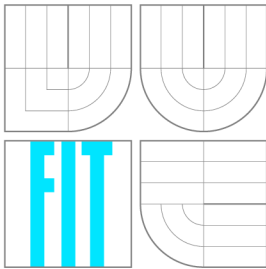


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

OVLÁDÁNÍ MULTIMEDIÁLNÍHO PŘEHRÁVAČE GESTY RUKY

MUTLIMEDIA PLAYER CONTROL USING HAND GESTURES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ ZATLOUKAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2016

Zadání bakalářské práce

Řešitel: **Zatloukal Tomáš**

Obor: Informační technologie

Téma: **Ovládání multimediálního přehrávače gesty ruky**
Multimedia Player Control Using Hand Gestures

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se základními postupy segmentace obrazu, detekce gest v hloubkových datech a postupech při návrhu uživatelského rozhraní.
2. Navrhněte metodu, která v hloubkových datech detekuje gesta ruky, a klasifikuje gesto. Dále navrhněte rozhraní pro ovládání multimediálního přehrávače pomocí těchto gest.
3. Připravte a anotujte datovou sadu pro vývoj, trénování a testování systému.
4. Implementujte navrženou metodu s využitím existujících knihoven a proveďte experimenty a vyhodnocení stability a přesnosti metody v různých podmínkách.
5. S využitím metody realizujte ovládací prvek a demonstруйте jeho možnosti v multimediálním přehrávači.
6. Vytvořte plakát a krátké demonstrační video reprezentující Vaše řešení.

Literatura:

- Bradski G. R., Kaehler A. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc. 2008.
- M. Sonka, V. Hlaváč, R. Boyle: *Image Processing, Analysis, and Machine Vision*, CL-Engineering, ISBN-13: 978-0495082521, 2007.
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2, 3 a částečně bod 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 00 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá možnostmi snímání gest rukou provedené uživatelem. Cílem je vytvoření aplikace umožňující ovládání multimedialního přehrávače gesty ruky detekované v hloubkových datech. V textu jsou uvedeny použité nástroje a metody spolu s implementačními detaily a výsledky testů.

Abstract

This thesis deals with study possibilities of capturing hand gestures made by user. The goal is to create application that allows control multimedia player by hand gesture detected in depth data. The text discuss used tools and methods with implementation details and test results.

Klíčová slova

Počítačové vidění, segmentace obrazu, Kinect, detekce rukou, OpenCV, Libfreenect

Keywords

Computer vision, image segmentation, Kinect, hand detection, OpenCV, Libfreenect

Citace

ZATLOUKAL, Tomáš. *Ovládání multimedialního přehrávače gesty ruky*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Beran Vítězslav.

Ovládání multimediálního přehrávače gesty ruky

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D.

.....
Tomáš Zatloukal
19. května 2016

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Vítězslavu Beranovi, Ph.D. za odbornou pomoc a užitečné rady k řešení projektu.

© Tomáš Zatloukal, 2016.

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 Vybrané technologie	3
2.1 Zařízení pro snímání gest rukou	3
2.2 Detekce gest v hloubkových datech	7
2.3 Metody zpracování dat	9
2.4 Nástroje pro ovládání přehrávače	11
2.5 Tvorba grafických uživatelských rozhraní	11
2.6 Existující řešení	12
3 Systém pro ovládání multimediálního přehrávače gesty	15
3.1 Definice cílových požadavků	15
3.2 Rozmístění systému	15
3.3 Části systému	16
3.4 Detekce gest	19
3.5 GUI a ovládání aplikace	23
4 Ovládání nové generace	25
4.1 Grafické uživatelské rozhraní	25
4.2 Testování	25
4.3 Možná rozšíření	26
5 Závěr	27
Literatura	28
Přílohy	29
Seznam příloh	30
A Obsah CD	31

Kapitola 1

Úvod

Rychle se rozvíjející počítačové technologie dávají možnosti, které by dříve byly pouze fantazií. Ovládací prvky pro multimediální zařízení se však za ty roky změnily jen nepatrně. Televizi ovládáme nejčastěji klasickým ovladačem, počítač myší a klávesnicí. Jsou to prověřené metody, avšak mohou nastat situace, kdy chceme provést pouze jednoduchou činnost a tyto ovládací prvky nemusí být vždy po ruce.

S jinými přístupy jsme se již mohli nejčastěji setkat u herních konzol, kde se v odlišném ovládní nejlépe prosadil Microsoft se zařízením Kinect. Toto zařízení obsahuje kromě standardní RGB kamery také hloubkový senzor umožňující použít gesta vlastního těla, jako vstupní zařízení. Konzole Xbox s Kinectem měla za účel hráče zvednout ze sedačky a více jej vtáhnout do hry. Brzy se ale zjistilo, že se tento senzor dá používat i k jiným aplikacím, jako je například 3D skenování.

Cílem této práce je umožnit ovládní multimediálního přehrávače pomocí gest ruky. Ke snímání uživatele je použito zařízení Kinect. Data z něj jsou analyzována a detekovaná gesta převedena na stisky klávesnice, aby bylo dosaženo funkčnosti napříč různými přehrávači.

Další kapitola se zabývá senzory pro snímání gest, třetí kapitola nástroji pro zpracování dat z těchto senzorů. Kapitola čtvrtá pojednává o nástrojích pro simulaci stisku kláves. Pátá kapitola popisuje tvorbu grafických uživatelských rozhraní. V šesté kapitole se nachází návrh uživatelského rozhraní, ovládacích gest a aplikace. Sedmá kapitola se věnuje řešení práce a implementaci. Poslední kapitola je zaměřena na testování.

Kapitola 2

Vybrané technologie

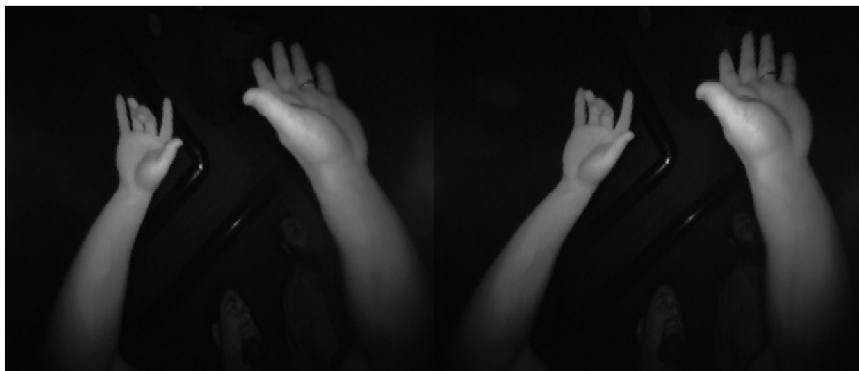
Tato kapitola se věnuje zařízením, technologiím a metodám, které je možno použít pro ovládání přehrávače a detekci gest. Rovněž jsou zde popsána některá existující řešení.

2.1 Zařízení pro snímání gest rukou

Leap Motion

Jedná se o zařízení od stejnojmenné společnosti, které umožňuje použití rukou a prstů jako vstupního zařízení.

Po hardwarové stránce je zařízení velmi jednoduché. Obsahuje dvojici kamer, trojici infračervených LED a hlavní čip. Tento čip synchronizuje osvětlení a kamery tak, aby nesvítilo, když se nepořizuje snímek, a tak šetří energii. Snímky nejprve uloží do své lokální paměti, provede potřebné úpravy rozlišení a následně data posílá po USB. Detekce poté probíhá na straně výpočetního zařízení, ke kterému je připojen.



Obrázek 2.1: Počáteční data ze senzoru Leap motion ¹

Data nejsou převedena na hloubkovou mapu, ale na šedotónový obraz viz obrázek 2.1. Nejprve jsou z obrazu odstraněna nežádoucí data v pozadí (například hlava uživatele) osvětlené okolním světlem. Poté je obraz zpracován a rekonstruován do 3D reprezentace toho, co zařízení vidí. V dalším kroku jsou datům přiřazeny informace typu pozice prstů a dalších článků ruky. Výsledná data jsou poté předány knihovně, které je převedou na objektově orientované struktury. Grafickou reprezentaci výsledku lze vidět na obrázku 2.2.

¹<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>

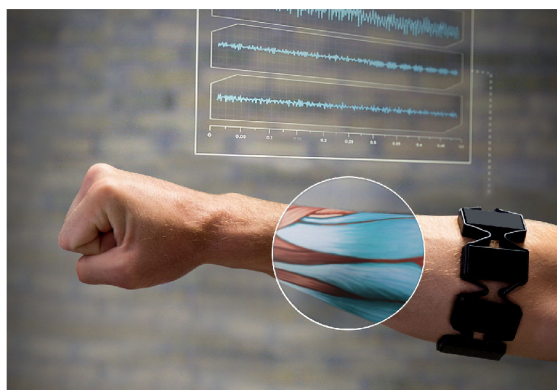


Obrázek 2.2: Ukázka senzoru Leap motion ²

Mezi velké klady lze zařadit velikost samotného senzoru, jeho přesnost a cenovou dostupnost. Omezen je však vzdáleností, na kterou je schopen rozeznání. Maximální vzdálenost je zde 60 centimetrů od senzoru ³.

Myo

Dalším gesta zpracovávající zařízením je Myo od společnosti Thalmic Labs ⁴, který však pracuje naprosto odlišně. Zařízení se nosí na předloktí. Měří elektrickou aktivitu svalstva a to mu umožňuje klasifikaci pěti gest. Spolu s 9-osým IMU (kombinace gyroskopu, akcelometru a magnetometru) je schopné rozumět pohybu a orientaci předloktí. Informace o gestě je přenášena bezdrátově pomocí Bluetooth podporovaným zařízením. Mezi tato podporovaná zařízení patří: Windows, Mac, iOS, Android a Linux. Výhodou tohoto zařízení je nejen jeho přesnost, díky zmíněné kombinaci různých senzorů, ale i to, že je bezdrátový.



Obrázek 2.3: Zařízení Myo ⁵

²<https://leapdev-production-cdn.leapmotion.com/assets/Leap%20Motion%20Technology%20Overview.pdf>

³https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html

⁴<https://www.myo.com/>

⁵<http://developerblog.myo.com/raw-uncut-drops-today/>

RGB kamery

Nejčastěji se můžeme setkat s detekcí za použití obrazu z klasické web kamery. Velkou výhodou je dnes její nízká pořizovací cena a rozšířenost. Mezi zápory pak skutečnost, že v horších světelných podmínkách jsou obrazy z kamer pro detekci nepoužitelné. Lze si zde pomoci IR přísvitem, jako v případě Leap Motion viz sekce 2.1, ale ani pak nejsou výsledky často dostačující. Pro výběr části obrazu s rukou se nejčastěji používá algoritmus detekce kůže, prahování a morfologické operace.

Získání hloubkové mapy je možné použitím dvojice RGB kamer. Takovým zařízením je například ZED od společnosti stereolabs.

ZED

Zed ⁶ je zařízení vyvinuto společností Stereolabs, která se snaží o napodobení lidského vidění. Její produkt ZED je stereo kamera schopná snímat hloubku, založená na pasivní stereo vizi. Výstupem je video, které se skládá ze dvou snímků ve vysokém rozlišení. K výpočetnímu zařízení je připojeno pomocí USB 3.0 kde na grafické kartě je video zpracováno na hloubkovou mapu.

Hloubková data jsou ve vzdálenosti jednoho až patnácti metrů od senzoru. Hlavní využití toto zařízení našlo v autonomní navigaci dronů, sledování pohybu a virtuální realitě. Stereolabs potvrzuje funkčnost uvnitř i venku, nedoporučuje však použití pro segmentaci ruky.



Obrázek 2.4: Zařízení ZED ⁷

Asus Xtion

Toto zařízení od taiwanské společnosti Asus hloubkově snímá scénu podobně jako to dělá Kinect 360 od Microsoftu. Umožňuje detekovat uživatele a sledování jeho kostry. Bylo vyrobeno několik modelů z nichž jeden obsahoval i dvojici mikrofonů pro práci s hlasem.

Oproti kinectu nevyžaduje přídavné napájení a má menší hmotnost. Výhodou je také jeho primární určení pro připojení k PC. Není ovšem tolik populární, což se odrazilo na jeho ceně. Na straně software je podporováno pouze pomocí OpenNI API. Po oznámení podpory kinectu na PC a vydání Kinect SDK společností Microsoft přišel Asus Xtion o svojí výhodu. Zařízení je možné vidět na obrázku 2.5

⁶<https://www.stereolabs.com>

⁷<http://www.luluhypermarket.com/GoodLife/stereolabs-launches-the-new-zed-3d-camera-zzehd529.html>

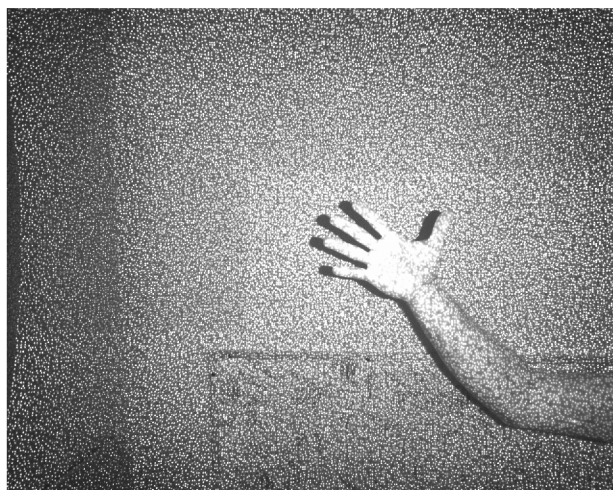
⁸https://www.asus.com/3D-Sensor/Xtion_PRO/



Obrázek 2.5: Senzor Xtion PRO ⁸

Kinect 360

V roce 2010 byl společností Microsoft představen nový druh ovladače pro její herní konzoli Xbox ⁹. Ovladač známý pod názvem Kinect 360, obsahuje RGB kameru, hloubkový senzor, pole mikrofonů a servo motor v podstavci. Zařízení je schopné pořizovat data z obou kamer frekvencí až 30hz při rozlišení 640x480 pixelů. Děň před Kinectem je nasvíceno paprsky přesně definovaným vzorem bodů, jeho vlastním infračerveným zářičem viz obrázek 2.6. Paprsky se od objektu odrazí a jsou snímány CMOS monochromatickou kamerou. Vzdálenost je poté vypočítána dle zpoždění mezi vyzářením bodu infra zářičem a jeho zachycením kamerou.



Obrázek 2.6: Paprsky z IR zářiče

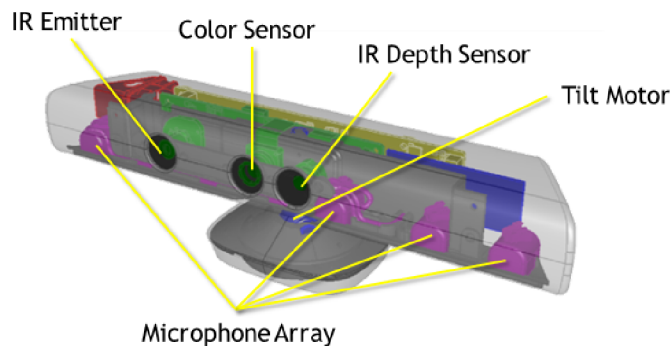
Limitem tohoto zařízení jsou malé předměty, na které není schopen vzor paprsků vysvítit. Vzdálenost objektu od senzoru je udávána od 50 centimetrů do 5 metrů, přesnost se však zhoršuje s vyšší vzdáleností. Doporučená vzdálenost je od 80cm do 3 metrů.

Pole čtyř mikrofonů se dá díky svému rozložení (jeden na pravé straně, tři na levé) použít k lokaci zdroje jimi zachyceného zvuku. Herní konzole, pro kterou byl původně určen, tohoto využívala k lokaci mluvících uživatelů.

Kinect lze připojit ke konzoli nebo k jinému zařízení rozhraním USB. Tento způsob však pro napájení zařízení není dostačující, a proto je třeba použít napájení z externího adaptéru.

⁹<http://www.xbox.com/en-US/xbox-360/accessories/kinect>

¹⁰<https://msdn.microsoft.com/en-us/library/jj131033.aspx>



Obrázek 2.7: Senzor Kinect 360 ¹⁰

2.2 Detekce gest v hloubkových datech

Hloubková data

Další možností detekce je použít jiný druh senzoru, který získává informace o vzdálenosti objektu. Hloubková data lze odlišně segmentovat. Kromě toho nebývají oproti RGB kamerám tolik závislé na okolních světelných podmínkách. Data je možné zpracovat jako 2D obraz nebo jako tzv. Point cloud – množina bodů v trojrozměrném souřadnicovém systému.



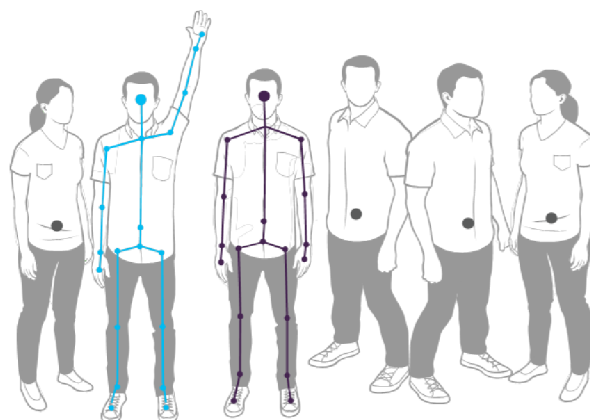
Obrázek 2.8: Hloubková data převedena na 2d obraz, barevný přechod znázorňuje vzdálenost

Microsoft Kinect SDK

V polovině roku 2011 vydala společnost Microsoft software pro vývoj aplikací využívajících jejich zařízení Kinect 360. Tento software bylo možné používat pouze na systému windows 7 v jazycích C++, C# a Visual Basic.

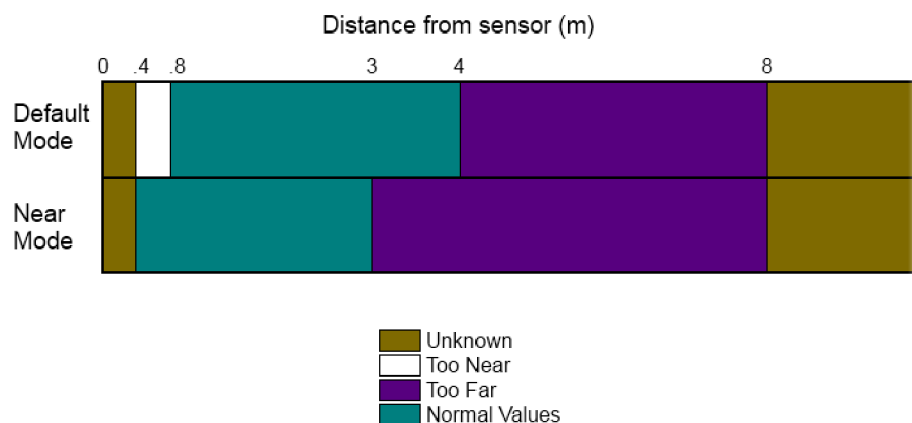
Oproti řešení OpenKinect ^{2.2} byl tento software vysokoúrovňový a poskytoval řadu užitečných funkcí. Mezi ně patří: poskytování dat ze všech senzorů kinetu, sledování kostry uživatele viz obrázek ^{2.9} nebo sledování až šesti uživatelů současně, rozpoznání řeči (pro vybrané jazyky), rekonstrukce scény s chybějícími daty, sledování obličeje a další.

¹¹<https://msdn.microsoft.com/en-us/library/hh973074.aspx>



Obrázek 2.9: Snímání kostry člověka senzorem kinect ¹¹

SDK také dovoľovalo přepínání módu kinectu mezi takzvané „Near Mode“ a „Default Mode“. V módu Near je možné snímat objekty ve vzdálenosti od čtyřiceti centimetrů do dvou až třech metrů se zvýšenou přesností, za cenu nemožnosti snímání objektů za touto hranicí. V defaultním módu lze provádět snímání od osmdesáti centimetrů do čtyř metrů viz obrázek 2.10.



Obrázek 2.10: Přehled módů pro hloubkový senzor kinect ¹²

OpenKinect

Krátce po představení senzoru Kinect 360, začali lidé tuto novinku zkoumat a zpřístupňovat jeho možnosti i pro jiné zařízení, než pro které byl původně určen.

OpenKinect ¹³ je komunita lidí, kteří vytvořili otevřenou knihovnu Libfreenect. Komunita čítá přes 2000 přispívajících členů. Knihovna umožňuje:

- Pořizovat snímky z RGB a hloubkové kamery
- Ovládat náklon kinectu

¹²<https://blogs.msdn.microsoft.com/kinectforwindows/2012/01/20/near-mode-what-it-is-and-isnt/>

¹³https://openkinect.org/wiki/Main_Page

- Získat data z akcelometru (náklon zařízení)
- Ovládat kontrolní LED
- Zachycovat zvuk pomocí pole mikrofonů

Knihovna nemá žádné vyšší funkce jako například detekci člověka a jeho částí, jako je tomu v případě oficiálního SDK od Microsoftu. Mezi podporované jazyky patří například C, C++, Java, Python. Knihovna je multiplatformní a lze jí použít na Linuxu, OSX a Windows. Knihovna samotná je dostupná pod licencí Apache20 nebo GPL2.

OpenCV

Celým názvem Open Source Computer Vision ¹⁴ je svobodná a otevřená knihovna pro počítačové vidění a strojové učení. Je distribuována pod BSD licencí pro snažší komerční využití a úpravy kódu. Obsahuje přes 2500 optimalizovaných algoritmů pro počítačové vidění a strojové učení.

Je užívána například k detekci obličejů, indentifikaci objektů, sledování pohybujících se objektů, sledování pohybu očí. Používána je zaměstnanci velkých firem jako je Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota. Knihovnu lze použít v jazycích C++, C, Python, Java a MATLAB pod operačními systémy Windows, Linux, Android a Mac OS.

Mezi její přednosti bych zařadil její jednoduchost pro implementaci a výbornou dokumentaci s řadou ukázek fungujících programů. Také existenci řady knih, věnující se této knihovně jako například [2].

PCL

Point Cloud Library (PCL) ¹⁵ je značně rozsáhlý, otevřený projekt pro zpracování 2D a 3D obrazu a point cloudu. Tento framework obsahuje řadu algoritmů pro filtrování, rekonstrukci povrchu nebo segmentaci.

Distribuována je pod BSD licencí jako otevřený software. Je zdarma pro komerční i výzkumné účely. Framework PCL je multiplatformní a použitelný pod systémy Linux, MacOS, Windows, Android a iOS. Pro zjednodušení vývoje je rozdělen do série menších knihoven, které mohou být přeloženy odděleně. Tato modulární filozofie je důležitá pro šíření PCL na platformy s omezeným výpočetním výkonem nebo paměťovou kapacitou.

Point cloud může být získán z řady různých senzorů od stereo kamer, 3D skenerů po zařízení typu Microsoft Kinect 2.1 a Asus Xtion 2.1. Příklad vizualizace point cloudu lze vidět na obrázku 2.11

2.3 Metody zpracování dat

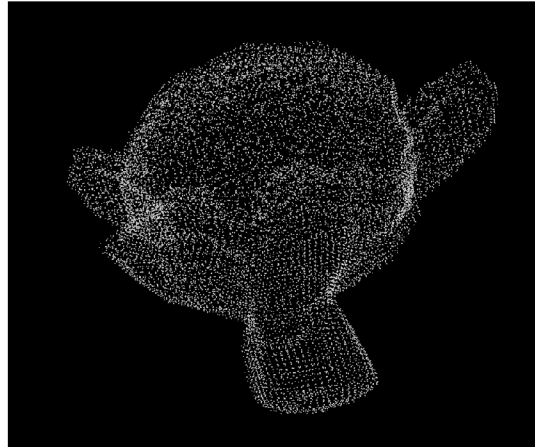
Segmentace

Segmentace obrazu jak uvádí kniha [3], je jedna z disciplín počítačového vidění, jež rozděluje obraz na oblasti, které mají význam pro konkrétní úlohu. Pro segmentaci existuje několik

¹⁴<http://opencv.org/about.html>

¹⁵<http://pointclouds.org/about/>

¹⁶http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_5:_3D_object_recognition_%28pipeline%29



Obrázek 2.11: Vizualizace point cloudu ¹⁶

metod a přístupů, z nichž se volí dle charakteristiky řešeného problému. Je to základní krok pro popis, klasifikaci a rozpoznávání obrazu a jeho složek. Mezi dva hlavní přístupy lze zařadit: regionální metody a hraniční metody. Regionální metody hledají podobnost. Hraniční metody jsou založeny na regionálních rozdílech – hranách. Nejčastěji mají segmentační metody oddělit objekty (popředí) od všeho ostatního (pozadí).

Regionální metody hledají propojené oblasti dle nějaké podobnosti jejich pixelů. Cílem je získat největší možnou propojenou oblast, a tedy získat co nejméně takovýchto oblastí. Jsou-li požadovány příliš podobné pixely, může dojít k přesegmentování obrazu, což může rozdělit samotné objekty. Cílem je tedy najít oblasti, tak jak by je rozdělil člověk, což není jednoduchý úkol.

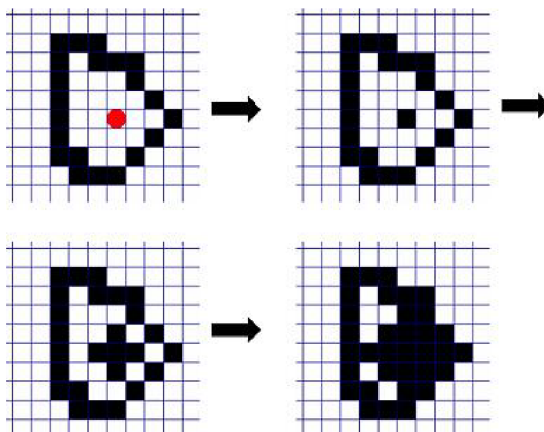
Konstrukce segmentu u regionálních metod probíhá zdola nahoru, od jednoho pixelu (semínka). Poté oblast roste přidáváním sousedů s podobnými vlastnostmi (jas, barva, textura, gradient, geometrické vlastnosti). Pro definici sousedů se používá čtyř nebo osmi okolí.

Hraniční metody jsou naopak založeny spíše na hledání rozdílů než podobností. Cílem je určit nejbližší hranici objektu nebo pozadí. Nejčastěji je možné se setkat s metodou sledování hranic.

Algoritmus semínkového vyplňování

Flood fill nebo také seed fill ^[5] je používán pro vyplnění propojených oblastí rastrových obrázků. Algoritmus ke své funkci potřebuje tři parametry: startovací bod, cílová barva a barva, kterou jsou propojené body obarveny. Vyplňování začíná ve startovacím bodě a postupuje na své sousedy (čtyř nebo osmi okolí). Mají-li stejnou barvu, jsou obarveny na barvu cílovou. Existují i modifikace této algoritmu, kde je možné zadat hodnotu rozmezí rozdílu mezi sousedními body. Takto lze tuto metodu využít například k získání objektů ze snímku kamery.

¹⁷http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#floodfill



Obrázek 2.12: Algoritmus Flood fill ¹⁷

2.4 Nástroje pro ovládání přehrávače

Xdotool

Tento nástroj ¹⁸ umožňuje simulovat vstup z klávesnice nebo myši, přesouvat okna aplikací, měnit jejich velikost a další. Využívá rozšíření XTEST spolu s knihovnu Xlib. Podporuje systémy Debian, Ubuntu, FreeBSD, Fedora, Mac OS X a OpenSUSE. Jeho použití je velmi jednoduché a pracuje jako konzolový program.

AutoIt

AutoIt ¹⁹ je skriptovací jazyk připomínající Basic. Byl vytvořen pro automatizaci Windows GUI a běžné skriptování. Umožňuje simulovat stisky kláves, pohyb myši a manipulaci s okny aplikací. Nepodporuje žádný jiný operační systém mimo Windows, za to podporuje mnoho verzí tohoto systému.

AutoIt je zdarma ke stažení na stránkách vývojáře.

2.5 Tvorba grafických uživatelských rozhraní

Qt

Qt ²⁰ je multiplatformní framework pro vývoj aplikací pro desktop, vestavěná zařízení a mobilní telefony. Mezi podporované systémy patří Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS a další.

Qt sám o sobě není programovací jazyk. Je to framework napsaný v C++. Použitý preprocesor MOC (Meta-Object Compiler) zde slouží k rozšíření C++ o signály a sloty. Před kompilací MOC prochází zdrojové soubory napsané v Qt rozšiřujícím C++ a generuje z nich standardní kód v C++. Přeložení je možné běžným způsobem za použití C++ překladače, jako je Clang, GCC, ICC, MinGW a MSVC.

¹⁸<http://www.semicomplete.com/projects/xdotool/>

¹⁹<https://www.autoitscript.com/site/autoit/>

²⁰http://wiki.qt.io/About_Qt

Společnost Qt prodává licence pro komerční účely. Dostupný je však také jako svobodný software pod licencemi GPL a LGPL. Existuje mnoho knih o tomto frameworku, některé se specializují přímo na konkrétní jazyk [1].

IDE

Společnost Qt také vytvořila svoje vlastní integrované vývojářské prostředí (IDE) nazvané **Qt Creator**. Pod systémy Linux, OS X a Windows nabízí inteligentní dokončování kódu, zvýraznění syntaxe, integrovanou systémovou nápovědu, debugger, nástroj pro profilování a integrace verzovacích systémů (git a Bazaar). Na systému Windows také existuje doplněk pro Visual Studio. Samozřejmostí je možnost použití zcela jiného IDE.

Widgets

S Qt se dají rovněž psát GUI aplikace přímo v jazyce C++ za pomoci Qt Widgets modulu. Dostupný je také interaktivní grafický nástroj nazvaný **Qt Designer**, který slouží ke generování zdrojových kódů pro tvorbu GUI. Qt Designer může být používán sám o sobě, ale je také integrován v Qt Creator.

QtQuick

Dalším modulem umožňujícím tvorbu GUI s Qt je QtQuick. Grafická rozhraní používající QtQuick jsou psána v QML. QML je deklarativní objektově orientovaný jazyk, který obsahuje Javascript pro procedurální programování. QtQuick poskytuje potřebné moduly tvorbu GUI v QML. Je možné napsat kompletní aplikaci pouze v QML, ale obvykle je v QML napsáno jen GUI a backend v C++. Qt Creator také obsahuje QtQuick GUI návrhový nástroj a profiler.

GTK+

Známý i pod názvem GIMP Toolkit ²¹ je multiplatformní knihovna pro tvorbu grafických uživatelských rozhraní. Podporuje Linux, Windows a OS X. Vhodný je jak pro malé i velké projekty. Podporuje řadu platforem a chlubí se jednoduchým API, šetřícím vývojářův čas.

Je napsán v programovacím jazyce C, ale je navržen tak, aby podporoval řadu dalších jazyků mimo C a C++. Dostupný je také návrhový nástroj **Glade**, který generuje kód v XML. Toto XML je možné použít v řadě programovacích jazyků jako C, C++, C#, Vala, Java, Perl, Python.

GTK+ je svobodný software a součást projektu GNU. Licence GNU LGPL umožňuje využívání GTK+ všemi vývojáři pro nejrůznější projekty.

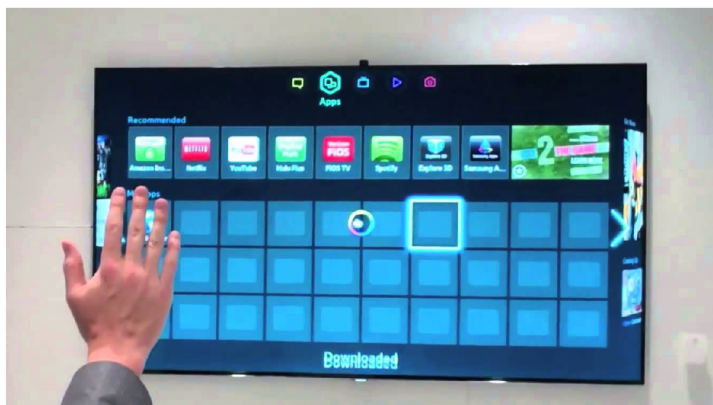
2.6 Existující řešení

Samsung Smart TV

Firma samsung již v roce 2012 ohlásila, že její televize bude možné ovládat gesty ruky a také hlasem. V prvních verzích sloužila ruka jako virtuální myš. Ruka uživatele ovládala pohyb kurzoru na televizi a sevřením ruky v pěst došlo k potvrzení – kliknutí. V dnešní

²¹GTK<http://www.gtk.org/>

verzi je již toto ovládání zdokonalené. Podporuje hned několik gest, zahrnující v některých případech obě ruce. Toto ovládání je umožněno kamerou v rámu televize.



Obrázek 2.13: Ukázka ovládání gesty televize Samsung ²²

LG Smart TV

Další společností, která se rozhodla zabudovat do svých televizí nestandardní ovládání je LG. V roce 2013 představila nové modely svých televizí, které umožňovaly ovládání gesty ruky. Toto ovládání sloužilo hlavně k vypnutí televize, přeoínání kanálů, a ovládání hlasitosti bez použití klasického ovladače. Ke snímání uživatele slouží kamera zabudovaná v horní části rámu televize. Při detekci uživateli ruky televize zobrazí menu tlačítek. Mezi nimi se uživatel přemísťuje pohybem ruky a sevřením v pěst dojde k potvrzení.



Obrázek 2.14: Ukázka ovládání gesty televize LG ²³

ControlAir

Jedním z produktů společnosti eyeSight, zabývající se počítačovým viděním pro ovládání zařízení je ControlAir. Tato aplikace využívá pouze webovou kameru zabudovanou v zařízení pro jednoduché ovládání přehrávače. Podporuje pouze vybrané aplikace jako je: iTunes,

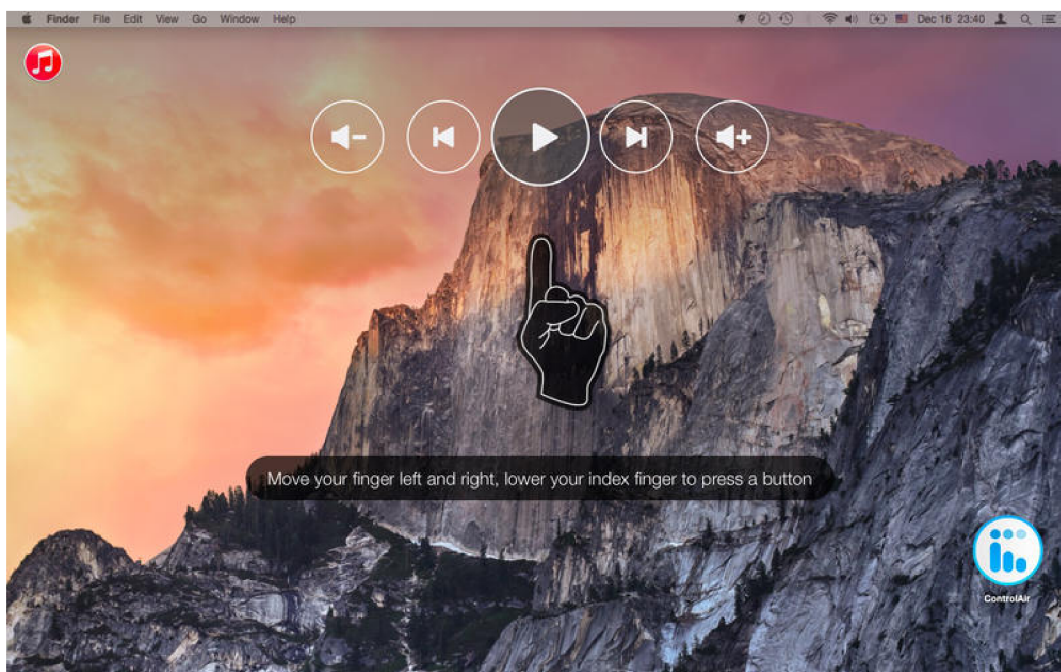
²²<https://www.youtube.com/watch?v=wDmosRnEfiw>

²³https://www.youtube.com/watch?v=900PY_JWSq0

QuickTime Player, Rdio, Spotify, VLC a VOX. Nejedná se o multiplatformní aplikaci. Podporovány jsou pouze počítače Mac, vyrobené od roku 2011 společností Apple.

Aplikace se ovládá ukazováčkem. Jakmile systém detekuje ruku uživatele se zdviženým ukazováčkem zobrazí menu pro ovládání viz obrázek 2.15. Mezi nimi se uživatel přesouvá horizontálním pohybem ruky a pokrčením prstu k displeji dojde k potvrzení. Zdvížením ukazováčku před ústy dojde k utlumení hlasitosti nebo opětovnému zesílení.

Toto bezdotykové ovládání funguje pouze v případě, že uživatel je otočen čelem k zařízení a je vzdálen od padesáti centimetrů do jednoho a půl metru.



Obrázek 2.15: Ukázka aplikace ControlAir ²⁴

²⁴<https://itunes.apple.com/us/app/controlair/id950009491?mt=12>

Kapitola 3

System pro ovládání multimediálního přehrávače gesty

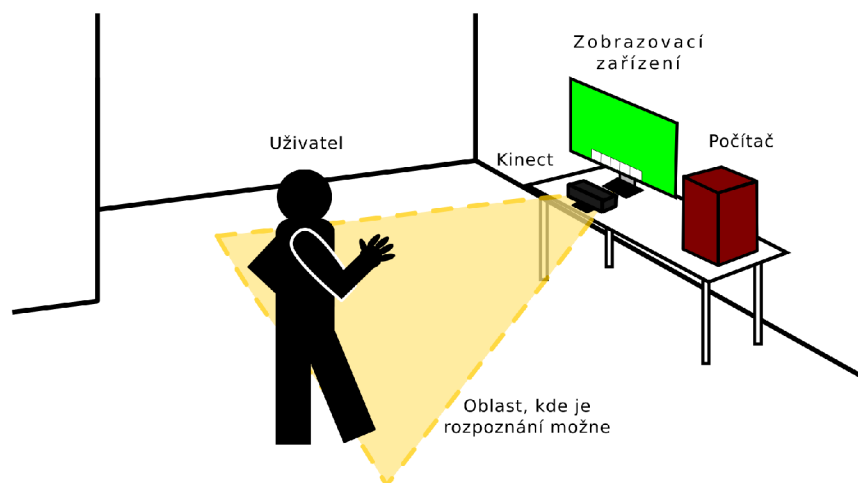
3.1 Definice cílových požadavků

Ze zadání této práce vyplývá řada požadavků na výsledný prvek. Ten má využívat hloubkový senzor pro snímání scény s uživatelem. V hloubkových datech detekovat gesta ruky a ta klasifikovat. Navržené rozhraní musí umožňovat ovládání multimediálního přehrávače pomocí těchto gest.

3.2 Rozmístění systému

Mezi hlavní části systému patří:

- Kinect – snímá scénu s uživatelem
- Počítač – zde běží celá aplikace
- Zobrazovací zařízení – zobrazuje grafické rozhraní aplikace

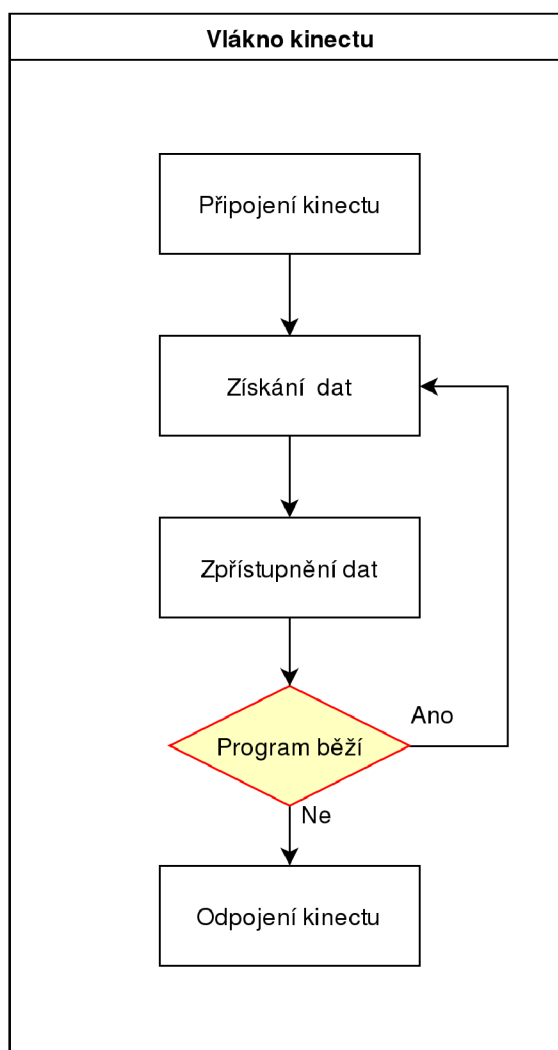


Obrázek 3.1: Znázornění rozmístění systému

3.3 Části systému

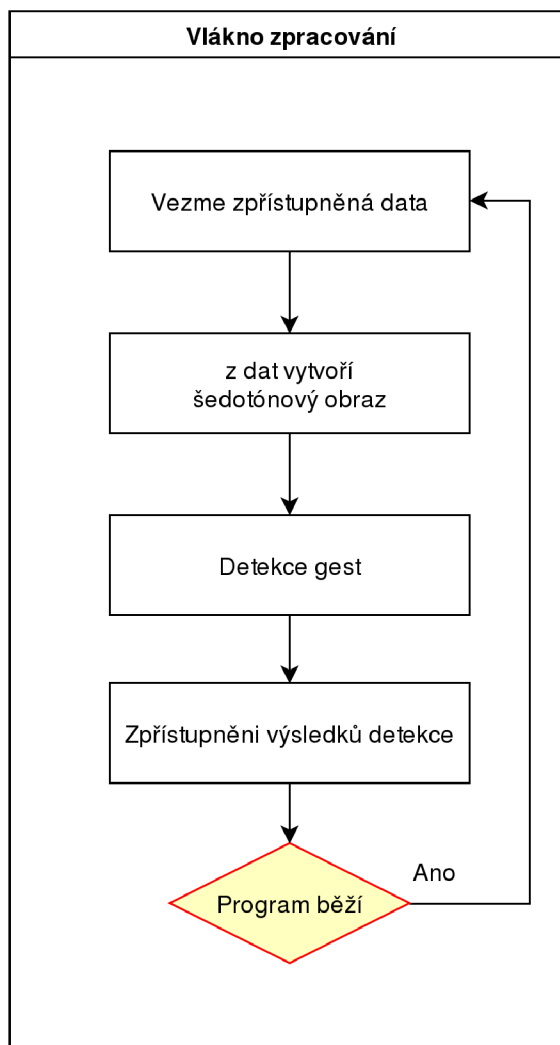
Program je rozdělen do tří částí, kde každá část má jiný úkol. První část pracuje s kinectem – pořizuje data. Druhá část data zpracovává a detekuje v nich ruku uživatele. Třetí část dle výsledků zpracování vykresluje GUI a dává povely přehrávači. Všechny tyto části jsou realizovány jako vlákna. Přístup ke sdíleným prostředkům je řešen zámky. Povely pro přehrávač jsou vykonány simulací stisku multimediálních kláves, aby bylo docíleno kompatibility s největším počtem přehrávačů.

Vlákno kinectu



Obrázek 3.2: Vlákno pro pořizování dat z kinectu

Při spuštění programu se nejprve zkontroluje, zda je možné zařízení použít, poté s ním pracuje pouze toto vlákno. Vlákno se k zařízení připojí a nastaví s ním asynchronní komunikaci. Když tedy kinect pořídí data, je volána funkce, která data vyzvedne a zpřístupní je vláknu pro zpracování. Pokud nastane chyba (např. odpojení kinectu) vlákno nastaví příznak pro ukončení programu a samo se ukončí.



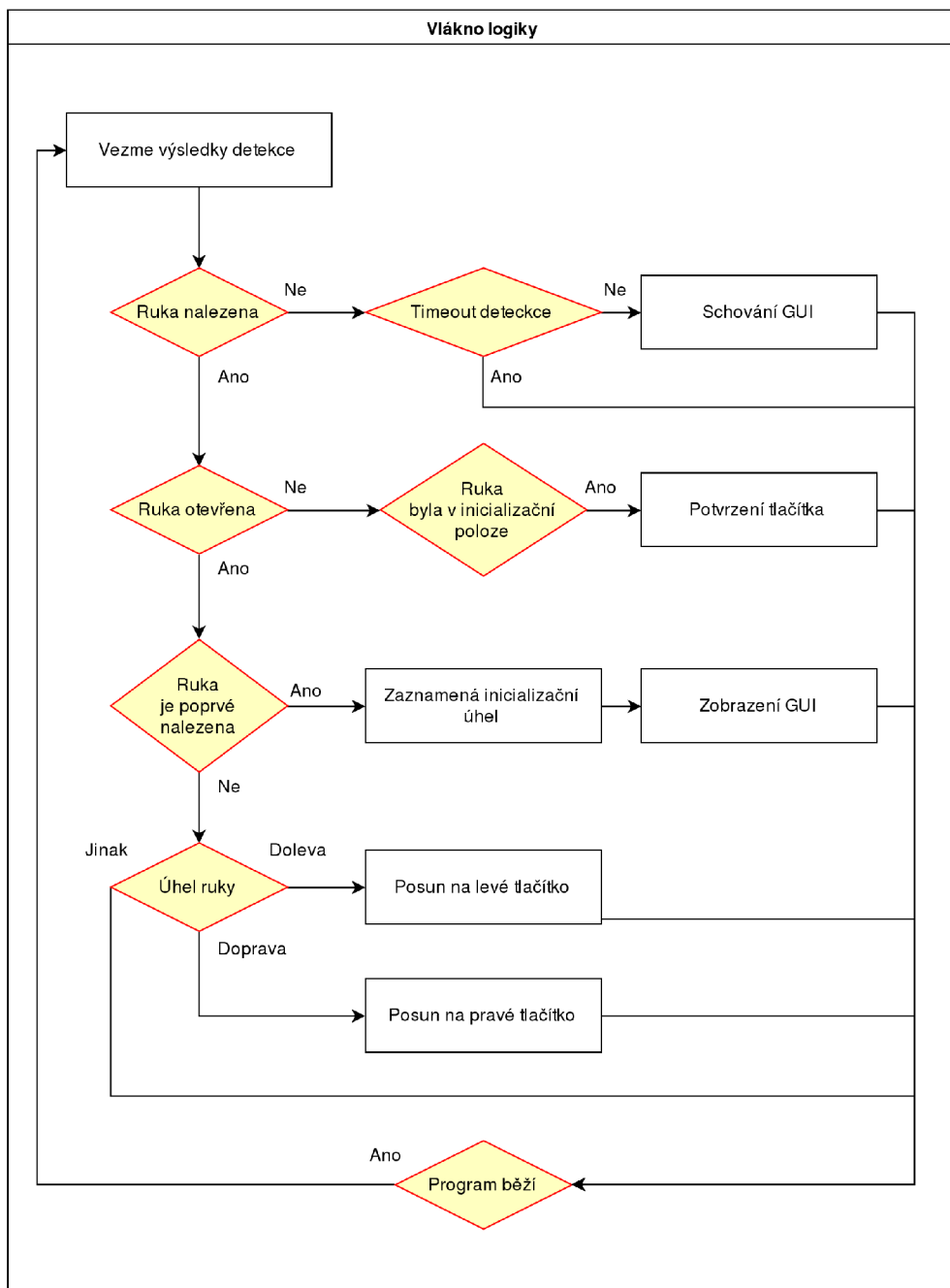
Obrázek 3.3: Vlákno pro zpracování hloubkových dat

Vláknó zpracování

Toto vlákno vykonává hlavní a nejvíce náročnou část programu. Nejprve z hloubkových dat vytvoří šedotónový obraz, poté za pomoci tohoto obrazu i hloubkových dat detekuje a klasifikuje gesta. Pokud byla ruka nalezena, pak se do struktury uloží údaje o její pozici, vzdálenosti a úhlu natočení a také času nalezení. V opačném případě je ve struktuře nastaven příznak nalezení na `false`. Schéma vlákna zpracování je vidět na obrázku 3.3

Vláknó logiky

Úkolem tohoto vlákna je vykreslení GUI a zadávání povelů pro přehrávač. Vlákno čeká dokud nemá data ze zpracování připravena, poté je probuzeno. Na základě těchto dat provede následující akce.



Obrázek 3.4: Vlákno pro GUI

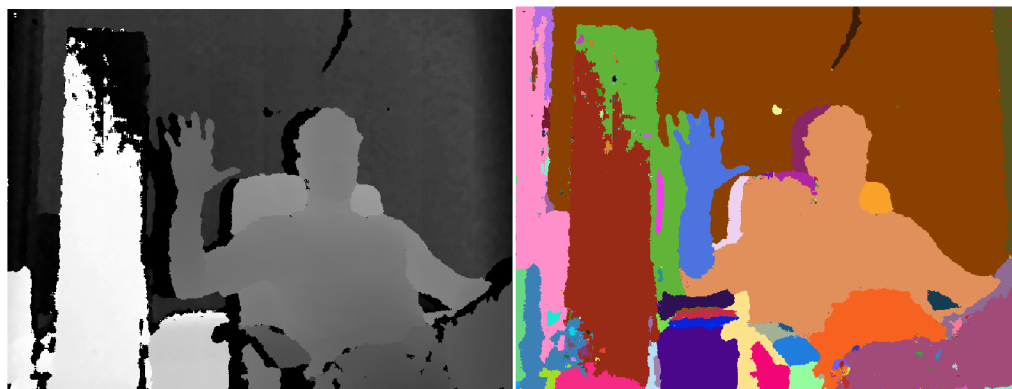
3.4 Detekce gest

Segmentace hloubkových dat

Data z kinectů jsou získána jako pole vzdáleností v milimetrech pro jednotlivé pixely. Tato data jsou lineárním mapováním převedena na obraz ve stupních šedi. Pokusně bylo zjištěno, že ruku vzdálenou více než tři metry od senzoru již není možné na obraze rozpoznat. Scéna je proto oříznuta do vzdálenosti tří metrů od senzoru. Výsledek lze vidět na obrázku 3.5 vlevo. Bližší objekty k senzoru jsou tmavě šedé, vzdálenější postupně přechází do barvy bílé.

Na získaném obraze z předešlé kapitoly je nyní potřeba najít jednotlivé objekty pro další zkoumání. Obraz je procházen v cyklu pixel po pixelu. Na každý pixel (pokud již nebyl zpracován předešlým průchodem nebo není černý – kinect nic nevidí) je aplikována metoda semínkového vyplňování [4]. Spojené komponenty obarvuje zadanou barvou. Zda jsou komponenty spojeny lze definovat parametry pro maximální a minimální rozdíl jasu nebo barvy mezi aktuálně pozorovaným pixelem (semínkem) a jeho sousedem.

Na obrázku 3.5 jsou vidět barevně oddělené segmenty – objekty. Obraz ovšem není potřeba procházet celý, jakmile je nalezena ruka, není pro použití v této aplikaci důvod obraz dále segmentovat a více zpracovávat.



Obrázek 3.5: Vlevo segmentovaný obraz. Vpravo zdrojový obraz po segmentaci.

Funkce implementující semínkové vyplňování v OpenCV také nastaví masku právě obarvených bodů. Tento binární obraz lze prohlásit za segmentovaný objekt viz 3.6. Funkce navíc vrací počet obarvených bodů. Tuto informaci lze využít pro jednoduchý filtr. Pokud je tento počet příliš malý, je možné říci, že se jedná o šum nebo příliš malý objekt a netřeba jej tedy zkoumat. Masku převyšující minimální počet obarvených bodů je dále analyzována.

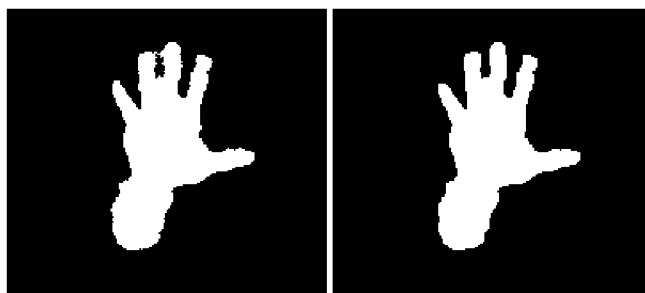
Klasifikace gest

Získaný obraz ruky je často pro detekci nedokonalý. Například pokud jsou prsty příliš u sebe, pak na obraze z kinectů splývají. Je proto potřeba využít morfologických operací, konkrétně eroze a dilatace. Eroze obraz zredukuje, čímž oddělí prsty od sebe. Pokud je ovšem ruka od kinectů vzdálená a prsty se na snímku jeví malé, je zde nebezpečí, že by eroze mohla zredukovat i část prstů. Proto je ihned po ní aplikována dilatace, která objekt naopak zvětší. Výsledek operace je vidět na obrázku 3.7.

Nyní je k dispozici finální snímek a je tedy možné začít s jeho analýzou. Nejprve je použita funkce pro nalezení kontur objektu. Je parametrem nastavena tak, aby se nezanořovala,



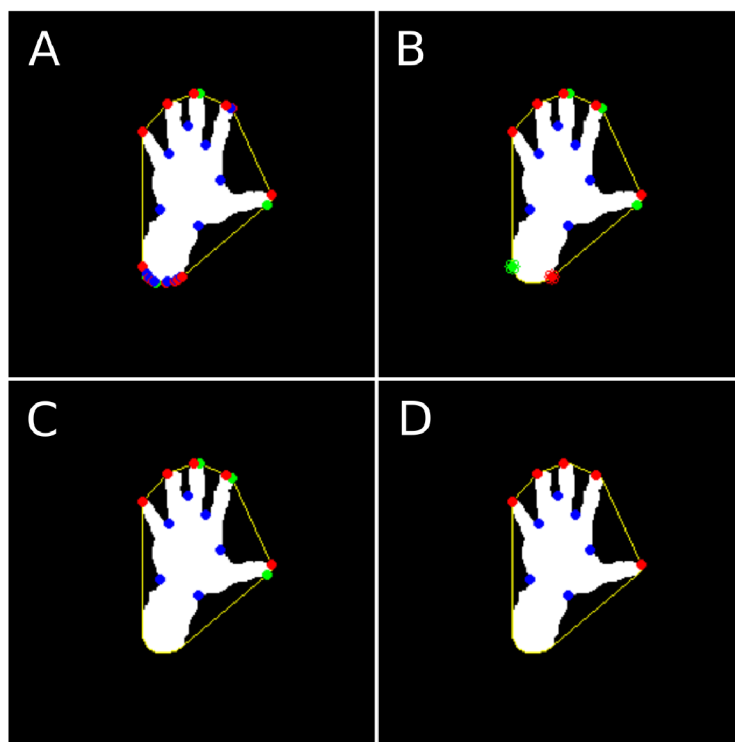
Obrázek 3.6: Binární masky objektů získané segmentací



Obrázek 3.7: Vlevo obraz před provedením morfologických operací. Vpravo po aplikaci morfologických operací.

a tedy hledala jen největší obrys na vstupním obrazu.

Dále je okolo obrysu sestrojena konvexní obálka. První jednoduchý filtr odstraní takové objekty, jejichž rozdíl ploch obrysu a konvexní obálky je minimální. Takový objekt odpovídá spíše pěsti než otevřené ruce, a proto je zahozen a je zkoumán další objekt. Jsou nalezeny konvexní defekty viz obrázek 3.8 A. Získané body defektů jsou dále redukovány dle hloubky defektu 3.8 B. Dále jsou odstraněny body konce viditelné části ruky 3.8 C. Body začátků a konců defektů se již příliš neliší, a proto jsou začátky odstraněny. Obraz s výslednými body prstů a defektů lze vidět na obrázku 3.8 D.

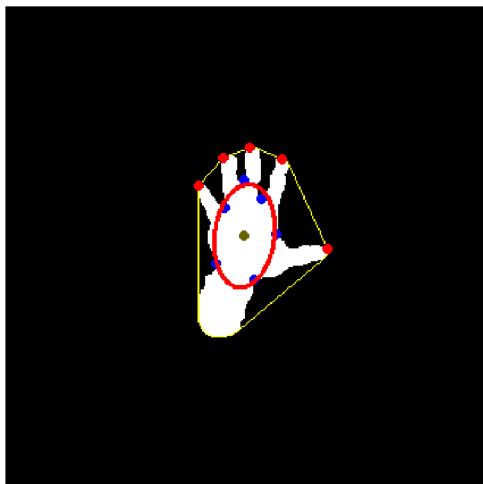


Obrázek 3.8: Červeně jsou znázorněny začátky defektů, modře nejvzdálenější bod defektu, zeleně konec defektu, žlutě konvexní obálka. A) Nalezené defekty. B) Vyfiltrované užitečné body. C) Po odstranění bodů konce ruky. D) Výsledné body pro detekci.

Následně je zkontrolováno, zda je počet prstů (červené body na obrázku 3.8 D) roven pěti. Body defektu (modré body na obrázku 3.8) jsou použity k sestrojení elipsy. Elipsa by měla v ideálním případě přesně kopírovat tvar dlaně, ale nemusí tomu tak vždy být, jako je tomu na obrázku 3.9. Je proto testován tvar elipsy s jistou tolerancí, následně její přibližná reálná velikost. K zjištění vzdáleností dvou bodů je použita funkce z knihovny libfreenect, která převádí x-ové a y-nové souřadnice bodu v pixelech na souřadnice v milimetrech. Přesnost této metody se liší dle vzdálenosti od senzoru. Reálné souřadnice jsou vypočítány vzorcem pro vzdálenost dvou bodů, odvozeným z Pythagorovy věty. Takto je testována výška a šířka elipsy – dlaně.

Pro další zpracování je třeba seřadit prsty v pořadí palec, ukazováček, prostředníček, prsteníček, malíček nebo obráceně. Nejprve se vyberou body pro palec a malíček, ty svírají největší úhel. Jeden z těchto bodů je vybrán a zařazen na první pozici pro posloupnost prstů. Ostatní prsty jsou seřazeny podle úhlu, které svírají s prstem na první pozici.

Po seřazení je ověřena velikost prstů, konkrétně vzdálenost bodu konce prstu a středu



Obrázek 3.9: Obrýs dlaně vytvořený z konvexních defektů

dlaně. Kontrola probíhá podobně jako u testování velikosti dlaně v reálných jednotkách.

Nakonec je zkoumán úhel mezi sousedními prsty, který by až na úhel mezi palcem a ukazováčkem neměl přesáhnout 60 stupňů.

Pokud objekt prošel všemi testy je prohlášen za ruku. Je naplněna struktura s informacemi o nalezené ruce a další objekty se již neanalyzují. Úhel natočení ruky je spočítán jako průměr úhlů jednotlivých prstů.

Klasifikace gesta potvrzení

Gesto ruky sevřené v pěst je klasifikováno velmi odlišně. Vychází se z předpokladu, že mu předcházelo nalezení ruky a tedy je známá poloha středu dlaně – modré kolečko na obrázku 3.10 vpravo. Objekt na této pozici je rovněž vysegmentován metodou semínkového vyplňování. Je nalezen obrýs objektu a jeho konvexní obálka, ty jsou vyznačeny červeně a modře na obrázku 3.10 vpravo. Je testován rozdíl ploch obrýsu zkoumaného objektu a obrýsu otevřené ruky, která mu předcházela. Ta by měla mít plochu menší. Dále je testován rozdíl ploch obrýsu a konvexní obálky, ten by měl být minimální. Projde-li objekt těmito testy je považován za pěst. Výsledky klasifikace jsou rovněž zaznamenány ve struktuře s informacemi o ruce.

Inicializační gesto

Pokud ruka doposud nebyla nalezena, pak se uloží úhel natočení ruky jako inicializační úhel. Gesta pro změnu tlačítka jsou dále odvozeny relativně od tohoto úhlu. Dále je vykresleno GUI s tlačítky symbolizující povely pro přehrávač a čeká se na další gesto.

Gesto změny tlačítka

Změna zvoleného tlačítka proběhne pouze v případě, že mu předcházelo gesto v inicializační poloze. Tedy buď byla ruka nalezena nebo navrácena do původního natočení. Tím je předcházeno situaci, kdy by se uživatel přepnul o více tlačítek, než měl v úmyslu. Pro posun do zvoleného směru je tedy třeba natočit ruku tímto směrem a opět vrátit do původní polohy pro další akci.



Obrázek 3.10: Vlevo zdrojový obraz pro detekci pěsti. Vpravo detekovaná pěst.

Potvrzovací gesto

Podobně jako gesto pro změnu tlačítka i potvrzení zvoleného tlačítka, je provedeno pouze pokud tomuto gestu předcházelo gesto inicializační. Sevřením ruky v pěst dojde k potvrzení zvoleného tlačítka. Povel je přehrávači předán simulací stisku multimedialního tlačítka klávesnice. Tímto způsobem je docíleno kompatibility s větším počtem přehrávačů.

Optimalizace

Pro zlepšení odezvy a celkového fungování systému byly implementovány tyto optimalizace.

Při provádění gesta pro posun na další tlačítko GUI, je obraz rychlým pohybem deformován tak, že prsty uživatele splývají. Tato deformace se nejčastěji týká prostředníčku a jednoho ze sousedních prstů. Proto jakmile byla ruka detekována (bylo nalezeno 5 prstů), klasifikátorem projde i ruka se čtyřmi prsty.

Dálší optimalizace se snaží řešit výpadky sledování ruky, který způsobuje šum i při nehybné scéně. Optimalizace spočívá v nastavení časové tolerance výpadku, která je vyšší než výpadky způsobené šumem. Teprve až po překročení času nastavené tolerance je ruka prohlášena za ztracenou a je schováno GUI.

Poslední optimalizace redukuje scénu a je tedy počet analyzovaných objektů pro detekci ruky. Po nalezení ruky je zaznamenán bod středu dlaňe, a také vzdálenost tohoto bodu od kinect. Dokud není ruka ztracena, je tato hloubka předávána jako parametr funkce pro vytváření šedotónového obrazu. Tato funkce pak vytváří obraz pouze s body, jejichž hloubka je 20 centimetrů před nebo 20 centimetrů za tímto bodem.

3.5 GUI a ovládání aplikace

Pro snadné ovládání přehrávače nestačí jen detekovat gesta a převádět je na povely pro přehrávač, ale je vhodné ovládání zobrazovat graficky.

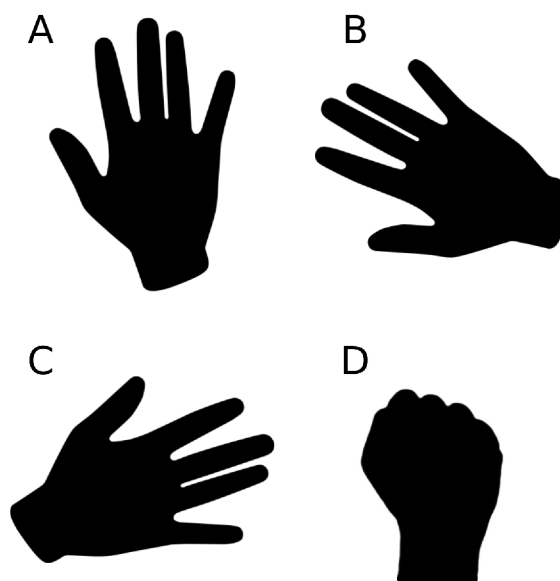
Pro pohodlné ovládání postačí 4 gesta viz obrázek 3.12. Jedno gesto pro zobrazení ovládacího panelu viz obrázek 3.11, dvě pro přepínání mezi tlačítky a poslední pro zvolení tlačítka.

Inicializačním gestem uživatel vyvolá ovládací panel s tlačítky pro jednotlivé povely pro přehrávač. Mezi nimi se uživatel pohybuje natočením ruky do požadovaného směru. Chce-li tedy uživatel zvolit tlačítko úplně vpravo, musí nejprve natočit ruku, vrátit jí zpět a znovu



Obrázek 3.11: Návrh grafického uživatelského rozhraní programu

natočit. Pro zvolení tlačítka uživatel ruku zatne v pěst. Poté je možné provádět další úkony opětovným otevřením dlaní v inicializační poloze. Pokud ruka není detekována ovládací panel zmizí, aby na zobrazovacím zařízení nepřekážel.



Obrázek 3.12: Návrh gest pro ovládání programu. A) gesto pro zobrazení GUI B) gesto pro posun doleva C) gesto pro posun doprava D) gesto potvrzení

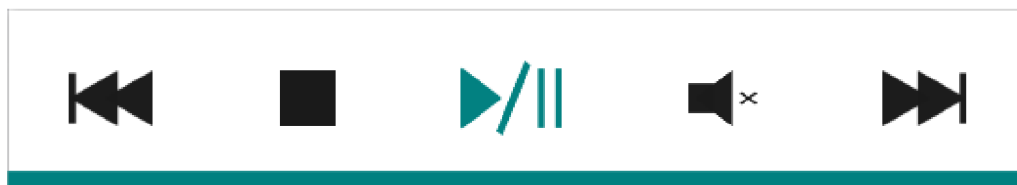
Kapitola 4

Ovládání nové generace

Program je psán v jazyce C++. Pro komunikaci s kinectem používá knihovnu libfreenect viz sekce 2.2, pro zpracování obrazu knihovnu OpenCV ze sekce 2.2. GUI je vykresleno pomocí knihovny Qt. Stisky multimediálních kláves jsou simulovány za použití nástroje `xdotool`.

4.1 Grafické uživatelské rozhraní

Rozhraní bylo vytvořeno dle návrhu na obrázku 3.11 v programu Qt Creator a grafika vytvořena programem pro tvorbu vektorové grafiky Inkscape. Výsledné rozhraní je vidět na obrázku 4.1. Vybrané tlačítko je zvýrazněno zelenou barvou. Panel se při nalezení ruky uživatele vykreslí uprostřed spodní části zobrazovacího zařízení.



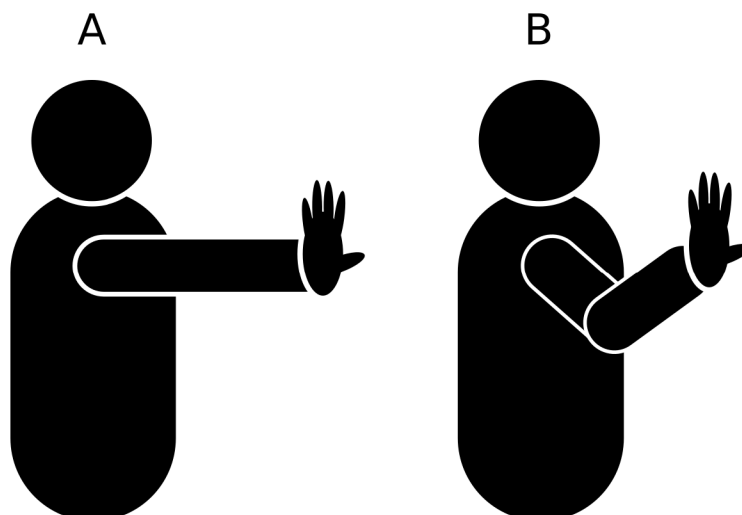
Obrázek 4.1: Finální podoba GUI

4.2 Testování

Systém byl testován jako celek s deseti uživateli různých věkových kategorií na operačním systému Fedora verze 23. Zaznamenány byly údaje o výpadku sledování ruky, odlišném chování než uživatel předpokládal a zpětné vazbě od uživatelů.

Testování systému

Při prvním testu bylo uživatelům vysvětleno, jak ovladač přehrávače používat a následně ho sami vyzkoušeli. První návrh systému předpokládal, že uživatel při ovládání bude mít ruku před sebou napnutou viz obrázek 4.2 A. Při testech se ukázala nedokonalost tohoto řešení spojená s bolestí zápěstí při delším používání. Systém poté prošel změnami, aby bylo umožněno ovládání s rukou pokrčenou v loktu viz obrázek 4.2 B a následně se testy opakovaly.



Obrázek 4.2: Změny v návrhu ovládání. A) první návrh B) finální návrh

Výsledkem finální sady testů bylo zjištění, že ne všichni uživatelé jsou schopni roztáhnout prsty od sebe, tak aby na snímku z kinectu nesplývaly. Takoví uživatelé nebyli schopni ovladač používat při stejném roztažení prstu na vzdálenost vyšší než 1.3 metrů. Uživatelé schopni roztažení prstů neměli problém s používáním přehrávače na vzdálenost až 3 metry.

Dalším výsledkem testování byl poznatek, že uživatelé i po instruování o rychlosti změny gest pro pohyb na jiné tlačítko, vykonávali toto gesto příliš rychle. Jak již bylo zmíněno v sekci 2.1, kinect použitý jako hloubkový senzor snímá scénu frekvencí 30hz. Při rychlejším pohybu dojde k splnutí prstů a nemožnosti detekce ruky viz obrázek 4.3.



Obrázek 4.3: Rozmazané gesto příliš rychlým pohybem ruky.

4.3 Možná rozšíření

Případné možnosti rozšíření aplikace:

- Zpracovávat snímek pouze pokud došlo ve scéně k pohybu.
- Detekovat ruku na základě nalezení uživatele (detekce obličeje nebo celého těla)
- Implementovat možnost změny barevného stylu grafického rozhraní ovladače.

Kapitola 5

Závěr

Cílem této práce bylo navrhnout a implementovat prvek schopný ovládat multimediální přehrávač gesty rukou. Uvedeny jsou zde metody, nástroje a zařízení, které je možné použít pro splnění tohoto zadání.

Pro řešení práce bylo potřeba nastudovat metody pro zpracování hloubkových dat, segmentaci a zpracování obrazu, detekci gest a návrh uživatelských rozhraní. Seznámit se s knihovnamí vhodnými pro tuto práci a nastudovat jejich dokumentaci. Na základě získaných znalostí jsem byl schopen navrhnout gesta pro ovládání přehrávače. Tato gesta detekovat v hloubkových datech a následně je klasifikovat. Systém prošel několika iteracemi a byl postupně testován.

Výsledný systém je možné použít s řadou přehrávačů podporujících ovládání multimediálními klávesy, například VLC. Jako hloubkový senzor bylo použito zařízení kinect 360. Díky použitému senzoru a principu jeho hloubkovému snímání je systém schopen pracovat i v úplné tmě. Kinect snímá scénu před ním s uživatelem a data posílá do počítače, ke kterému je připojen. Při zpracování těchto dat jsou objekty ve scéně segmentovány a analyzovány. Pokud systém detekuje ruku, klasifikuje následně gesto, které je převedeno na povel pro přehrávač.

Cílovou platformou byl zvolen Linux. Portace na jiné platformy by byla možná díky zvoleným technologiím, které jsou, až na xdotool pro simulaci stisku kláves, multiplatformní. Možná náhrada za xdotool je zde také popsána.

Výsledek této práce by mohl sloužit k dálkovému ovládání přehrávače na počítači stejně jako na multimediálním centru. Výsledný klasifikátor by také bylo možné použít pro jiné způsoby ovládání typu virtuální myš. Možná vylepšení na straně softwaru jsou zde také popsány. Použitím dokonalejšího senzoru by umožnilo zaznamenat rychlejší pohyby uživatele, tedy i rychlost interakce a přesnější snímání scény.

Literatura

- [1] Blanchette, J.: *C++ GUI programming with Qt 4*. Upper Saddle River: Prentice-Hall, 2008, ISBN 978-0-13-235416-5.
- [2] Gary Bradski, A. K.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0.
- [3] Milan Sonka, R. B., Vaclav Hlavac: *Image Processing, Analysis, and Machine Vision*,. THOMSON, 2008, ISBN 978-0495082521.
- [4] OpenCV: floodFill. online.
URL http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#floodfill
- [5] Wikipedia: Flood fill. online.
URL https://en.wikipedia.org/wiki/Flood_fill

Přílohy

Seznam příloh

A Obsah CD

31

Příloha A

Obsah CD

Příložené CD obsahuje:

- adresář se zdrojovými kódy.
- adresář s anotovanou testovací sadou.
- adresář se zdrojovými kódy této technické zprávy.
- tuto technickou zprávu v PDF.
- video a plakát prezentující tuto práci.