metóda:

##### 1. Inicializácia

- a) Vytvoria sa čítacie matice  $A_c, B_c, \pi_c$ .
- b) Hodnoty matíc sa nastavujú na 0, ak korešpondujúce elementy matíc  $A, B, \pi$  žiadaného modelu sú nulové, inak sa elementy nastavujú na užívateľom špecifikovanú inicializačnú hodnotu  $c_0$ , pričom vyššia hodnota  $c_0$  znamená vyššiu váhu inicializačným hodnotám čítačov oproti novým hodnotám.

##### 2. Rekúzia

pre všetky trénovacie kolá:

pre všetky trénovacie sekvencie:

- Vytvoria sa dočasné normalizované čítacie matice a to nasledovne:

$$\begin{aligned} A_n &= \text{norm} \left( A_c \cdot \frac{\max(A_c)}{rf} \cdot \text{model} \cdot \text{rand}(N, N) \right) \\ B_n &= \text{norm}(B_c) \\ \pi_n &= \text{norm}(\pi_c) \end{aligned} \quad (24)$$

Kde  $rf$  je faktor, ktorý určuje mieru náhodnosti, ktorá je pridávaná do čítacej matice prechodov, aby sa zvýšila flexibilita hľadania najlepšej cesty, vyhýbaním sa pastiam lokálnych miním a **model** je maska žiadaného modelu.

- Na nájdenie najpravdepodobnejšej sekvencie stavov  $w$  pre aktuálnu sekvenciu pozorovaní, sa použije algoritmus Viterbi s dočasnými normalizovanými maticami  $A_n, B_n, \pi_n$ .
- Inkrementuje sa čítacia matica  $\pi_c$  na pozícií  $w(1)$ , čím sa zvýši pravdepodobnosť počiatočného stavu pre prvý stav aktuálnej najpravdepodobnejšej sekvencie stavov.
- Inkrementuje sa čítacia matica  $B_c$  na pozícií  $w(1), O(1)$ , čím sa zvýši pravdepodobnosť emisie prvého symbolu sekvencie pozorovaní pri prvom stave sekvencie stavov.
- Prejde sa celá sekvencia stavov a inkrementuje sa matica  $A_c$  na pozícií  $w(t), w(t+1)$ , teda pre každý prechod modelu do nového stavu a matica  $B_c$  na pozícií  $w(t), O(t)$ , teda pre každý výstupný symbol pri danom stave.

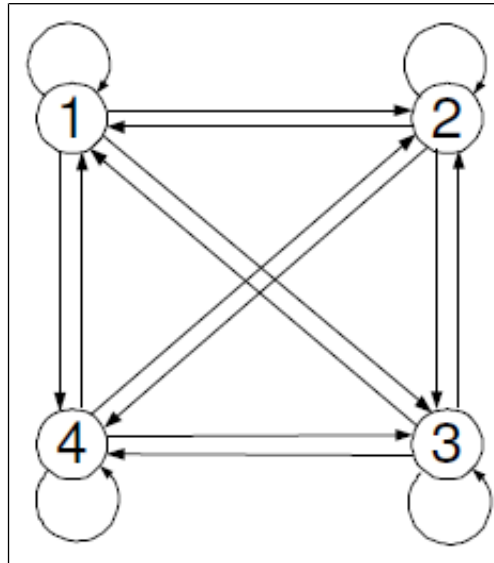
##### 3. Terminácia

Výsledkom sú normalizované čítacie matice, teda:

$$\lambda = (\text{norm}(A_n), \text{norm}(B_n), \text{norm}(\pi_n)) \quad (25)$$

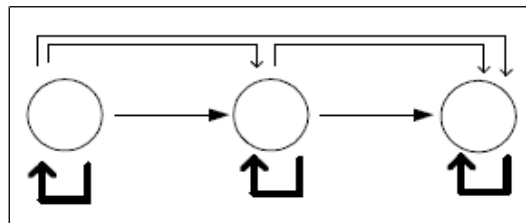
Všetky spomenuté základné algoritmy a prístupy sú taktiež detailne popísané v [35]. V praxi [41] [42] sa používajú tri typy modelov štruktúr HMM, ktoré závisia od možností prechodov medzi stavmi HMM:

- *Fully Connected*<sup>6</sup> (FC)



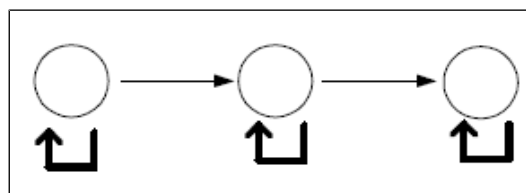
Obrázok 3.5: Fully Connected HMM [41]

- *Left-Right*<sup>7</sup> (LR)



Obrázok 3.6: Left-Right HMM [42]

- *Left-Right Banded*<sup>8</sup> (LRB)



Obrázok 3.7: Left-Right Banded HMM [42]

Ako je názorne ukázané na obrázku 3.5, *Fully Connected* modely umožňujú prechody medzi všetkými stavmi HMM, zatiaľ čo *Left-Right* modely neumožňujú prechody do predchádzajúcich stavov (obrázok 3.6) a teda priestor pozorovaní by nemal byť väčší ako počet stavov HMM. Počet nasledujúcich stavov, do ktorých je z aktuálneho stavu prechod povolený sa nazýva stupeň LR

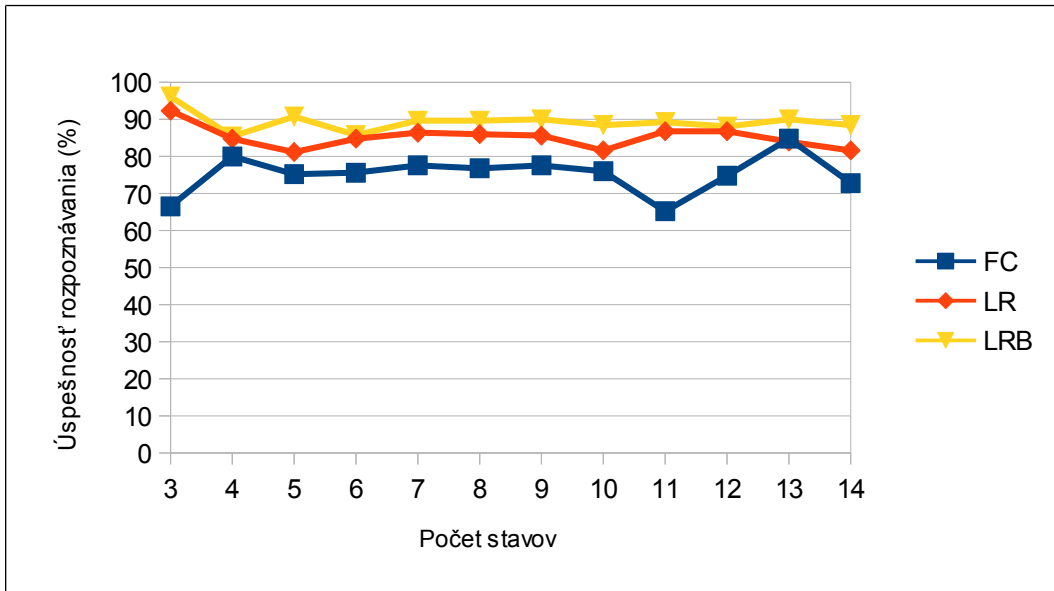
6 V preklade: plne prepojené.

7 V preklade: „zľava doprava“

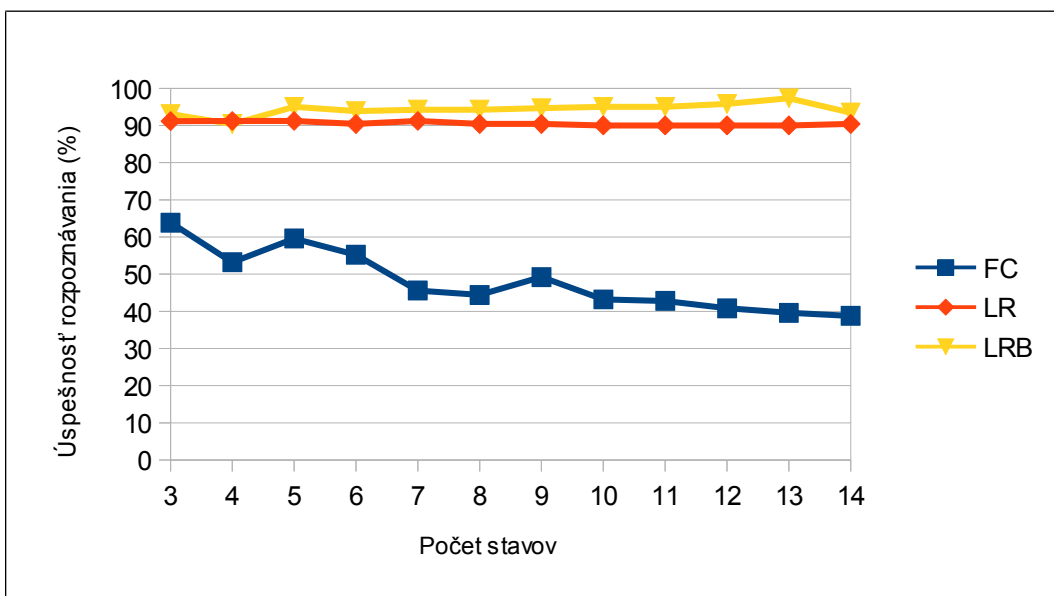
8 V preklade: „zľava do prava zväzkový“

modelu. Špeciálnym prípadom sú práve LRB modely (obrázok 3.7), ktoré sú v podstate LR modelmi so stupňom 1 a teda sú umožnené len prechody do aktuálneho a nasledujúceho stavu.

Využitelnosť jednotlivých typov HMM závisí od problematiky, ktorá sa Markovskými modelmi modeluje. Na základe testov úspešnosti rozpoznávania gest, založeného na HMM [41], sú na obrázkoch 3.8 a 3.9 porovnania týchto typov, s využitím tréningových algoritmov Baum-Welch a Viterbi Path Counting. Z oboch grafov vyplýva, že pre detekciu gest sú najvhodnejšie LRB modely.

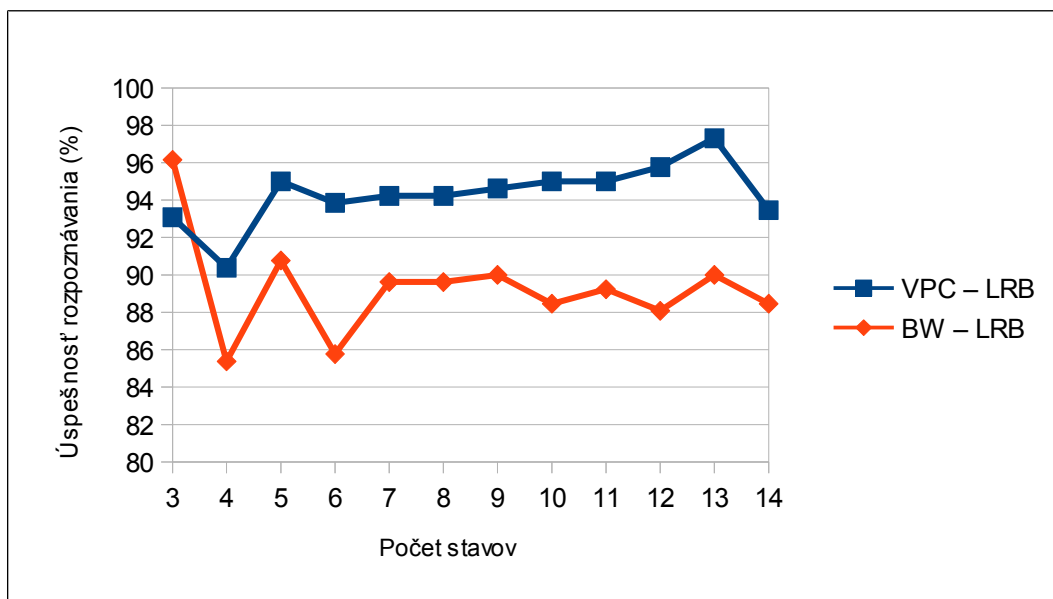


Obrázok 3.8: Porovnanie modelov pre BW



Obrázok 3.9: Porovnanie modelov pre VPC

Ďalej na obrázku 3.10 sú na základe rovnakých dát porovnané spomenuté tréningové algoritmy s využitím LRB modelov, pričom algoritmus VPC sa javí ako výkonnejší.



Obrázok 3.10: Porovnanie VPC a BW

Podobne ako pri DTW, ak berieme do úvahy dynamické gesto, závislé len od sekvencie polôh ruky, môžeme tieto polohy stotožniť s výstupnými symbolmi HMM, pomocou ktorých môžeme najskôr Markovský model vhodného typu (LRB) trénovať vhodným algoritmom (VPC) a neskôr pomocou algoritmu Forward-Backward testovať model gesta a tým zisťovať podobnosť modelovaného gesta s iným gestom. Zdroje: [34] [35]

### 3.6 Existujúce využitia rozhraní pre gestá rúk

Pre inšpiráciu k ďalšej práci bol vykonaný malý prieskum existujúcich využití rozhraní pre gestá rúk a tieto praktické využitia sú stručne opísané nasledujúcom texte.

Prvým príkladom praktického využitia rozhrania pre gestá rúk, je virtuálne prostredie Battlefield, navrhnuté Berrym, kde sú ovládacie gestá využité nielen k navigácii vo virtuálnom prostredí, ale aj k výberu a pohybovaniu virtuálnych objektov. [18]

Ďalším príkladom je virtuálne prostredie umožňujúce interaktívne modelovanie biopolymérov spájaním 3D grafických modelov molekúl. Prostredie taktiež poskytuje program pre dynamické simulácie s využitím gest rúk. [19]

Perspektívnou oblasťou použitia pre ovládanie gestami sú taktiež motorové vozidlá. Konkrétne boli navrhnuté systémy ovládania sekundárnych systémov auta gestami. Tieto systémy môžu mať významný vplyv na bezpečnosť, pretože ovládanie gestami upútava vodičovú pozornosť vo významne menšej miere ako zaužívané ovládanie pomocou tlačidiel. [20] [21]

Obrovský význam majú bezdotykové technológie v oblasti starostlivosti o ľudské zdravie, kde hrozí pri dotykovom ovládaní kontaminácia. V roku 2008 bol vyvinutý systém pre prehliadanie radiologických snímok ovládaný gestami rúk, ktorý je dokonale sterilný a je vynikajúcou náhradou ovládania pomocou dotykových obrazoviek, prípadne iných rozhraní. [22]

Poslednou zmienkou aplikáciou je Toshiba Media Center, ktorá v spojení s notebookmi Qosmio umožňuje gestami rúk prehrávať a zastavovať prehrávanie audia i videa, ovládanie kurzoru myši zatvorenou päťou i klikanie vztýčením palca. [23]

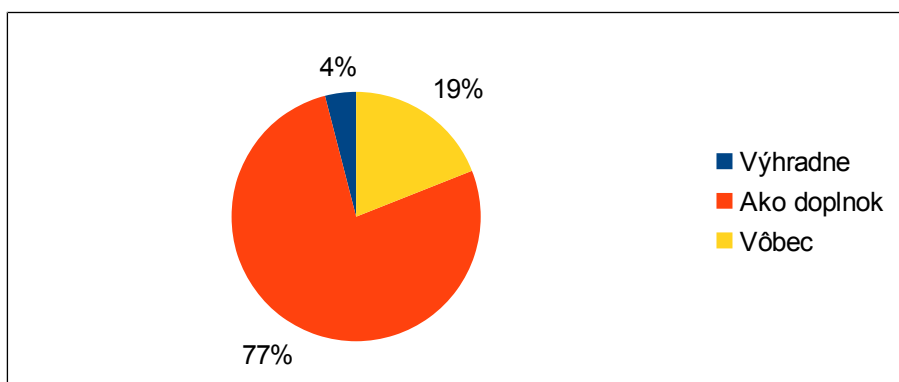
## 4 Návrh demonštračnej aplikácie

Po technologických rozpravách je v rámci cieľov tejto práce potrebné navrhnuť demonštračnú aplikáciu, ktorá bude obsahovať rozhranie pre prácu s multimediálnym obsahom založené na gestách rúk.

### 4.1 Užívateľské štúdie

Prvým krokom návrhu sú užívateľské štúdie týkajúce sa základných aspektov navrhovanej aplikácie. Cieľom štúdií je posun práce bližšie k rovine reálneho využitia a obmedzenie vplyvu subjektívnych názorov autora. Prieskumu týkajúceho sa základných otázok sa zúčastnilo 67 respondentov vo veku od 19 do 50 rokov. Väčšiu časť z nich tvorili počítačovo zdatní používatelia, ale nebola zanedbaná ani laická menšina. Ďalšie nejasnosti, ktoré vznikli až v priebehu pokročilého návrhu a implementácie boli s užívateľmi priamo konzultované a bude o nich vždy v rámci danej problematiky pojednávané.

Prvá, najzákladnejšia a zároveň najkritickejšia otázka prieskumu zisťovala záujem respondentov o navrhované bezdotykové užívateľské rozhranie založené na ovládaní gestami rúk. Zisťovalo sa, či by užívatelia uvítali rozhranie ovládané výhradne gestami rúk, alebo ako doplnok konvenčného rozhrania, prípadne vôbec.

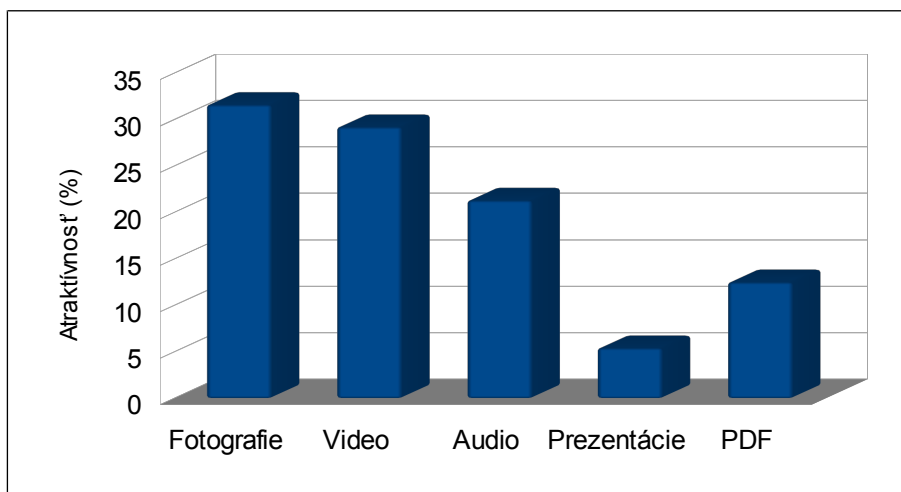


Obrázok 4.1: Záujem o rozhranie ovládané gestami rúk

Výsledky (obrázok 4.1) sú pre ďalšiu prácu povzbudivé, pretože významná väčšina ľudí o nové rozhranie má záujem, avšak väčšina zväčša len ako doplnok klasického rozhrania. Návrh a implementácia konvenčného rozhrania ako súčasť navrhovanej aplikácie je teda viac ako vhodný.

Ďalším otáznikom projektu bolo: S akým multimediálnym obsahom by malo užívateľské rozhranie manipulovať, resp. pre akú multimediálnu aplikáciu je ovládanie gestami rúk vhodné, či atraktívne? Na túto otázku odpovie graf na obrázku 4.2.

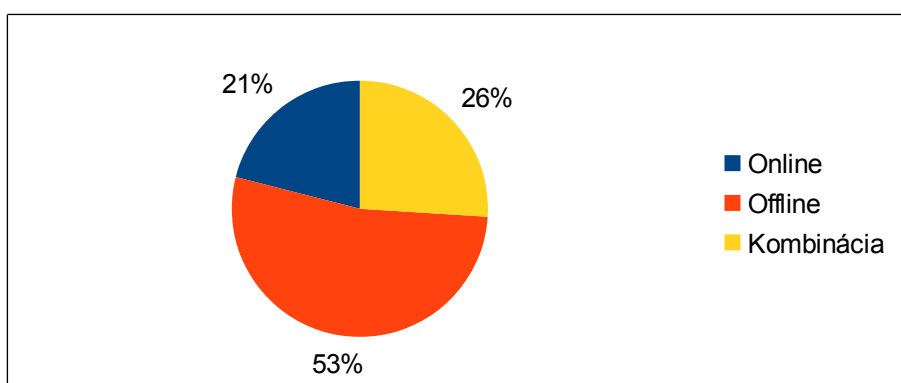




Obrázok 4.2: Atraktivnosť multimediálneho obsahu

Z grafu je zrejmé, že najatraktívnejším multimediálnym obsahom z hľadiska bezdotykového ovládania je relatívne vyrovnaná trojica fotografie, video a audio a bolo by výzvou navrhnúť a realizovať rozhranie pre všetky tri typy.

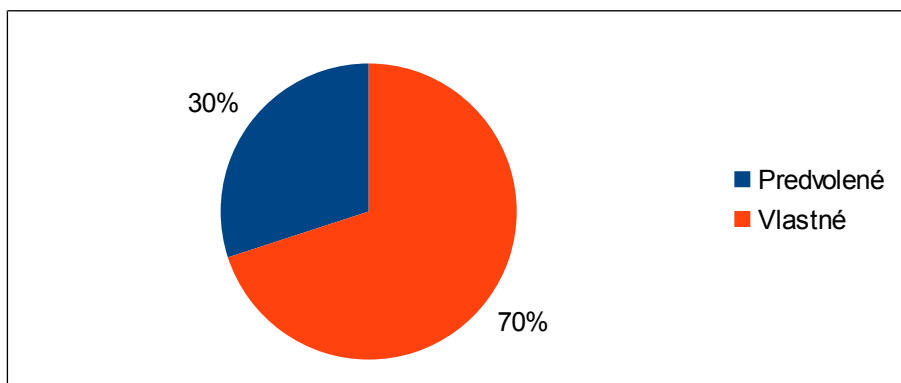
Ďalším nejasným aspektom rozhrania bol konkrétny spôsob interakcie užívateľa s objektmi rozhrania. Ako je uvedené v predchádzajúcej kapitole, počítačové užívateľské rozhrania rozlišujú dva typy gest a to online a offline gestá. Otázkou pre užívateľov bolo teda, či pokladajú za vhodnejšiu a pohodlnejšiu priamu, nepriamu, či dokonca kombinovanú interakciu s multimediálnymi objektami typu fotografie, video a audio.



Obrázok 4.3: Online vs. offline gestá

Výsledky (obrázok 4.3) sú prekvapivé. I keď väčšina opýtaných priznala, že priama interakcia s obsahom je zaujímavá, nakoniec usúdili, že „hranie sa“ s rozhraním by ich po chvíli omrzelo a k slovu by sa dostala realita a s ňou tvrdá kritika každého zbytočného pohybu. Najvhodnejším riešením v prípade tohto projektu budú zrejme jednoduché offline gestá.

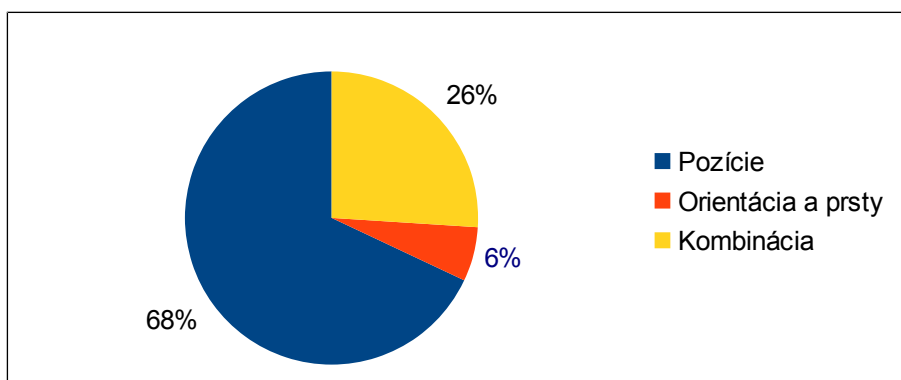
Z odpovede na druhú otázku sa však priamo núka tretia. Ovládanie offline gestami vyžaduje priradenie konkrétnych gest ku konkrétnym akciám, ktoré sú gestom spustené. Nie je ale zrejmé, či by užívatelia preferovali intuitívne predvolené a nemenné gestá, ktoré by nemuseli konfigurovať, či nahrávať, alebo by dali prednosť gestám, ktoré si sami nahrávajú alebo nakonfigurujú. Pohľad používateľov na tento kompromis medzi vynaloženým úsilím a pohodlím priblíži graf na obrázku 4.4.



Obrázok 4.4: Predvolené vs. vlastné gestá

Z výsledkového grafu a zdôvodnení uvedených v prieskume je zrejmé, že okrem s počítačom menej zručných užívateľov, ktorí majú obavy čokoľvek konfigurovať, vysoká väčšina opýtaných preferuje používanie vlastných gest a preto je takmer nevyhnutné, aby budúce rozhranie túto funkčnosť umožňovalo.

Poslednou informáciou, ktorú prieskum mal za úlohu zistiť, bolo: Aký názor má užívateľská komunita na zahrnutie sledovania pozícií jednotlivých prstov do offline gest rukami. Na výber boli nasledovné tri možnosti: gestá závislé len na pozícii rúk v priestore, gestá závislé na polohe jednotlivých prstov a orientácii ruky v priestore a nakoniec gestá závislé na kombinácii oboch predchádzajúcich faktoroch, teda na pozícii rúk v priestore i orientácii rúk a polohe prstov.



Obrázok 4.5: Preferovaná podstata offline gest rukami

Respondenti sa vyjadrili jasne (obrázok 4.5), sledovanie prstov považujú pri offline gestách za zbytočné, únavné a zmätočné a nedoporučujú ich do rozhrania zahŕňať.

Z užívateľského prieskumu vyplynulo viacero užitočných zistení, ktoré budú použité pri ďalšom návrhu.

## 4.2 Návrh konvenčného rozhrania

V súlade s výsledkami užívateľského prieskumu bude aplikácia obsahovať tri samostatné moduly pre prácu s multimediálnym obsahom a to: modul pre prácu s obrázkami, modul pre prácu s videom a modul pre prácu s audiom. Každý modul bude umožňovať základné operácie nad preň relevantným typom dát. Takže okrem neodmysliteľných funkcií ako je samotné zobrazovanie obrázkov, či prehrávanie audia a videa sa od modulov vyžadujú operácie ako napríklad prechod na ďalší

obrázok, či zoom pre modul pre prácu s obrázkami a zmenu hlasitosti, či zastavenie prehrávania pre ďalšie dva moduly. Ďalšie upresnenie podporovaných operácií sa očakáva počas testovania použiteľnosti implementácie.

Keďže podľa prieskumov by konvenčné rozhranie nemalo byť zanedbané, všetky tieto operácie budú prepojené i s klasickým rozhraním založeným na polohovacom zariadení a klávesnici. Návrh teda predpokladá v každom module sadu klasických prvkov užívateľského rozhrania ako sú tlačidlá, posuvníky, vstupné polia a podobne. Vzhľadom na predpokladanú jednoduchosť tohto rozhrania nie je detailný návrh potrebný. Ďalej sa predpokladá konfiguračný modul pre konfiguráciu a komunikáciu s rozhraním pre gestá rúk, ktorý je však nutne závislý od návrhu samotného rozhrania pre gestá.

## 4.3 Návrh rozhrania pre gestá rúk

Analýza technologických a užívateľských štúdií týkajúcich sa rozhraní založených na gestách rúk dáva podklad pre čo najoptimálnejší návrh takéhoto rozhrania pre prácu s multimediálnym obsahom. Počiatočné smerovanie návrhu je určené práve užívateľmi a predpokladá rozhranie založené na užívateľsky definovateľných offline gestách určených výhradne pohybom ruky.

Avšak je nutné špecifikovať ďalšie vlastnosti tohto rozhrania. Keďže pomocou zariadenia *Kinect* je sledovanie rúk v 3D prostredí, je kvôli vyššej variabilite ovládania uprednostnené chápanie gesta ako pohybu ruky v priestore a nie v rovine. S týmto faktom súvisí spôsob, akým bude možné vlastné gestá definovať. Z dvoch zvažovaných možností: modelovanie gesta a zaznamenanie gesta, je zvolená možnosť nahrávania, pretože sa javí ako intuitívnejšie a modelovanie priestorového gesta, aj keď sa predpokladá vyššia presnosť, môže byť pre bežného užívateľa nebanálny problém.

Ďalej je potrebné zvážiť chápanie gesta ako pohybu jednej ruky, či oboch rúk súčasne. Variabilita obojručných gest je nepochybne vyššia, avšak na úkor pohodlnosti ovládania. Zatiaľ čo predpokladaná variabilita priestorových gest jednej ruky je dostatočná a navyše takto navrhnuté rozhranie by umožňovalo nezávislú interakciu pre obe ruky súčasne a preto navrhované rozhranie predpokladá práve interakciu na báze jednoručných gest.

Veľmi podstatnou problematikou rozhrania je spúšťanie a priebeh detekcie gest. Zvažované alternatívy s ich výhodami a nevýhodami sú nasledovné:

- Kontinuálna detekcia gest, ktorá gesto rozpozná v rámci ľubovoľného pohybu ruky.
  - Výhody: rýchlosť, maximálna užívateľská prívetivosť.
  - Nevýhody: problematická implementácia – falošné detekcie.
- Sledovanie pohybu ruky medzi dvomi definovanými statickými pózami.
  - Výhody: jednoduchšia implementácia, možnosť presného určenia počiatočnej a koncovkej polohy ruky v geste.
  - Nevýhody: zdržanie oproti kontinuálnej detekcii, nechcené spúšťanie detekcie.
- Vymedzenie vhodne navrhutej aktívnej zóny v okolí tela, v ktorej bude detekcia prebiehať.
  - Výhody: jednoduchšia implementácia.
  - Nevýhody: počiatočná a konečná poloha gesta len na hranici aktívnej zóny, nechcené spúšťanie detekcie.
- Spustenie hlasovým príkazom.
  - Výhody: môže zamedziť falošným detekciám.

- Nevýhody: nutnosť spracovania zvukového vstupu, hluk.

Navrhuje sa využitie jednej z jednoduchších alternatív (statické pózy, aktívna zóna, prípadne ich kombinácia) s možnosťou nastavenia prihlasovacieho gesta, ktoré si užívateľ môže zvoliť tak, aby bola pravdepodobnosť jeho nechcenej detekcie minimálna, pričom až po úspešnej detekcii tohto gesta bude možná detekcia ostatných gest.

Posledným pojednávaným aspektom rozhrania je otázka závislosti gest od nasledujúcich troch faktorov:

- od rýchlosti ich vykonania, prípadne zmien rýchlosti v ich vykonávaní – dynamiky gesta,
- od ich veľkosti,
- od polohy a orientácie ich vykonania.

Závislosť gesta od jeho rýchlosti a dynamiky je neintuitívna. Užívateľia očakávajú, že dve gestá rovnakej trajektórie vykonané rôznou rýchlosťou, či dynamikou, bude rozhranie považovať za rovnaké gestá. Podobne, ak užívateľ registruje svoje vlastné gesto a neskôr ho použije v inom merítke, očakáva, že rozhranie dokáže túto zmenu veľkosti prekonať a zistiť ich podobnosť.

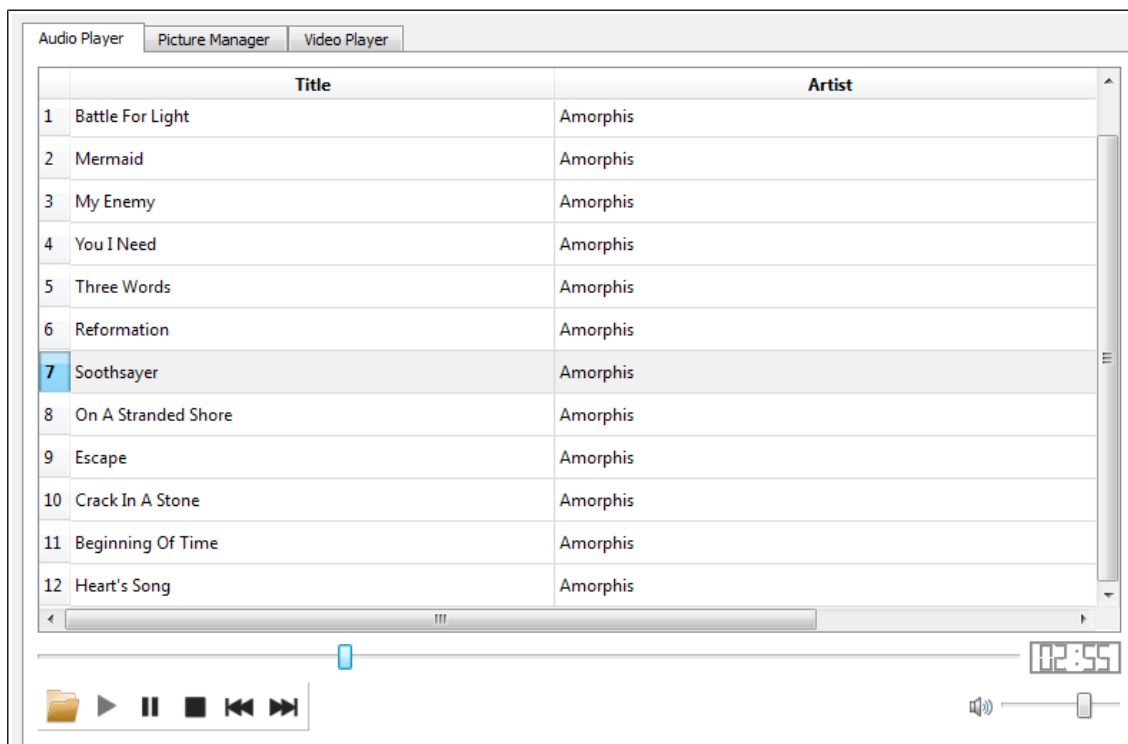
Pamätanie si presnej polohy, v ktorej bolo gesto vykonané, je užívateľsky neakceptovateľné a preto je závislosť gesta od polohy vylúčená. Na druhej strane orientácia gesta v priestore je už z intuitívneho hľadiska jeho neoddeliteľnou súčasťou a je nutné určiť k čomu sa bude orientácia gesta vzťahovať. Ak má byť systém robustný a pracovať nezávisle na polohe senzora a užívateľa, navrhuje sa vzťahovať orientáciu gesta k orientácii celého tela.

# 5 Implementácia konvenčného rozhrania

Podľa vytvoreného návrhu bola implementovaná aplikácia *Multimedia Gestures*, ktorá umožňuje prehrávanie audio a video súborov a prezeranie obrázkov a disponuje možnosťou ovládania gestami rúk. Aplikácia bola vyvinutá vo vývojovom prostredí *Qt SDK* verzia 4.7.4 [27] v jazyku C++ s využitím knižníc *OpenCV* verzia 2.4.3 [11] a *Kinect for Windows SDK* verzia 1.6 (podkapitola 2.2). Táto kapitola je zameraná na implementáciu konvenčného rozhrania aplikácie, ktoré je celé v anglickom jazyku a pre interakciu s užívateľom predpokladá využitie polohovacieho zariadenia a klávesnice, zatiaľ čo kapitola 6 sa zameriava na implementáciu rozhrania pre gestá rúk.

## 5.1 Audio prehrávač

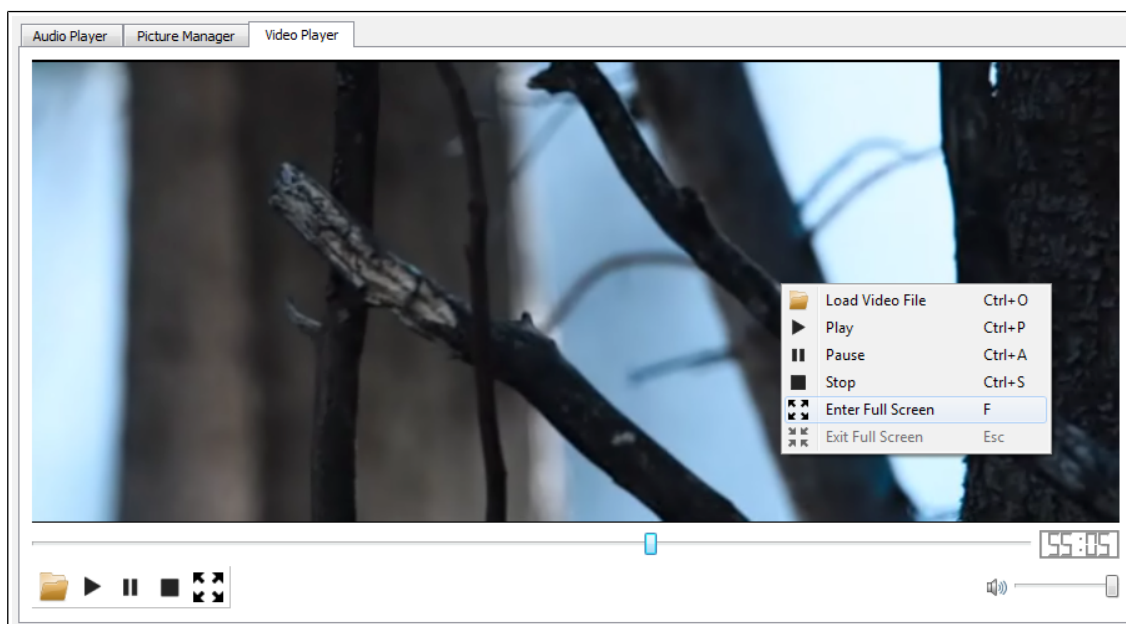
Prvým prostriedkom práce s multimediálnym obsahom je prehrávač zvukových súborov, ktorý je implementovaný triedou *AudioPlayer* s využitím multimediálnej knižnice *Qt Phonon* [30]. Ako možno vidieť na obrázku 5.1, rozhranie prehrávača sa skladá zo zoznamu skladieb a ovládacieho panelu. Ovládací panel obsahuje lištu pre skokovú zmenu aktuálneho času prehrávania skladby, displej zobrazujúci čas prehrávania skladby, tlačidlo pre pridanie súborov do zoznamu skladieb, tlačidlá pre riadenie prehrávania (*Play*, *Pause*, *Stop*, *Next*, *Prev*) a lištu pre nastavenie hlasitosti. Zoznam skladieb sa pri zatvorení aplikácie automaticky ukladá vo formáte m3u.



Obrázok 5.1: Audio prehrávač

## 5.2 Video prehrávač

Ďalším multimediálnym podprogramom je prehrávač video súborov, ktorý je implementovaný triedou *MyVideoPlayer*. Rozhranie prehrávača sa skladá zo scény prehrávania videa a ovládacieho panelu (obrázok 5.2). Ovládací panel obsahuje lištu pre skokovú zmenu aktuálneho času prehrávania videa, displej zobrazujúci čas prehrávania videa, tlačidlo pre otvorenie video súboru, tlačidlá pre riadenie prehrávania (*Play*, *Pause*, *Stop*), tlačidlo pre zapnutie celoobrazovkového režimu a lištu pre nastavenie hlasitosti.

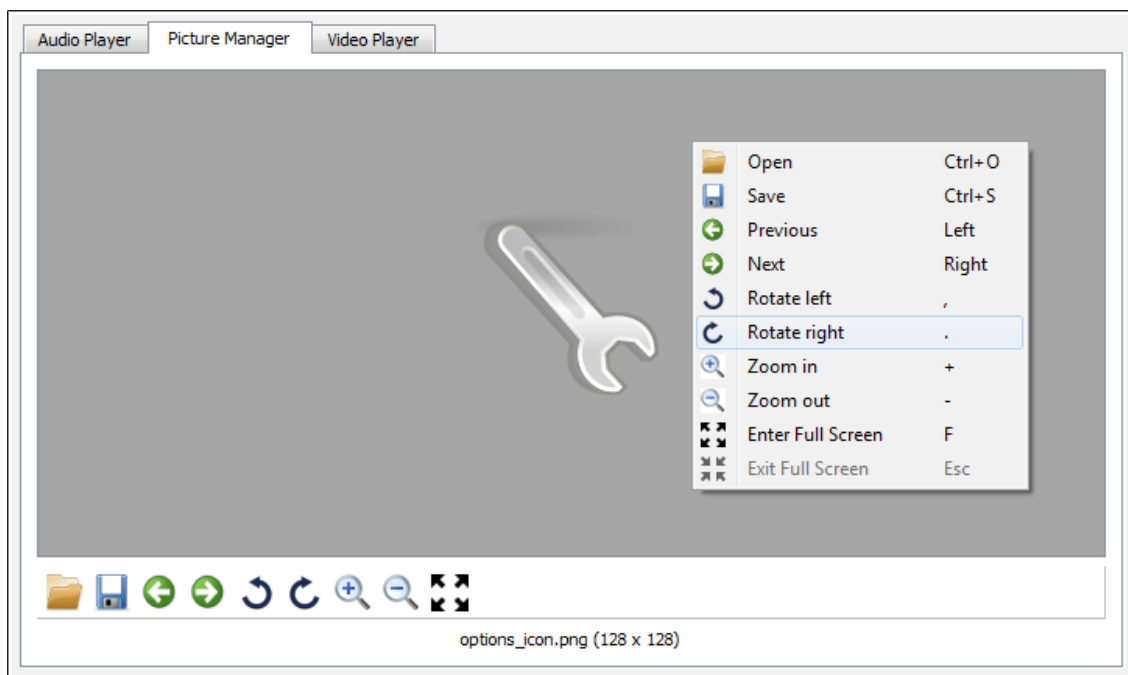


Obrázok 5.2: Video prehrávač

## 5.3 Prehliadač obrázkov

Posledným nástrojom pre prácu s multimediálnym obsahom je prehliadač obrázkov, ktorého funkčnosť je obsiahnutá v triede *PictureManager*. Rozhranie prehliadača sa skladá z 3 základných častí: scény prezerania obrázkov, ovládacieho panelu a stavovej lišty, v ktorej sa zobrazuje názov otvoreného súboru a jeho rozlíšenie (obrázok 5.3).

Ovládací panel tvoria tlačidlá, ktoré umožňujú: otvorenie obrazového súboru, uloženie obrazového súboru, zobrazenie predchádzajúceho obrazového súboru v aktuálnej zložke, zobrazenie nasledujúceho obrazového súboru v aktuálnej zložke, rotáciu obrázka o 90° v protismere hodinových ručičiek, rotáciu obrázka o 90° v smere hodinových ručičiek, priblíženie obrázka, oddialenie obrázka a vstup do celoobrazovkového režimu.

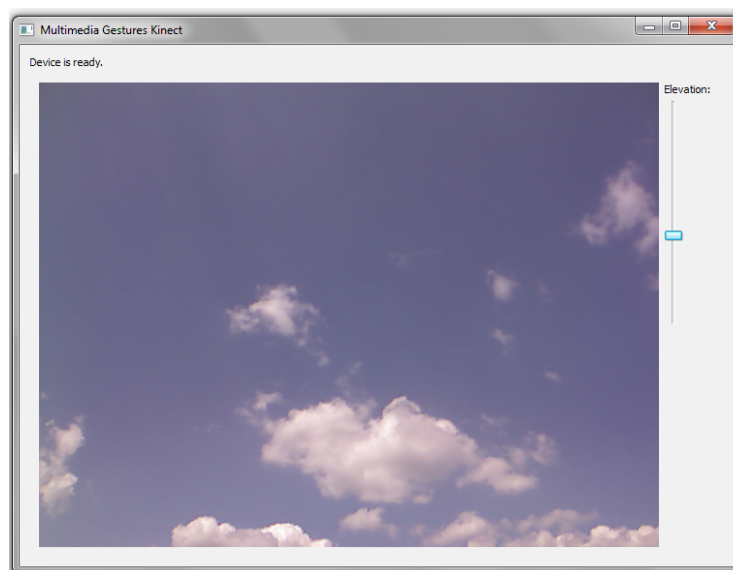


Obrázok 5.3: Prehliadač obrázkov

## 5.4 Konfiguračné rozhranie

### Okno Kinect

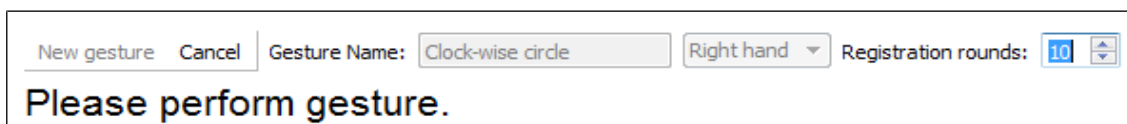
Na obrázku 5.4 je zobrazené okno, ktoré automaticky po pripojení Kinectu začne zobrazovať výstup jeho RGB kamery a obsahuje posuvník ovládajúci naklonenie zariadenia. Okno je veľmi nápomocné pri fyzickom umiestňovaní zariadenia, pre čo najlepšie snímanie scény a taktiež pri nahrávaní vlastných gest.



Obrázok 5.4: Okno Kinect

## Rozhranie registrácie gest

Toto rozhranie, ktoré je znázornené na obrázku 5.5, poskytuje obojsmernú komunikáciu so systémom registrácie gest a po pripojení zariadenia Kinect sa automaticky sprístupní. Skladá sa z ovládacieho panelu a stavového okna. Ovládací panel obsahuje vstupné pole pre zadanie názvu registrovaného gesta, panel pre výber pravej, či ľavej ruky, ktorá bude gesta realizovať a položku „Registration rounds“, ktorá určuje koľkokrát bude užívateľ gesto opakovať, aby sa vytvoril model tohto gesta. O priebehu registrácie informuje stavové okno, ktoré je pod ovládacím panelom.

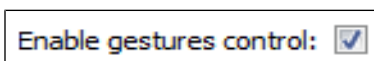


Obrázok 5.5: Rozhranie registrácie gest

## Rozhranie konfigurácie ovládania gestami

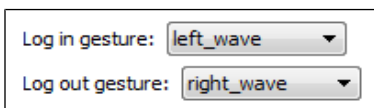
Aby bolo možné konfigurovať systém ovládania gestami, je implementované rozhranie umožňujúce vykonávať konfiguračné akcie, ktoré sú ilustrované a popísané na nasledujúcich obrázkoch a k nim patriacich popisoch.

Na obrázku 5.6 je zobrazené pole slúžiace na aktiváciu a deaktiváciu ovládania gestami.



Obrázok 5.6: Aktivácia ovládania gestami

Obrázok 5.7 ilustruje, akým spôsobom je možný výber gesta pre prihlásenie sa do systému, resp. odhlásenie sa z neho.

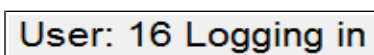


Obrázok 5.7: Voľba prihlasovacieho a odhlasovacieho gesta

Aby bolo možné informovať užívateľa o detekcii gesta, či úspešnom prihlásení a odhlásení užívateľa, rozhranie obsahuje informačné textové pole, ktoré je aj s príkladmi jeho využitia zobrazené na obrázkoch 5.8 a 5.9.



Obrázok 5.8: Správa o rozpoznanom geste.



Obrázok 5.9: Správa o prihlásení sa užívateľa do systému

Ďalej rozhranie konfigurácie ovládania gestami umožňuje nastavenie minimálnej podobnosti gesta s registrovaným gestom, aby bola detekcia gesta úspešná, nastavenie parametrov aktívnej zóny a v neposlednom rade nastavovanie spúšťačích gest pre vybrané akcie jednotlivých multimediálnych modulov programu. Všetky tieto akcie sú prehľadne vypísané v nasledujúcom zozname:



- Audio prehrávač: spustenie prehrávania, pauza, zastavenie prehrávania, prechod na ďalšiu skladbu, prechod na predchádzajúcu skladbu, zvýšenie hlasitosti, zníženie hlasitosti.
- Video prehrávač: spustenie prehrávania, pauza, zastavenie prehrávania, spustenie celoobrazovkového režimu, ukončenie celoobrazovkového režimu, zvýšenie hlasitosti, zníženie hlasitosti.
- Prehliadač obrázkov: uloženie obrázka, prechod na ďalší obrázok, prechod na prechádzajúci obrázok, rotácia obrázka doľava, rotácia obrázka doprava, priblíženie obrázka, vzdialenie obrázka, spustenie celoobrazovkového režimu, ukončenie celoobrazovkového režimu.

# 6 Implementácia rozhrania pre gestá rúk

Táto kapitola sa venuje implementácii rozhrania pre gestá rúk, obsiahnutého v aplikácii *Multimedia Gesutres*. Jadro rozhrania je založené na skrytých Markovských modeloch (podkapitola 3.5). Rozhranie v súlade s návrhom umožňuje registráciu vlastných gest a ich rozpoznávanie pre obe ruky zvlášť a pre viacerých užívateľov súčasne. Pre ovládanie aplikácie gestami je potrebné sa do rozhrania prihlásiť pomocou užívateľom zaregistrovaného a zvoleného gesta. Za gesto sa považuje každý pohyb ruky, ktorý je vykonaný v aktívnej zóne v okolí tela, pričom sa sleduje 3D poloha dlane vzhľadom k telu. Rozpoznávanie gest je invariantné k rýchlosti vykonania gesta (i zmenám rýchlosti v rámci jedného gesta), k rovnomernej zmene veľkosti gesta i k zmene polohy gesta v priestore. V nasledujúcich kapitolách je celý systém detailne popísaný.

## 6.1 Detekcia rúk bez využitia skeletonizácie

Prvým krokom implementácie rozhrania, založeného na gestách rúk je implementácia detekcie rúk. Výsledkom prvého pokusu implementácie takejto detekcie je aplikácia schopná rozpoznávať ľudské ruky priamo z hĺbkovej mapy bez využitia skeletonizácie. Aplikácia bola taktiež využitá v projekte TA2 - Together Anywhere, Together Anytime [26], na ktorom participuje Fakulta Informačných Technológií VUT v Brně. Program využíva knižnicu *OpenNI* (podkapitola 2.2) a *OpenCV*. Postupy a princípy v ňom využité budú popísané a znázornené algoritmom 6.2 a s ním súvisiacimi obrázkami.

### *Algoritmus 6.1 - Detekcia rúk z hĺbkovej mapy*

*Vstup:*

- hĺbková mapa,
- maska segmentácie.

*Výstup:*

- absolútne pozície rúk v priestore.

*Metóda:*

1. Program pomocou knižnice *OpenNI* načíta zo zariadenia hĺbkovú mapu a masku segmentácie scény. Maska segmentácie scény je obrázok v stupnici šedi, ktorého hodnota jednotlivých bodov určuje, ktoré body scény pripadajú detekovaným užívateľom (vtedy má bod masky hodnotu užívateľovho ID) a ktoré body nepripadajú žiadnemu užívateľovi (hodnota 0). Hĺbková mapa testovacej situácie je znázornená na obrázku 6.1.
2. Z hĺbkovej mapy sa vyberie oblasť, ktorá pripadá žiadanému užívateľovi. Z týchto hĺbkových bodov sa vytvorí relatívna hĺbková mapa užívateľa, kde hĺbka jednotlivých bodov bude zodpovedať ich vzdialenosti od stredu tela.

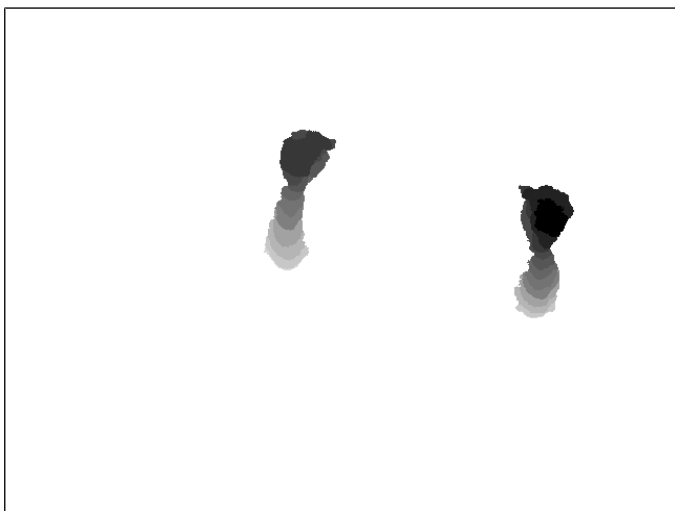
$$z_{ij}^{rel} = z_{ij}^{stred} - z_{ij} \quad (26)$$

3. Pomocou prahovania sa v relatívnej hĺbkovej mape vynulujú body blízke stredu tela. Výsledok je znázornený na obrázku 6.2.

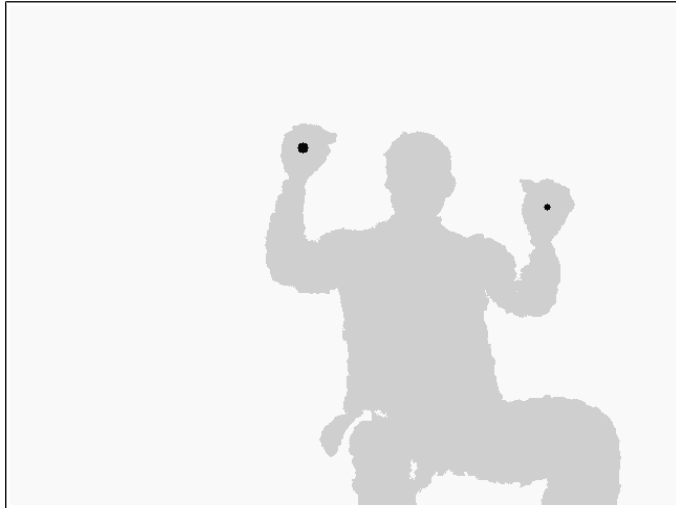
4. V relatívnej hĺbkovej mape sa hľadá extrém – bod s najväčšou vzdialenosťou od stredu tela, ktorý je pravdepodobne koncom jednej ruky. Stabilný stred ruky je získaný aritmetickým spriemerovaním súradníc najbližšieho okolia tohto bodu a prevedie sa pomocou knižnice OpenNI na priestorové súradnice.
5. Ruka je z relatívnej hĺbkovej mapy zmazaná semienkovým vyplňovacím algoritmom [37].
6. Postup detekcie stredu ruky sa opakuje pre druhú ruku (bod 4).
7. Výber pravej a ľavej ruky potrebuje kalibráciu, teda pri prvom snímku sa rozhoduje podľa súradníc v priestore. V každom ďalšom snímku sa vyberá pravdepodobnejší variant podľa predchádzajúcej pozície rúk. Výsledok detekcie prezentuje obrázok 6.3.



Obrázok 6.1: Hĺbková mapa



Obrázok 6.2: Relatívna hĺbková mapa po prahovaní



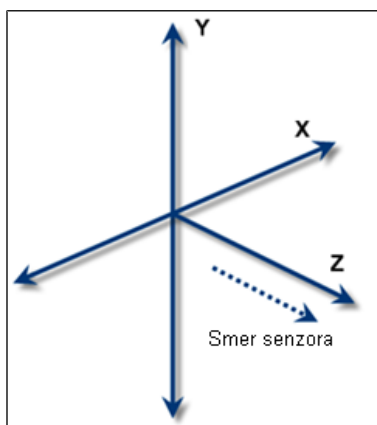
Obrázok 6.3: Silueta s označenými detekovanými rukami

Testovaním implementovaného algoritmu bola overená jeho správna funkčnosť, avšak len v špecifických podmienkach. Ruky je možné detekovať len, ak sú v dostatočnej vzdialenosti pred telom a správnu funkčnosť môže obmedziť prekryvanie rúk v zábere. Ďalším nedostatkom je, že algoritmus nedokáže určiť relatívne pozície rúk voči telu, čo je kritickým problémom z hľadiska implementácie viacerých aspektov návrhu (aktívna zóna, orientácia gesta voči telu). Kvôli týmto nedostatkom nie je program pri implementácii rozhrania využitý a pre detekciu bol použitý systém využívajúci skeletonizáciu, ktorý je opísaný v nasledujúcej kapitole.

## 6.2 Detekcia rúk s využitím skeletonizácie

Systém detekcie rúk, ktorý využíva skeletonizáciu, je implementovaný prostredníctvom knižnice *Kinect for Windows SDK - NUI (Natural User Interface)* (podkapitola 2.2). Konfigurovanie zariadenia *Kinect*, tak ako aj získavanie potrebných dát zo zariadenia, sprostredkúva trieda *KinGet*. Trieda umožňuje zistenie informácií o pripojení a stave pripravenosti zariadenia. Ďalej dovoľuje nastavenie náklonu zariadenia pomocou zabudovaného motorčeka.

Automaticky po pripojení zariadenia *Kinect*, začne objekt triedy *KinGet* poskytovať výstup z jeho RGB kamery a na každého rozpoznaného človeka sa začne aplikovať systém skeletonizácie v režime *Seated*. Avšak systém skeletonizácie poskytuje súradnice rúk v súradnicovej sústave, ktorá má počiatok a orientáciu určenú polohou senzoru. Pre lepšiu predstavu viď obrázok 6.4. Keďže navrhnutý systém rozpoznávania gest vyžaduje znalosť relatívnych pozícií rúk voči telu, sú takéto súradnice krajne nevhodné a to ako pre nevhodný počiatok súradnicovej sústavy, tak aj pre nevhodnú orientáciu súradnicovej sústavy.



Obrázok 6.4: Súradnicový systém výstupov skeletonizácie [24]

S týmto problémom sa aplikácia vysporiada transformáciou súradníc rúk transformačnou maticou, zloženou z translačnej a projekčnej matice. Parametre transformácie sú určené polohou ostatných dostupných rysov tela. Podrobný postup a princíp transformácie je vysvetlený algoritmom 6.2.

### Algoritmus 6.2 - Transformácia súradníc rúk

**Vstup:**

- Absolútne pozície: ľavá ruka, pravá ruka, ľavé rameno, pravé rameno, stred medzi ramenami, hlava.

**Výstup:**

- Relatívne pozície: ľavá ruka, pravá ruka.

**Metóda:**

1. Vytvorí sa translačná matica, ktorá posunie začiatok súradnicovej sústavy do stredu medzi ramenami:

$$Translat = \begin{pmatrix} 1 & 0 & 0 & -stred_x \\ 0 & 1 & 0 & -stred_y \\ 0 & 0 & 1 & -stred_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (27)$$

2. X-ová báza novej súradnicovej sústavy sa určí ako normalizovaný (jednotkový) vektor ramien:

$$báza X = norm(\text{ľavé rameno} - \text{pravé rameno}) \quad (28)$$

3. Vypočíta sa vektor krku (hlavy):

$$\text{vektor krku} = \text{hlava} - \text{stred medzi ramenami} \quad (29)$$

4. Z-ová báza novej súradnicovej sústavy sa určí ako normalizovaný vektorový súčin x-ovej bázy a vektora krku:

$$báza Z = norm(báza X \times \text{vektor krku}) \quad (30)$$

5. Y-ová bázu novej súradnicovej sústavy sa určí ako vektorový súčin x-ovej a z-ovej bázy. Vektorový súčin zaručuje kolmosť výsledného vektora voči oboj vektorom súčinu. Normalizácia nie je nutná, keďže je zrejmé, že vektorový súčin dvoch jednotkových vektorov je taktiež jednotkový vektor.

$$báza Y = báza X \times báza Z \quad (31)$$

6. Z vytvorených báz novej súradnicovej sústavy sa zostrojí projekčná matica nasledovne:

$$Proj = \begin{pmatrix} \text{báza } X_x & \text{báza } X_y & \text{báza } X_z & 0 \\ \text{báza } Y_x & \text{báza } Y_y & \text{báza } Y_z & 0 \\ \text{báza } Z_x & \text{báza } Z_y & \text{báza } Z_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (32)$$

7. Vynásobením translačnej a projekčnej matice vznikne finálna transformačná matica.

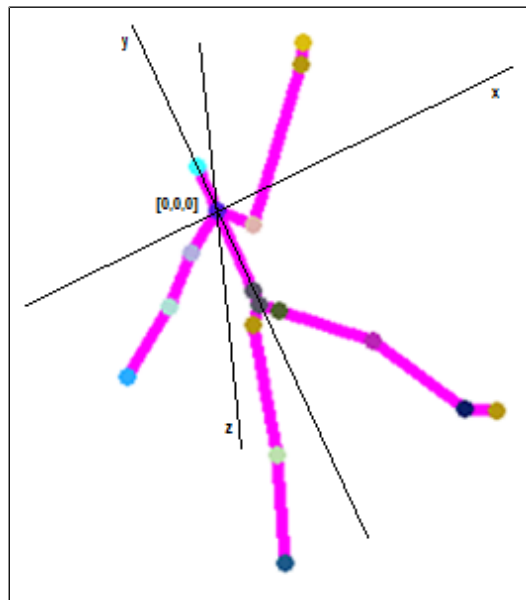
Transformácie nie sú komutatívne, preto je nutné dodržať poradie translácia-projekcia, čo pri násobení matíc znamená opačné poradie:

$$Transform = Proj \cdot Translat \quad (33)$$

8. Výsledné súradnice rúk sa vypočítajú vynásobením transformačnej matice s originálnymi súradnicami.

$$\begin{aligned} \text{relatívna ľavá ruka} &= Transform \cdot \text{ľavá ruka} \\ \text{relatívna pravá ruka} &= Transform \cdot \text{pravá ruka} \end{aligned} \quad (34)$$

Vo výsledku vzniká pre každého detekovaného človeka nový lokálny systém súradníc, ktorý je znázornený na obrázku 6.5. Systém má počiatok v strede medzi ramenami a osi orientované prirodzene podľa tela.



Obrázok 6.5: Lokálny súradnicový systém

Súradnice rúk v takomto súradnicovom systéme sa predávajú ďalej do systému pre spracovanie gest.

## 6.3 Spracovanie gest

Spracovanie, registráciu a detekciu gest implementuje trieda *GestureRecognizer*. Ak sa objekt tejto triedy nachádza v stave registrácie alebo rozpoznávania gesta, tak sa polohy sledovanej ruky v lokálnej súradnicovej sústave ukladajú ako sekvencia priestorových bodov. Vo chvíli, keď je sekvencia kompletná, začne sa jej spracovanie. Prvým krokom spracovania je výpočet novej sekvencie priestorových bodov, ktorá má pevne určenú dĺžku. Výpočet je realizovaný pomocou trilineárnej krokovej interpolácie (algoritmus 6.3).

## Algoritmus 6.3 - Trilineárna kroková interpolácia

### Vstup:

- *origPoints* – pôvodná postupnosť priestorových bodov dĺžky *N*,
- *M* – požadovaná dĺžka novej postupnosti.

### Výstup:

- *interPoints* – postupnosť dĺžky *M* interpolovaná zo vstupnej postupnosti.

### Metóda:

#### 1. Pripravná fáza

- *mutualDistances* – vektor vzájomných vzdialeností susedných bodov,
- *firstDistances* – vektor vzdialeností bodov od prvého bodu (po trase predchádzajúcich bodov),
- *length* – dĺžka celej trasy.

sa vypočítajú nasledovne:

##### a) Inicializácia:

$$\begin{aligned} \text{mutualDistances}(0) &= 0 \\ \text{firstDistances}(0) &= 0 \\ \text{length} &= 0 \end{aligned} \quad (35)$$

##### b) Indukcia:

$$\begin{aligned} \text{mutualDistances}(i+1) &= |\text{origPoints}(i) - \text{origPoints}(i+1)| \\ \text{length} &= \text{length} + \text{mutualDistances}(i+1) \\ \text{firstDistances}(i+1) &= \text{length} \\ i &= 0, 1, \dots, N-2 \end{aligned} \quad (36)$$

#### 2. Interpolácia

##### a) Inicializácia

$$\begin{aligned} \text{interPoints}(0) &= \text{origPoints}(0) \\ \text{interPoints}(M-1) &= \text{origPoints}(N-1) \\ \text{actualPoint} &= 1 \end{aligned} \quad (37)$$

##### b) Indukcia

$$\begin{aligned} \text{actualDist} &= i \cdot \frac{\text{length}}{(M-1)} \\ \text{while}(\text{firstDistances}(\text{actualPoint}) < \text{actualDist}) \\ &\quad \text{actualPoint} = \text{actualPoint} + 1 \\ k &= \frac{(\text{actualDist} - \text{firstDistances}(\text{actualPoint} - 1))}{\text{mutDistances}(\text{actualPoint})} \\ \text{interPoints}(i) &= (1 - k) \cdot \text{origPoints}(\text{actualPoint} - 1) + k \cdot \text{origPoints}(\text{actualPoint}) \\ i &= 1, 2, \dots, N-1 \end{aligned} \quad (38)$$

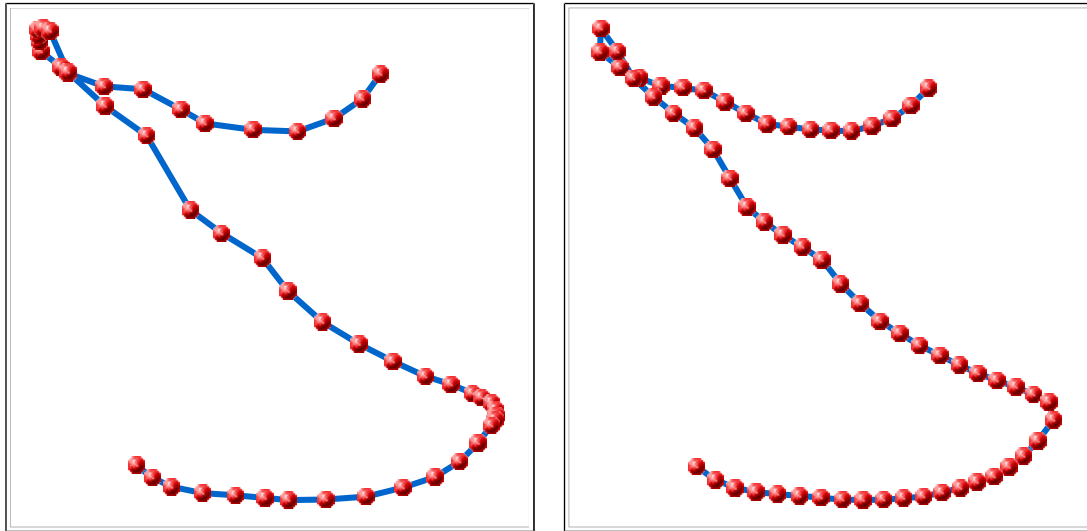
Algoritmus prechádza trajektóriu vytvorenú z pôvodnej postupnosti bodov po krokoch daných podielom dĺžky tejto trajektórie a žiadanej dĺžky novej sekvencie (preto „kroková“ interpolácia). V každom kroku sa všetky tri dimenzie nového bodu vypočítajú z najbližších dvoch bodov (po trajektórii) lineárne váženým priemerom podľa vzdialeností týchto bodov od nového bodu (preto „trilineárna“ interpolácia).

Výsledkom tohoto spracovania je fakt, že body trajektórie gesta sú po trajektórii rovnomerne rozložené a ich počet je pre všetky gesta rovnaký, čo má za následok dva fakty:

- Gesto už viac nie je závislé od rýchlosti jeho vykonania.

- Gesto je taktiež invariantné k zmenám rýchlosti počas jeho vykonávania a je dôležitá len jeho trajektória.

Na obrázkoch 6.6 a 6.7 je trilineárna kroková interpolácia ilustrovaná na príklade gesta v tvare obráteného „Z“. Na obrázku vľavo je pôvodná sekvencia bodov priestorového gesta premietnutá do roviny. Na obrázku vpravo je sekvencia bodov vzniknutá interpoláciou pôvodnej sekvencie, ktorá je taktiež premietnutá do roviny.



Obrázky 6.6 a 6.7: Ukážka trilineárnej krokovej interpolácie

Druhým krokom spracovania je normalizácia už interpolovanej sekvencie bodov gesta, realizovaná algoritmom 6.4. Jedná sa o úpravu, ktorá presunie začiatok súradnicového systému, v ktorom sa gesto nachádza, do stredu gesta a v tomto novom súradnicovom systéme sa gesto preškáluje tak, aby bod gesta s najväčšou vzdialenosťou od stredu gesta mal vzdialenosť 1.

#### **Algoritmus 6.4 - Normalizácia zhluku bodov**

##### **Vstup:**

- *origPoints* – pôvodná postupnosť priestorových bodov dĺžky  $N$ .

##### **Výstup:**

- *normPoints* – normalizovaná pôvodná postupnosť – stred zhluku bodov je na pozícii  $[0,0,0]$  a vzdialenosť najvzdialenejšieho bodu od stredu je 1.

##### **Metóda:**

1. Vypočíta sa *center* – aritmetický stred zhluku bodov:

$$center = \frac{\sum_{i=0}^{N-1} origPoints(i)}{N} \quad (39)$$

2. Presunie sa počiatok súradnicového systému a vypočíta sa *maxDistance* – vzdialenosť najvzdialenejšieho bodu:

- a) Inicializácia

$$maxDistance = 0 \quad (40)$$

- b) Indukcia



$$\begin{aligned} normPoints(i) &= origPoints(i) - center \\ maxDistance &= \max(maxDistance, |normPoints(i)|) \\ i &= 0, 1, \dots, N-1 \end{aligned} \quad (41)$$

3. *Preškálovanie:*

$$\begin{aligned} normPoints(i) &= \frac{normPoints(i)}{maxDistance} \\ i &= 0, 1, \dots, N-1 \end{aligned} \quad (42)$$

Algoritmus vypočíta súradnice stredu gesta ako aritmetický priemer súradníc všetkých bodov a následne od všetkých bodov tieto súradnice stredu odčíta. Nakoniec sa súradnice ešte podelia vzdialenosťou najvzdialenejšieho bodu. Následkom tohto spracovania sú opäť dva fakty:

- Gesto je nezávislé od pozície, v ktorej bolo vykonané.
- Gesto je nezávislé od veľkosti jeho vykonania.

Takto spracované gestá sú pripravené pre systém registrácie/detekcie gest.

## 6.4 Registrácia gest

Definovanie vlastných gest, ktoré rozhranie poskytuje, pracuje na princípe priameho zaznamenania pohybu ruky a je spúšťané výhradne z konvenčného rozhrania, ktoré od užívateľa vyžaduje zvoliť meno gesta, ruku, ku ktorej sa gesto bude viazať a počet kôl registrácie. Aby bolo zaznamenávanie gesta jednoznačné a nebolo ničím rušené, rozhranie vyžaduje, aby sa v priestore detekcie zariadenia Kinect vyskytoval len jeden užívateľ. Registrácia gesta spočíva vo vytvorení troch položiek, pomocou ktorých je neskôr možné porovnávať podobnosť ľubovoľného gest s registrovaným gestom. Týmito položkami sú stredy klastrov gesta, vytrénovaný skrytý Markovský model a hraničná hodnota rozpoznávania. Určeniu optimálnych hodnôt parametrov týkajúcich sa detekcie gest sa venuje podkapitola 7.1.

Rozhranie umožňuje viac kôl registrácie, čo znamená opakované zaznamenávanie práve registrovaného gesta, pričom každé sa bude podieľať rovnakou mierou na parametroch modelu gesta. Priebeh registrácie riadi objekt triedy *GestureRecognizer* a model gesta implementuje trieda *Gesture*, ktorá okrem spomenutých položiek obsahuje aj názov gesta a informáciu, či sa jedná o gesto ľavej alebo pravej ruky. Presný postup registrácie je popísaný algoritmom 6.5, v ktorom sa vyskytuje odkaz na algoritmus k-means [28], ktorý je implementovaný v knižnici *OpenCV* a znázornený na obrázku 6.8.

### ***Algoritmus 6.5 - Registrácia gesta***

***Vstup:***

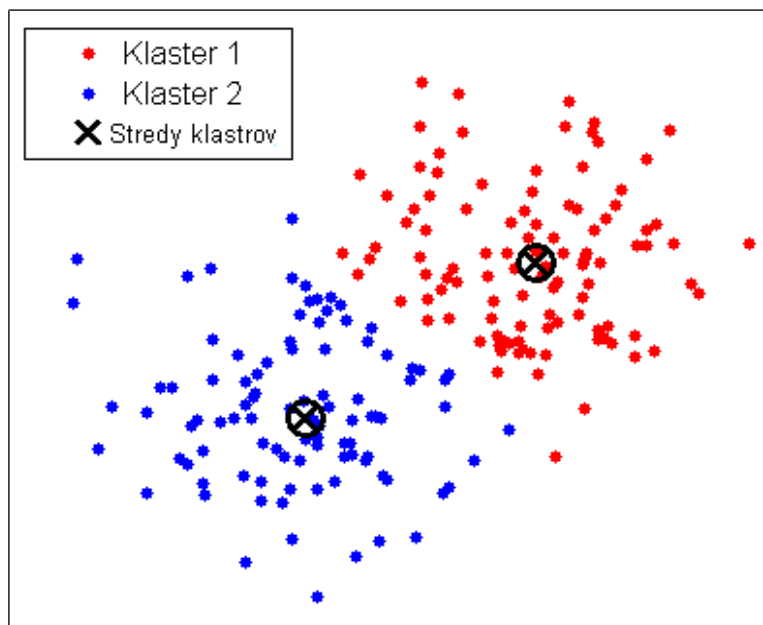
- množina gest – interpolovaných a normalizovaných sekvencií polôh ruky.

***Výstup:***

- stredy klastrov gesta,
- hraničná hodnota rozpoznávania
- skrytý Markovský model, ktorý je v spojení so stredami klastrov schopný určiť podobnosť vstupného gesta s iným gestom.

***Metóda:***

1. Pomocou algoritmu *k-means* sú polohy ruky všetkých vstupných gest rozdelené do  $N$  očíslovaných klastrov (zhlukov) tak, že vzdialenosť bodu od stredu klastra, do ktorého bod patrí je menšia ako vzdialenosť od ostatných stredov klastrov a  $N$  je plánovaný počet výstupných symbolov skrytého Markovského modelu.
2. Uložia sa stredy klastrov.
3. Pre každé vstupné gesto sa vytvorí postupnosť príslušností všetkých bodov gesta do jednotlivých klastrov.
4. Vytvorí sa skrytý Markovský model typu LRB.
5. Skrytý Markovský model sa trénuje množinou postupností príslušností do klastrov pomocou algoritmu Viterbi Path Counting (algoritmus 3.3).
6. Uloží sa vytrénovaný skrytý Markovský model (pravdepodobnosti počiatočného stavu, pravdepodobnosti prechodov, výstupné pravdepodobnosti).
7. Pomocou algoritmu Forward-Backward (algoritmus 3.1) sa na skrytom Markovskom modeli postupne otestujú všetky postupnosti príslušností bodov do klastrov a priemerná hodnota výsledných pravdepodobností určí hraničnú hodnotu rozpoznávania gesta.



Obrázok 6.8: Výsledok klasterizácie pomocou algoritmu *k-means* [29]

## 6.5 Detekcia gest

Ak nie je objekt triedy *GestureRecognizer* v stave registrácie gest, automaticky prebieha detekcia gest, pre obe ruky všetkých detekovaných užívateľov. Detekcia gest spočíva v počítaní podobnosti gesta so všetkými registrovanými gestami pre danú ruku. Ak je podobnosť s niektorým registrovaným gestom vyššia ako minimálna požadovaná podobnosť pre detekciu, rozhranie potvrdí úspešnú detekciu gesta. Ak je rozpoznané viac ako jedno gesto, vyberie sa gesto s najvyššou podobnosťou. Celý postup detekcie detailne popisuje algoritmus 6.6.

### *Algoritmus 6.6 - Detekcia gesta*

**Vstup:**

- gesto – interpolovaná a normalizovaná sekvencia polôh ruky,
- množina registrovaných gest pre danú ruku,
- minimálna požadovaná podobnosť.

**Výstup:**

- názov (ID) detekovaného gesta, alebo oznámenie žiadnej detekcie.

**Metóda:**

1. Vypočíta sa podobnosť so všetkými registrovanými gestami:

- a) Vypočíta sa postupnosť príslušností do klastrov registrovaného gesta tak, že pre každý bod je zvolený klaster, ktorého stred je k bodu najbližší.
- b) Pomocou algoritmu Forward-Backward (algoritmus 3.1) sa na skrytom Markovskom modeli registrovaného gesta vypočíta pravdepodobnosť postupnosti príslušností do klastrov.
- c) Podobnosť gesta s registrovaným gestom je určená podielom vypočítanej pravdepodobnosti ( $p$ ) a hraničnej hodnoty rozpoznávania registrovaného gesta ( $t$ ), vypočítanej pri registrácii.

$$\text{podobnosť} = \frac{p}{t} \quad (43)$$

2. Vyberie sa gesto s najvyššou podobnosťou.
3. Ak podobnosť tohto gesta je vyššia alebo rovná minimálnej požadovanej podobnosti, oznámi sa detekcia gesta. Inak sa oznámi žiadna detekcia.

## 6.6 Ovládanie gestami

Ovládanie jednotlivých multimedialných modulov aplikácie je umožnené prepojením objektu *MainWindow*, ktorý tieto moduly obsahuje, s objektom detekcie gest – *GestureRecognizer*. Toto prepojenie funguje tak, že pre každé detekované gesto odošle objekt triedy *GestureRecognizer* objektu *MainWindow* dvojicu (ID užívateľa, ID gesta), kde ID užívateľa je pre každého užívateľa scény unikátne číslo, ktoré má užívateľ automaticky pridelené pri jeho prvej detekcii, až pokiaľ neopustí snímanú scénu a ID gesta je poradové číslo gesta v zozname gest.

Konfiguračná časť konvenčného rozhrania aplikácie umožňuje upraviť nasledovné parametre ovládania gestami:

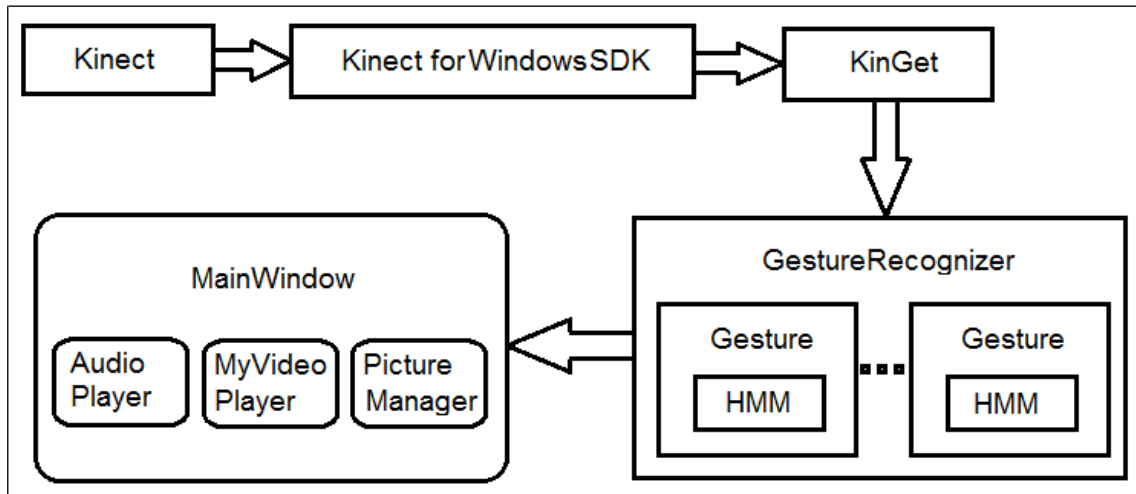
- zapnutie/vypnutie ovládania gestami,
- voľba prihlasovacieho a odhlasovacieho gesta,
- priradenie gesta k jednotlivým akciám multimedialných modulov aplikácie.

Ak je ovládanie gestami povolené, tak sa každá prijatá dvojica (ID užívateľa, ID gesta) spracuje podľa nasledujúcich zásad:

- Udržiava sa zoznam ID prihlásených užívateľov, do ktorého je užívateľ pridaný po úspešnej detekcii gesta, ktoré je nastavené ako prihlasovacie a logicky je z neho odobraný po detekcii odhlasovacieho gesta.
- Detekcia iného ako prihlasovacieho gesta neprihláseného užívateľa nevedie k žiadnej akcii.
- Po detekcii gesta prihláseného užívateľa sa v aktuálne aktívnom multimedialnom module vykonajú všetky akcie, pre ktoré je práve rozpoznané gesto nastavené ako spúšťačie.

## 6.7 Tok dát

Pre ucelenejší obraz o toku dát a použitých triedach na ceste od zariadenia *Kinect* až po ovládanie multimediálnych aplikácií gestami, je na obrázku 6.9 znázornená schéma toku dát a pod obrázkom sa nachádza popis schémy.



Obrázok 6.9: Schéma toku dát pri rozpoznávaní gest

1. Objekt *KinGet* získava pomocou knižnice *Kinect for Windows SDK* zo senzoru *Kinect* dáta o snímaných užívateľoch. Tieto dáta spracováva a posíla objektu *GestureRecognizer* pozície rúk užívateľov.
2. Objekt *GestureRecognizer* buď z prijatých dát vytvára množinu registrovaných gest - objektov triedy *Gesture*, alebo dáta využíva k detekcii vytvorených gest.
3. Výsledky detekcie gest posíla objekt *GestureRecognizer* objektu *MainWindow*, ktorý obsahuje multimediálne moduly: *AudioPlayer*, *MyVideoPlayer* a *PictureManager*, v ktorých podľa prijatých informácií o detekcii gest vykonáva akcie.

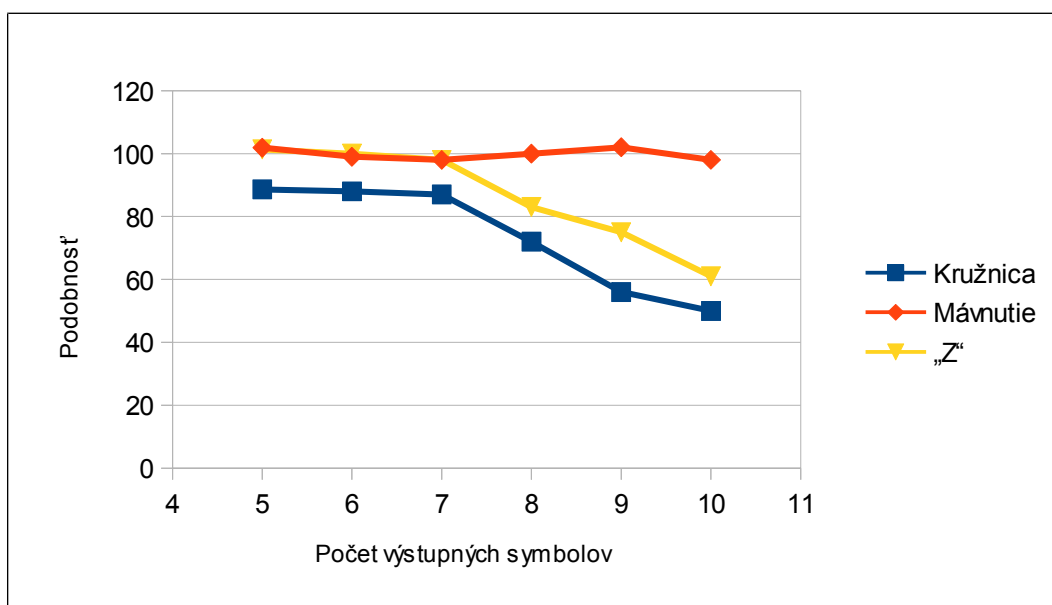
# 7 Testovanie rozhrania pre gestá rúk

## 7.1 Parametre detekcie gest

Kvalitná detekcia gest vyžaduje optimálne nastavenie parametrov ich registrácie a detekcie, ktoré bolo nutné získať komplexným testovaním, ktorému sa venuje táto kapitola.

### Počet výstupných symbolov HMM

Prvým testovaným parametrom je počet výstupných symbolov používaných skrytých Markovských modelov. Keďže počet výstupných symbolov určuje počet klastrov, do ktorých sú polohy gesta rozdelené, ovplyvňuje tento závislosť systému rozpoznávania od komplexnosti gesta. Testovanie prebehlo na troch rôznych gestách: jednoduché mávnutie rukou, pohyb ruky po kružnici a pohyb ruky po trajektórii v tvare písmena „Z“. Registrované a rozpoznávané gestá boli nahrávané zvlášť (tak ako tomu je v reálnom používaní aplikácie), čiže tieto gestá sa mierne líšili. Pri meniacom sa počte výstupných symbolov a nemenných ostatných parametroch detekcie sa pre všetky tri spomenuté gestá testovala odozva systému detekcie gest pre veľmi podobné gestá s rôznou úrovňou komplexnosti. Výsledky testov vidíme na obrázku 7.1.

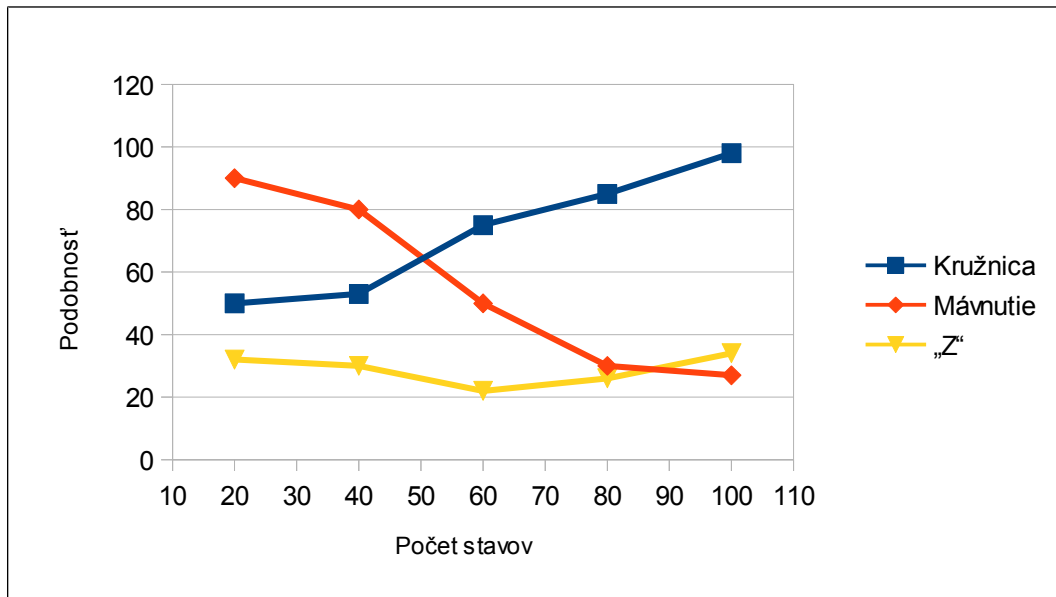


Obrázok 7.1: Testovanie počtu výstupných symbolov

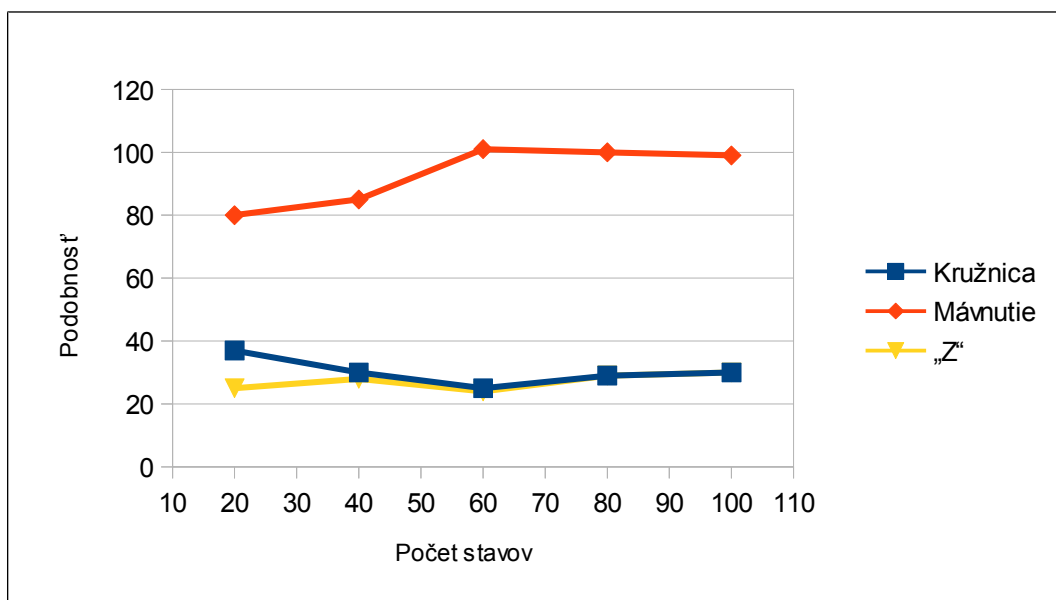
Z grafu vyplýva, že pre jednoduché gesto, akým je mávnutie rukou, je schopnosť rozpoznat' podobné gesto vysoká aj so zvyšujúcim sa sledovaním zložitosti gesta. Naopak, pri vykonávaní zložitejších gest nastávajú odchýlky, ktoré sú pri použití počtu výstupných symbolov vyššom ako 7, zahrnuté do komplexnosti gesta, čo vedie k nízkej detekcii podobnosti gesta.

## Počet stavov HMM a dĺžka interpolovaného gesta

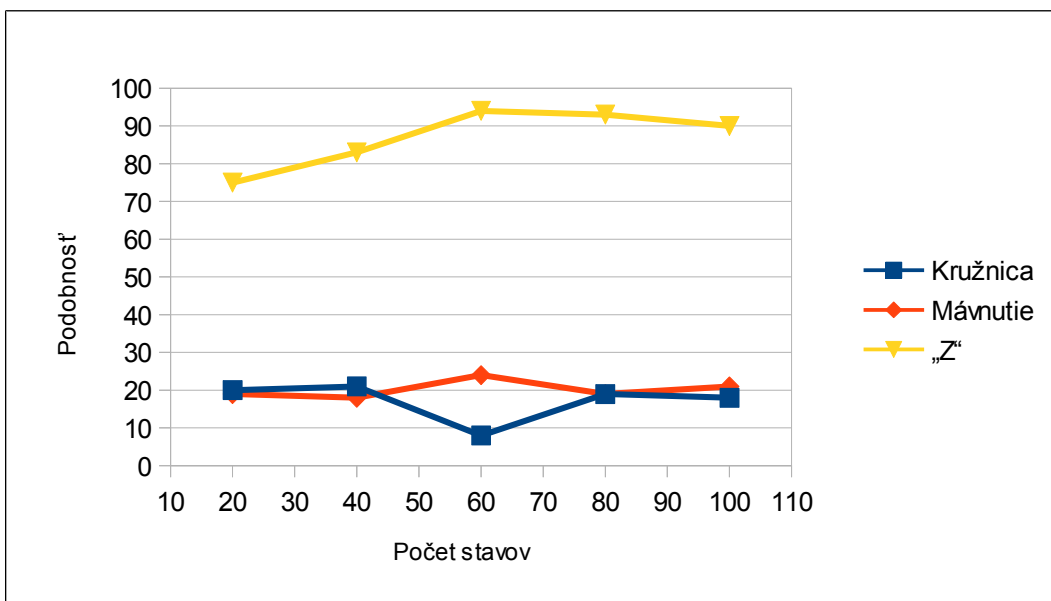
System rozpoznávania gest využíva skryté Markovské modely typu LRB a z ich princípu vyplýva, že dĺžka modelovaného javu, teda v tomto prípade interpolovaného gesta, by nemala byť väčšia ako počet stavov HMM. Preto testovanie oboch týchto parametrov prebehlo súčasne a so zhodnými hodnotami. Testy zisťovali závislosť určenia podobnosti zhodných i rôznych gest pri meniacom sa počte stavov HMM a teda i dĺžke interpolovaného gesta. Využité boli opäť gestá: jednoduché mávnutie rukou, pohyb ruky po kružnici a po trajektórii v tvare písmena „Z“, pričom na obrázku 7.2 sú zobrazené vypočítané podobnosti s gestom typu „kružnica“, na obrázku 7.3 podobnosti s gestom typu „mávnutie“ a na obrázku 7.4 podobnosti s gestom typu „Z“.



Obrázok 7.2: Testovanie počtu stavov (1)



Obrázok 7.3: Testovanie počtu stavov (2)

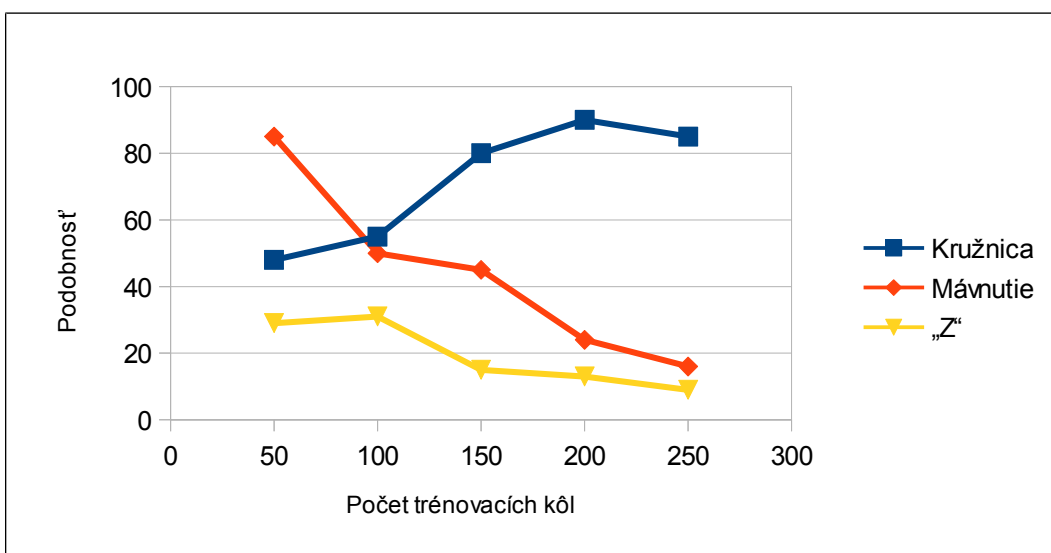


Obrázok 7.4: Testovanie počtu stavov (3)

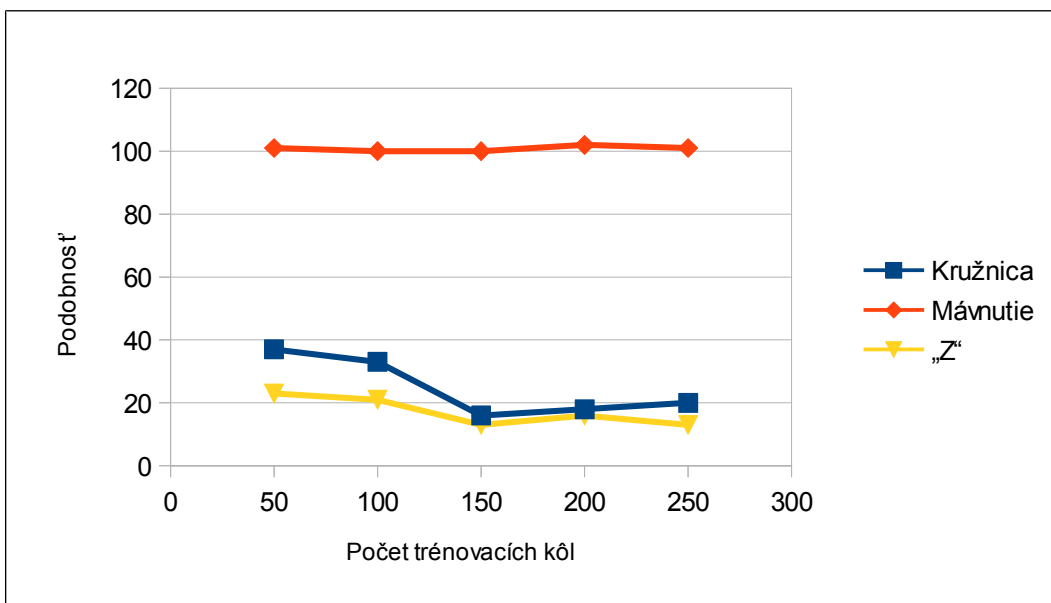
Z grafov vyplýva, že stabilita rozpoznávania od počtu stavov a dĺžky interpolovaného gesta silne závisí. So zväčšujúcim sa počtom stavov sa zvyšuje zistená podobnosť gesta s takmer zhodným gestom a znižuje podobnosť gesta s rôznym gestom. Minimálnym počtom stavov pre stabilnú detekciu je približne 70 stavov.

## Počet tréningových kôl

Tréning skrytých Markovských modelov realizuje algoritmus VPC, ktorého parametrom je počet tréningových kôl, v ktorých aktualizuje hodnoty modelu. Na už z predchádzajúcich testov známych gestách sa testoval vplyv zmeny tohto parametra na kvalitu rozpoznávania. Výsledky testov sú na obrázkoch 7.5, 7.6 a 7.7.



Obrázok 7.5: Testovanie počtu tréningových kôl (1)



Obrázok 7.6: Testovanie počtu trérovacích kôl (2)



Obrázok 7.7: Testovanie počtu trérovacích kôl (3)

Podobne ako v predchádzajúcom teste sú na prvom obrázku zobrazené vypočítané podobnosti s gestom typu „kružnica“, na druhom obrázku podobnosti s gestom typu „mávnutie“ a na treťom obrázku podobnosti s gestom typu „Z“. Z testov vyplýva, že zvyšovanie počtu kôl trérovania zvyšuje kvalitu rozpoznávania, avšak len po určitú hranicu. Približne po hodnotu 200 trérovacích kôl je na grafoch zjavná tendencia zvyšovania sa zistenej podobnosti so správnym gestom a zároveň znižovania sa podobnosti s nesúhlasiacimi gestami. Pri ďalšom zvyšovaní počtu kôl sa ale tento trend mení a podobnosť s takmer identickým gestom sa znižuje, pretože model je „pretrénovaný“ na registrovanom geste a drobné odchýlky v rozpoznávanom geste sa na vypočítanej podobnosti prejavujú vo veľkej miere.

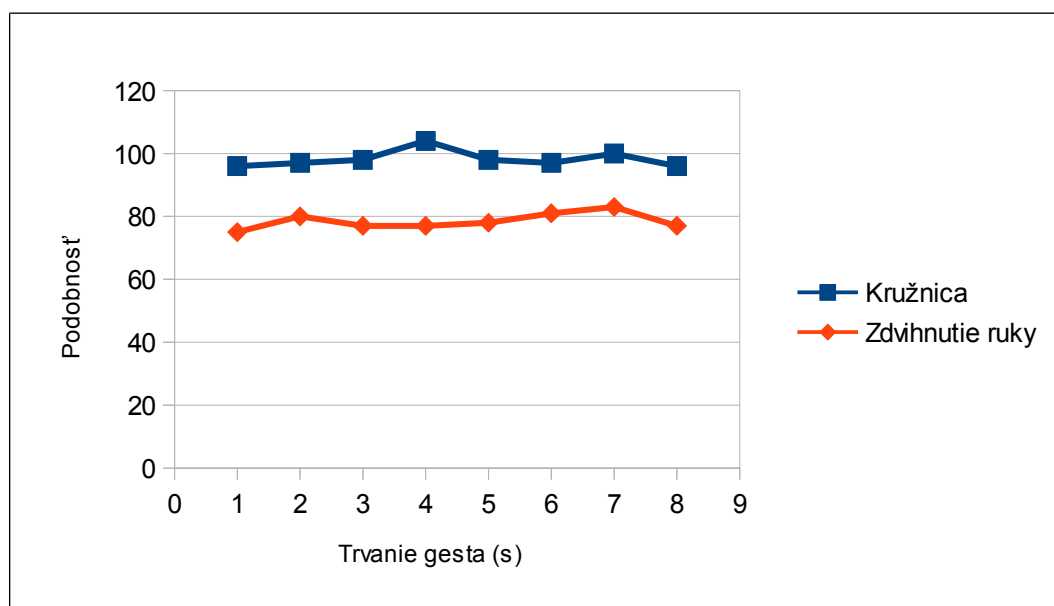


## 7.2 Splnenie návrhu rozhrania

Nasledujúce testy overujú úspešnosť implementácie navrhnutého rozhrania v otázke nezávislosti systému detekcie gest na rýchlosti vykonania gesta, veľkosti gesta a polohe gesta.

### Invariancia k rýchlosti vykonania gesta

Invariancia rozhrania k rýchlosti vykonania gesta sa overovala na geste v tvare kružnice a geste zdvihnutia ruky nad hlavu. Test prebiehal zisťovaním podobnosti zaregistrovaných gest s gestami po rovnakej trajektórii, avšak vykonávaných rôznymi rýchlosťami. Výsledok testu ilustruje obrázok 7.8.

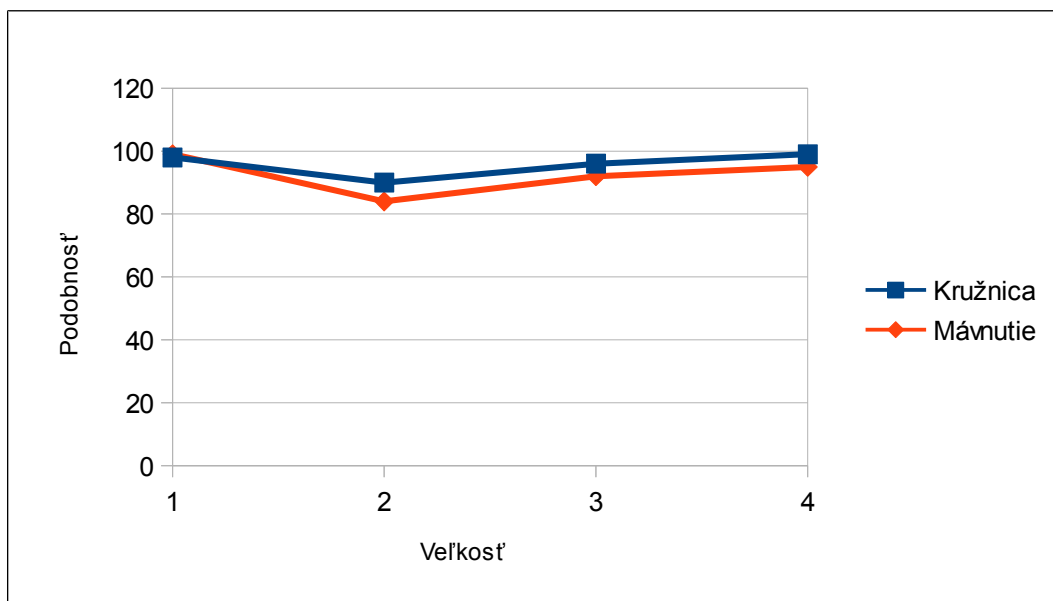


Obrázok 7.8: Testovanie invariance k rýchlosti vykonania gesta

Ako možno vidieť na výsledkovom grafe, test potvrdil úspešnosť implementácie rozhrania v otázke invariance rozpoznávania gest k rýchlosti ich vykonania, pretože vypočítaná podobnosť gest je nezávislá na dobe vykonávania gesta.

### Invariancia k veľkosti vykonania gesta

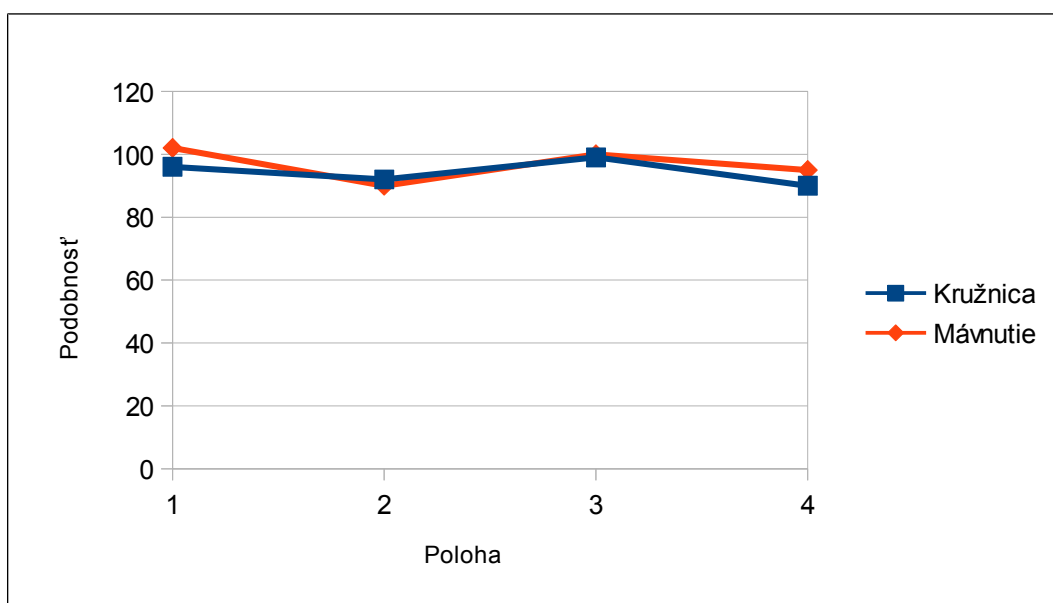
V tomto teste boli najskôr zaregistrované gestá „kružnica“ a „mávnutie“, aby mohli byť následne vykonávané v štyroch rôznych veľkostiach. Ako môžeme vidieť v grafe na obrázku 7.9, systém detekcie gest s vysokou a stálou pravdepodobnosťou zistil podobnosť registrovaných a rozpoznávaných gest, aj keď boli vykonávané v rôznom merítku. Test teda potvrdil úspešnosť implementácie i v otázke invariance detekcie gest k veľkosti ich vykonania.



Obrázok 7.9: Testovanie invariance k veľkosti vykonania gesta

## Invariancia k polohe gesta

Overovanie invariance rozhrania k polohe, v ktorej je gesto vykonané, prebiehalo veľmi podobne ako predchádzajúci test. Jediným rozdielom bola skutočnosť, že pri rozpoznávaní gest boli gestá vykonávané v rovnakej veľkosti v štyroch rôznych polohách vzhľadom k telu.



Obrázok 7.10: Testovanie invariance k polohe gesta

Z grafu na obrázku 7.10, ktorý ilustruje výsledky testu, vyplýva, že úspešnosť rozpoznávania je stabilná a vysoká pre obidve gestá vykonané v rôznych polohách. Implementované rozhranie teda súhlasí s návrhom i v tomto aspekte.

## 7.3 Užívateľské testovanie

Po vyladení rozhrania pre gestá rúk pomocou optimálnych parametrov detekcie gest získaných pri vývojovom testovaní, bolo uskutočnených množstvo užívateľských testov tohto rozhrania zameraných na jeho reálnu využiteľnosť. Testovanie jednoznačne potvrdilo, že práca s multimediálnym obsahom pomocou navrhnutého rozhrania je možná a má svoje výhody i nevýhody.

Dokázalo sa, že konvenčné rozhranie umožňuje pracovať s multimediálnym obsahom nepochybne rýchlejšie ako implementované alternatívne rozhranie. Na druhej strane, využívanie bezdotykového rozhrania môže byť pohodlnejšie, pretože umožňuje ovládanie aplikácie z ľubovoľného miesta, ktoré je v dosahu senzora a to pre viacerých užívateľov súčasne. V rozpore s predpokladmi vychádzajúcimi z prvotných užívateľských štúdií sa ukázalo, že integrácia online gest by v rozhraní našla uplatnenie a to pri plynulej úprave hlasitosti, priamom nastavovaní pozície prehrávania, plynulom nastavovaní priblíženia obrázka a rýchlym prechádzaní súborového systému. Keďže rozhranie podporuje výhradne offline gestá, je realizácia vymenovaných operácií prenechaná konvenčnému rozhraniu, prípadne riešená pomocou kompromisov, akým je zmena hlasitosti a priblíženia po pevne stanovených krokoch, čo nezanedbateľne znižuje celkovú použiteľnosť rozhrania.

Stabilita detekcie vhodne zvolených gest je dostatočná a registrácia gest je jednoduchá. Avšak problém nastáva po registrácii viacerých podobných gest, kedy je rozpoznanie užívateľom zamýšľaného gesta problematické. Na druhej strane, škála možných gest je pre obe ruky dostatočná pre všetky akcie, ktoré je gestami možné ovládať. V najväčšej miere sa pri ovládaní multimediálnej aplikácie osvedčili krátke a jednoduché gestá pre rýchlosť ich prevedenia. Ďalším zisteným potencionálnym problémom je možnosť, že užívateľ počas viackolovej registrácie dodá registračnému systému v jednotlivých kolách navzájom príliš odlišné gestá a vytvorený model gesta má potom nepredvídateľné vlastnosti. Bolo by vhodné, aby rozhranie dokázalo takúto neúspešnú registráciu rozpoznať.

Veľmi dôležité je vhodné nastavenie minimálnej podobnosti gesta pre jeho detekciu, pretože pri jeho vysokých hodnotách je užívateľ nútený vykonávať gestá s vysokou presnosťou a naopak pri príliš nízkych hodnotách a viacerých zaregistrovaných gestách vedie prakticky každý pohyb ruky v aktívnej zóne k pozitívnej detekcii gesta. Systém prihlasovania sa do systému pomocou vhodne zvoleného (dostatočne komplexného) gesta však ponúka spôsob, ako zabrániť vykonávaniu nechcených akcií.

Nedostatkom rozhrania je podľa niektorých užívateľov skutočnosť, že orientácia gesta je závislá výhradne od polohy tela užívateľa a neberie sa v úvahu vodorovná rovina, v ktorej majú užívatelia tendenciu gestá vykonávať.

Testovanie ďalej ukázalo, že viacero používateľov intuitívne preferuje rovnaké alebo podobné gestá pre určité akcie, ako napríklad mávnutie rukou doľava pre prechod na ďalší obrázok, alebo pohyb ruky po kružnici pre rotáciu obrázka. Z tohto zistenia vyplýva skutočnosť, že preddefinované gestá by v rozhraniach prepojených s multimediálnym obsahom našli svoje uplatnenie.

Nároky aplikácie na výpočtový výkon nie sú vysoké, všetkým testujúcim užívateľom pracovala registrácia gest s minimálnym oneskorením a rozpoznávanie gest takmer v reálnom čase. Aj preto užívatelia aplikáciu hodnotili kladne a uvítali by podobné rozhranie na svojich počítačoch, televíznych prijímačoch a domácich kinách.

## 8 Záver

Prvým výsledkom tejto diplomovej práce je súhrn poznatkov o senzore Kinect. Súhrn umožňuje čitateľovi oboznámenie sa s technickými detailami zariadenia, s princípmi v ňom využitými a s možnosťami jeho využitia pri vývoji aplikácií prostredníctvom dostupných knižníc. Ďalej práca sumarizuje poznatky o detekcii gest so špeciálnou pozornosťou venovanou detekcii gest rukami založenej na počítačovom videní. Ďalším produktom práce teoretického charakteru je užívateľský prieskum zameraný na požadované vlastnosti rozhrania pre gestá rúk určeného pre prácu s multimediálnym obsahom.

Praktickým produktom práce je multimediálna aplikácia, ktorá obsahuje rozhranie pre gestá rúk a bola navrhnutá s ohľadom na získané teoretické poznatky. Aplikácia umožňuje prehrávanie zvukových i vizuálnych súborov a prezeranie obrázkov. Rozhranie pre gestá rúk dokáže pomocou zariadenia Kinect zaznamenávať vlastné gestá rúk a neskôr ich detekciu využiť pre interakciu s multimediálnym obsahom. Funkčnosť tohto rozhrania aplikácie je predvedená na krátkych video ukážkach, ktoré sú taktiež súčasťou práce.

Vývojové a užívateľské testy implementovanej aplikácie priniesli množstvo cenných poznatkov pre problematiku interakcie s multimediálnym obsahom pomocou gest rúk a odhalili jej silné i slabé stránky. Budúce projekty podobného zamerania môžu z výsledkov testov čerpať a pomocou nich sa vyhnúť zbytočným chybám už vo fáze návrhu projektu.

Implementované rozhranie pre gestá rúk má veľký potenciál a jeho zistené nedostatky je možné ďalej vylepšovať. Medzi navrhované úpravy patrí začlenenie online gest do rozhrania, vylepšenie posúdenia aktivity rúk, prípadne prechod na kontinuálne rozpoznávanie gest, zapracovanie posúdenia kvality registrovaných gest a pridanie možnosti využitia preddefinovaných gest.

# Literatúra

- [1] Kinect [online]. 20.12.2011 [cit. 1.1.2012].  
Dostupný z WWW: <<http://en.wikipedia.org/wiki/Kinect>>
- [2] Technical description of Kinect calibration [online] 9.12.2011 [cit. 1.1.2012].  
Dostupný z WWW: <[http://www.ros.org/wiki/kinect\\_calibration/technical](http://www.ros.org/wiki/kinect_calibration/technical)>
- [3] Kinect for Windows Features [online]. 2011 [cit. 1.1.2012].  
Dostupný z WWW: <<http://kinectforwindows.org/features/>>
- [4] About the OpenNI organization [online]. 2010 [cit. 1.1.2012].  
Dostupný z WWW: <<http://75.98.78.94/About.aspx>>
- [5] OpenNI Programmer Guide [online]. 2010 [cit. 1.1.2012].  
Dostupný z WWW: <<http://openni.org/Documentation/ProgrammerGuide.html>>
- [6] Nite [online]. 2011 [cit. 1.1.2012]. Dostupný z WWW: <<http://www.primesense.com/Nite/>>
- [7] OpenKinect [online]. 28.4.2011 [cit. 1.1.2012]. Dostupný z WWW: <<http://openkinect.org/>>
- [8] Feng-Sheng Chen, Chih-Ming Fu, Chung-Lin Huang: *Hand gesture recognition using a real-time tracking method and hidden Markov models*, Taiwan, 2003
- [9] Controls for Wii [online]. 2011 [cit. 1.1.2012].  
Dostupný z WWW: <<http://www.nintendo.com/wii/what-is-wii/#/controls>>
- [10] PlayStation Move [online]. 2011 [cit. 1.1.2012]. Dostupný z WWW:  
<<http://us.playstation.com/ps3/playstation-move/>>
- [11] OpenCV [online]. 24.8.2011 [cit. 1.1.2012]. Dostupný z WWW:  
<<http://opencv.willowgarage.com/wiki/>>
- [12] Pragati Garg, Naveen Aggarwal, Sanjeev Sofat: *Vision Based Hand Gesture Recognition*, Punjab, 2009
- [13] Ying Wu, Thomas S. Huang: *Hand modeling analysis and recognition for vision-based human computer interaction*, IEEE Signal Processing Magazine, 2001
- [14] Skeleton tracking [online]. 15.12.2010 [cit. 1.1.2012]. Dostupný z WWW: <<http://luka.tnode.com/gallery/projects-and-work/kinect-hackery/skeleton-tracking>>
- [15] Gesture recognition [online]. 26.12.2011 [cit. 1.1.2012]. Dostupný z WWW: <[http://en.wikipedia.org/wiki/Gesture\\_recognition](http://en.wikipedia.org/wiki/Gesture_recognition)>
- [16] Proxemics [online]. 1996 [cit. 1.1.2012].  
Dostupný z WWW: <<http://www.cs.unm.edu/~sheppard/proxemics.htm>>
- [17] B. Stenger, P. R. S. Mendonca, R. Cipolla: *Model-Based 3D Tracking of an Articulated Hand*, Cambridge, 2001
- [18] Berry, G.: Small-wall: *A Multimodal Human Computer Intelligent Interaction Test Bed with Applications*, Urbana-Champaign, 1998
- [19] Zeller, M., et al: *A Visual Computing Environment for Very Large Scale Biomolecular Modeling*, Zurich, 1997
- [20] Dong Guo Yonghua: *Vision-Based Hand Gesture Recognition for Human-Vehicle Interaction*, International Conference on Control, Automation and Computer Vision, 1998
- [21] Pickering, Carl A. Burnham, Keith J. Richardson, Michael J. Jaguar: *A research Study of Hand Gesture Recognition Technologies and Applications for Human Vehicle Interaction*, 3rd Conference on Automotive Electronics, 2007

- [22] Juan P. Wachs, Helman I. Stern, Yael Edan, Michael Gillam, Jon Handler, Craig Feied, Mark Smith: *A Gesture-based Tool for Sterile Browsing of Radiology Images*, Journal of the American Medical Informatics Association, 2008
- [23] Toshiba Formally Unveils Notebooks with SpursEngine Chip [online]. 24.6.2008 [cit. 1.1.2012]. Dostupný z WWW: <[http://www.xbitlabs.com/news/mobile/display/20080624203611\\_Toshiba\\_Formally\\_Unveils\\_Notebooks\\_with\\_SpursEngine\\_Chip.html](http://www.xbitlabs.com/news/mobile/display/20080624203611_Toshiba_Formally_Unveils_Notebooks_with_SpursEngine_Chip.html)>
- [24] Microsoft Korporation: *Kinect for Windows Programming Guide*, 2012
- [25] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake: *Real-Time Human Pose Recognition in Parts from Single Depth Images*, Microsoft Research Cambridge & Xbox Incubation, 2011
- [26] TA2 - Together Anywhere, Together Anytime [online]. 2011 [cit 5.5.2012]. Dostupný z WWW: <<http://www.ta2-project.eu>>
- [27] Qt SDK [online]. 2012 [cit 5.5.2012]. Dostupný z WWW: <<http://qt.nokia.com/products/qt-sdk>>
- [28] OpenCV documentation - Clustering [online]. 28.4.2012 [cit 5.5.2012]. Dostupný z WWW: <<http://docs.opencv.org/modules/core/doc/clustering.html>>
- [29] MathWorks Product Documentation - k-means [online]. 2012 [cit 5.5.2012]. Dostupný z WWW: <<http://www.mathworks.com/help/toolbox/stats/kmeans.html>>
- [30] Qt Documentation – Phonon [online]. 2011 [cit 5.5.2012]. Dostupný z WWW: <<http://qt-project.org/doc/qt-4.8/phonon.html>>
- [31] Miguel Reyes, Gabriel Domínguez, Sergio Escalera: *Feature Weighting in Dynamic TimeWarping for Gesture Recognition in Depth Data*, IEEE International Conference on Computer Vision Workshops, 2011
- [32] M. Parizeau and R. Plamondon: *A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 12, No. 7, Júl 1990
- [33] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani: *Algorithms*, 13.9.2006
- [34] Jozef Kovář: *Skryté Markovské modely a neuronové siete*, 2008
- [35] Nikolai Shokhirev: *Hidden Markov Models* [online]. 15.2.2010 [cit 5.5.2012]. Dostupný z WWW: <<http://www.shokhirev.com/nikolai/abc/alg/hmm/hmm.html>>
- [36] Paul E. Black : *"finite state machine"*, Dictionary of Algorithms and Data Structures [online]. 12.5.2008 [cit 5.5.2012]. Dostupný z WWW: <<http://xlinux.nist.gov/dads//HTML/finiteStateMachine.html>>
- [37] OpenCV documentation – Miscellaneous Image Transformation [online]. 2012 [cit 1.4.2013]. Dostupný z WWW: <[http://opencv.willowgarage.com/documentation/cpp/miscellaneous\\_image\\_transformations.html](http://opencv.willowgarage.com/documentation/cpp/miscellaneous_image_transformations.html)>
- [38] Nianjun Liu, Brian C. Lovell, Peter J. Kootsookos: *Evaluation of HMM Training Algorithms For Letter Hand Gesture Recognition*, 2003
- [39] R. I. A. Davis, B. C. Lovell: *Comparing and evaluating HMM ensemble training algorithms using train and test condition number criteria*, 2003
- [40] Lawrence R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in speech recognition*, Proceedings of the IEEE, Vol. 77, No. 2, Február 1989

- [41] Nianjun Liu, Brian C. Lovell, Peter J. Kootsookos: *Model Structure Selection & Training Algorithms for HMM Gesture Recognition System*, The University of Queensland, Brisbane, Australia, 2004
- [42] Nianjun Liu, R. I. A. Davis, Brian C. Lovell, Peter J. Kootsookos: *Effect of Initial HMM Choices in Multiple Sequence Training for Gesture Recognition*, The University of Queensland, Brisbane, Australia,

# Zoznam príloh

Príloha A – Obsah CD

Príloha B – CD



## Príloha A – Obsah CD

- **DP/xkubic05.odt** – zdrojový tvar diplomovej práce
- **DP/xkubic05.pdf** – diplomová práce vo formáte PDF
- **Multimedia Gestures/bin** – adresár so skompilovanou aplikáciou Multimedia Gestures
- **Multimedia Gestures/bin/build.txt** – informácie o OS a architektúre, na ktorej bola aplikácia skompilovaná
- **Multimedia Gestures/src** – adresár so zdrojovými súbormi aplikácie Multimedia Gestures
- **Multimedia Gestures/src/instal.txt** – pokyny k inštalácii aplikácie Multimedia Gestures
- **Multimedia Gestures/manual.pdf** – užívateľská príručka aplikácie Multimedia Gestures
- **video/registracia.avi** – videosekvencia prezentujúca registráciu gest v Multimedia Gestures
- **video/rospoznavanie.avi** – videosekvencia prezentujúca rozpoznávanie gest v Multimedia Gestures