

**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**

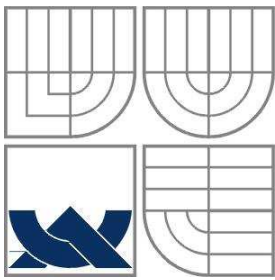
**OVLÁDÁNÍ POČÍTAČE GESTY**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

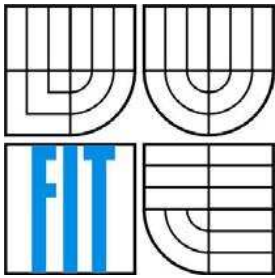
**AUTOR PRÁCE**  
AUTHOR

**JAN CHALOUPKA**

**BRNO 2011**



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 POČÍTAČE GESTY

GESTURE-BASED HUMAN-COMPUTER INTERFACE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

VEDOUCÍ PRÁCE  
SUPERVISOR

JAN CHALOUPKA

Ing., MICHAL ŠPANĚL, Ph.D.

BRNO 2011

## **Abstrakt**

Tato bakalářská práce pojednává o návrhu a implementaci jednoduchého rozhraní pro rozpoznávání gest rukou. Rozhraní slouží jako rozšíření aktuálních ovládacích prvků. Pro segmentaci gest je použit model barvy kůže, pro rozpoznání gest význačné body ruky (dlaň, prsty, konečky prstů). Navržené rozhraní je použito pro ovládání jednoduchého prohlížeče fotografií. Výsledná aplikace je schopna základních gest pro změnu velikost, rotaci, ukazování a kliknutí.

## **Abstract**

This bachelor's thesis is about design and implementation of simple interface for hand gesture recognition. The interface is used as an extension of actual GUI elements. Skin color model is used for gesture segmentation, significant hand points (palm, fingers, fingertips) for gesture recognition. Designed interface is used for simple picture explorer controlling. Final application is capable of using simple gestures for scaling, rotation, pointing and click.

## **Klíčová slova**

rozpoznávání gest, OpenCV, počítačové vidění, ovládání počítače gesty, webkamera

## **Keywords**

gesture recognition, OpenCV, computer Vision, gesture-based communication human-computer interaction, webcam

## **Citace**

Jan Chaloupka: Ovládání počítače gesty webkamerou, bakalářská práce, Brno, FIT VUT v Brně, 2011

## **Prohlášení**

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

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem při vypracování čerpal.

.....

Jan Chaloupka

15. května 2011

## **Poděkování**

Tímto si dovoluji poděkovat vedoucímu bakalářské práce panu Ing., Michalovi Španělovi, Ph.D. za cenné rady, připomínky a čas věnovaný při zpracování této práce.

© Jan Chaloupka, 2011.

*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.....	1
2. Současný stav v oblasti rozpoznávání gest.....	2
2.1 Kinect .....	2
2.2 Virtuální realita.....	3
2.3 Gesta v každodenním světě .....	3
2.4 Gestikulární rozhraní.....	4
3. Zpracování obrazu .....	7
3.1 Kalibrace kamery .....	7
3.2 Homografie.....	8
3.3 Barevný model HSV .....	9
4. Návrh řešení rozpoznávání gest a prohlížeče fotek.....	10
4.1 Cíle práce.....	10
4.2 Diagram systému .....	10
4.3 Detekce plochy .....	10
4.4 Detekce dlaně a rozpoznání gesta.....	13
5. Implementace rozpoznávání gest a prohlížeče fotek.....	20
5.1 Koncept rozhraní .....	20
5.2 Hierarchie tříd.....	20
5.3 Třída CalibrateCamera .....	21
5.4 Třída DetectDesktop.....	22
5.5 Třída GestureRecognition .....	23
5.6 Třídy SettingsDialog, Demo, imageviewer .....	26
5.7 Třída ExplorerManager .....	26
5.8 Grafické uživatelské rozhraní.....	27
5.9 Použité vývojové prostředky .....	28
Shrnutí .....	28
6. Návrh testování a vyhodnocení rozpoznávání gest .....	30

6.1	Detekce plochy .....	30
6.2	Rozpoznávání gest.....	30
6.3	Testování uživatelem.....	31
	Shrnutí .....	33
7.	Závěr.....	34
	Literatura .....	35
	Seznam obrázků .....	36
	Dodatek A .....	38
	Obsah CD .....	38
	Příloha .....	39

# Kapitola 1

## Úvod

Počítač je ve vyspělé civilizaci v dnešní době samozřejmostí. Ať už se jedná o automatizovaný proces výroby, regulace, přenos televizního vysílání nebo zpracování obrazů z vesmírných teleskopů. Člověk se s ním dostává do kontaktu denně a stále více jej používá. Mnohdy k němu přistupuje jako k živému: „proč nefunguješ?“, „myslel jsem, že sis uložil můj dokument, kam jsi ho jenom schoval?“, apod. Můžeme mluvit o komunikaci člověka se strojem? Počítači je zadán příkaz zobrazit obrázek, poslat email, přehrát video, atd. Ovšem tento proces je stále neohrabaný, člověk je nucen „mluvit“ za pomoci klávesnice, myši, popř. jiného hardwaru. V porovnání s lidskou komunikací, dva bavící se lidé používají gestikulaci, zvuky, posunky, řeč těla. S vývojem technologií se postupně objevují nové způsoby kontaktu s počítačem. Rozpoznávání řeči, rozpoznávání obrazu, na tom všem se pilně pracuje a hledají se metody, které by umožnili integrovat „typické“ lidské komunikační projevy.

Má práce je zaměřena právě na oblast gestikulace. Řada lidí přede mnou se touto oblastí zajímala a dnes jsou již vidět úchvatné výsledky jejich práce. Příkladem může být Kinect od firmy Microsoft. Otázkou je však, jaký dopad to bude mít na myšlení a vztah lidí k počítačům? To se dozvíme v průběhu let.

Cílem mé práce bylo seznámit se s problematikou detekce lidských gest a navrhnout a implementovat jednoduchou aplikaci demonstrující ovládání počítače gesty. Rozhodl jsem se pro vytvoření prohlížeče fotek umožňující jejich snadné prohlížení sadou čtyř gest.

Druhá kapitola je úvodem do aktuálních technologií a problematiky rozpoznávání gest. Třetí kapitola podává teoretický podklad. Čtvrtá kapitola se zabývá návrhem systému pro rozpoznávání gest, kde jsou uvedeny použité postupy pro detekci a klasifikaci. Obsahem páté kapitoly je implementace, ve které jsou popsány problémy, se kterými bylo nutné se vypořádat. Čtenář tak uvidí, že se nejedná o nic mystického, pouze o jednoduchou matematiku. Předposlední kapitola obsahuje návrh testování a vyhodnocení. Poslední je závěr, ve kterém je shrnuta celá práce a její budoucí pokračování.

## Kapitola 2

### Současný stav v oblasti rozpoznávání gest

Rozpoznávání gest je oblast z počítačových věd a jazykových technologií s cílem interpretovat lidská gesta skrze matematické algoritmy. Gesta mohou pocházet s libovolného pohybu nebo stavu těla, ale obvykle pochází z tváře nebo ruky. Aktuální soustředění v poli zahrnuje rozpoznávání emocí z tváře a rozpoznávání gest. Mnoho přiblížení bylo uskutečněno za pomoci kamery a algoritmů počítačového vidění s cílem interpretovat znakovou řeč. Nicméně, identifikace a rozpoznávání postavení, držení těla, proxemie<sup>1</sup> a lidské činnosti jsou také subjektem technik rozpoznávání gest. Přeloženo z [10].

Samotná kapitola pojednává o současných technologiích, kdy je představen Kinect a více známá virtuální realita. V další části je obsažen úvod do problematiky gest, jejich rozdělení a zmínění se o použitých metodách.

#### 2.1 Kinect

Kinect pro Xbox 360 nebo jednoduše Kinect je „hrací a zábavná novinka“ od Microsoftu pro videoherní platformu Xbox 360. Je založena na připojení webkamery ke stávajícím perifériím Xbox konsole, umožňuje uživateli kontrolovat a interagovat s Xboxem bez potřeby kontaktu s herním ovladačem prostřednictvím přirozeného uživatelského rozhraní použitím gest a verbálních příkazů. Technologie interpretuje informace o 3D scéně ze spojitě vyzařovaného infračerveného světla. Kinect senzor je horizontální krabička připojena na podstavec umožňující změnu polohy, která rozpoznává pohyb, tvář a hlas. Hráči tak interagují hlasem a pohybem rukou a těla. Tak jako populární Wii od Nintendo, lidé používají své tělo k interakci, avšak nepoužívají žádné držáky nebo bezdrátové ovladače. Kinect tak dramaticky rozvíjí koncept herní interakce. Čerpáno a přeloženo z [11], [12].

---

<sup>1</sup> <http://en.wikipedia.org/wiki/Proxemics>





Obrázek 2.1: Ovládání her bez použití herního ovladače (zdroj [12])

## 2.2 Virtuální realita

Virtuální realitu (VR) nazýváme počítačově simulované prostředí, které simuluje přítomnost v reálném nebo i imaginárním světě. Nejvíce dnešních VR jsou primárně vizuální záležitosti, ať na počítačové obrazovce nebo skrze speciální stereoskopické displeje, avšak některé zahrnují i další informativní senzory, jako mikrofon nebo sluchátka. Některé vyspělejší typy zahrnují zpětnou vazbu (tlak, vibrace, ...). VR navíc poskytuje dálkové komunikační prostředí, které poskytuje virtuální přítomnost uživatelům s koncepty teleprezence a teleexistence či virtuální artefakty skrze použití standardních vstupních zařízení jako myš či klávesnice nebo prostřednictvím multimodálních zařízení jako rukavice. Více v [8].

## 2.3 Gesta v každodenním světě

Pokud se oprostíme od světa počítačů a pouvažujeme na chvíli o lidské interakci, uvědomíme si, že používáme v komunikaci široké spektrum gest. I když se gesta ve velké míře liší jak v kontextu, tak i kultuře, jsou úzce spojena s komunikací. Je to vidět například tehdy, pokud lidé gestikulují, zatímco volají, a nemohou jeden druhého vidět tváří v tvář.

Gesta mohou existovat samostatně nebo ve spojení s vnějšími předměty. Bez jakéhokoliv předmětu se kroutíme, kýveme, oddalujeme a ve větší či menší míře (v závislosti na tréninku) používáme více formální znakovou řeč. Vzhledem k objektům, máme k dispozici široký okruh gest, která jsou téměř universální, zahrnující ukazování na předmět, dotýkání se nebo manipulaci s věcmi, změnu tvaru, aktivaci objektů jako ovladače. Z toho se dá usoudit rozdělení gest podle jejich funkce. Cadoz (1994) klasifikuje gesta do tří skupin:

- sémiotická: pro vyjádření smysluplné informace
- ergotická: k manipulaci s reálným světem a vytváření artefaktů
- epistemická: k učení se z okolního prostředí skrze dotykové a hmatové zkoumání

Dnešní interakce s počítačem je převážně negestikulační. Dominantním dogmatem je přímá manipulace, avšak zamysleme se nad tím, jak přímá tato manipulace je. Stačí se jen podívat na to, jak

efektivně lidé používají každodenně gesta a kolik z těchto gest lze použít v interakci s počítačem a jak dlouho trvá, než se je počítač naučí. I ty nejpokročilejší rozhraní typicky podporují pouze symbolická a deiktická rozpoznávání gest.

## 2.4 Gestikulární rozhraní

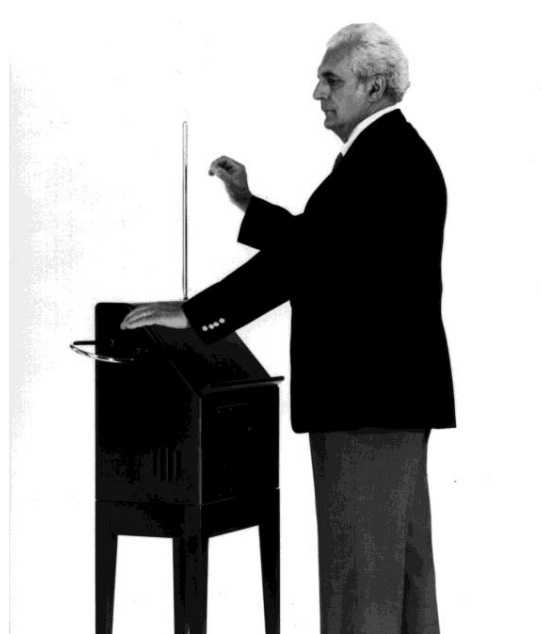
Ekvivalentem těchto rozhraní k přímé manipulaci s počítačem jsou taková, která používají pouze gesta. Ty se mohou různit od zařízení, která rozpoznávají pár symbolických gest až po ta, která plně podporují znakovou řeč. Podobně rozhraní mohou rozpoznat statické držení ruky nebo její pohyb, popř. kombinaci obou. V každém případě každé gesto má jednoznačnou sémantiku, která může být použita.

### 2.4.1 Sledovací zařízení

Rozhraní se syntaxí mnoha gest typicky vyžadují přesné sledování držení ruky. Obvyklou metodou je opatřit ruku rukavicí, která je vybavena počtem senzorů poskytující informace o její poloze, orientaci a ohybu prstů. Příkladem je Cyberglove nebo Dataglove používající tenké optické vlákna sbíhající se na zadní straně každé ruky, každé s malou prasklinkou v sobě. Světlo je těmito kabely vyzařováno a při ohybu prstu je těmito prasklinkami uvolněno. Tento úbytek je potom změřen a dává velmi přesně postavení ruky.



Obrázek 2.2: Rukavice CyberGlove pro přenos informací pro rozpoznání gesta (zdroj [9])



Obrázek 2.3: Zařízení Theramin pro ovládání zvuku (zdroj [9])

Jakmile je postavení ruky rukavicemi zaznamenáno, může být gesto rozpoznáno použitím řady různých technik. Neuronové sítě (NN) nebo porovnání statistických šablon jsou obecně používané praktiky k identifikaci statického držení ruky, často dosahující přesnosti větší než 95% (Väänänen a Böhm 1993). Časově závislé NN mohou být také použity k rozpoznání dynamických gest, i když

častějším přístupem je použití skrytých markovových modelů (HMM – Hidden Markov Models). HMM mohou být také použity pro rozdělení zaznamenávaných držení ruk na individuální gesta a k provedení online učení nových gest (Lee 1996). V tomto případě jsou gesta typicky rozpoznávána použitými předtrénovanými šablonami, avšak rukavice mohou být použity také k identifikaci přirozených nebo nenatrénovaných dat. Jedním z dalších metod je použití přístupu od shora dolů na základě zakřivení prstů a orientace ruky a časového rozdělení pro vytvoření snímků popisujících celé gesto (Wexelblat 1995). Tyto snímky pak mohou být zpracovány na vyšší úrovni pro další interpretaci.

I když technika za pomoci rukavic dává velmi přesné výsledky, je drahá a zatěžující. Další možností je použití technik počítačového vidění (CVT – computer vision techniques). Ty jsou přirozenější, než výše zmíněná technika, avšak mají i svá omezení (Sturman and Zeltzer 1994):

- rozlišení kamery je příliš nízké pro kvalitní nalezení prstů a pokrytí celého pohledu zahrnující široký pohyb ruky
- 30 nebo 60 snímků za sekundu není dostatečný pro zaznamenání náhlých pohybů
- Prsty se těžko sledují, jelikož jsou zakryty jeden druhým nebo rukou.

Existují dva různé přístupy k rozpoznávání gest založených na CVT: techniky založené na modelu ruky, které se snaží vytvořit trojdimenzionální reprezentaci ruky (např. špičky prstů a spoje kloubů k vytvoření válcovitého modelu ruky (Rehg and Kanade 1994)) a techniky založené na extrakci příznaků přímo z obrázku (nalezení ruky segmentací tónů kůže z pozadí a následně extrakcí konečků prstů, rohů ruky nebo hrubých částí).

#### **2.4.2 Zařízení pro rozpoznání přirozených gest**

Efektivní gestikulární rozhraní reagující na přirozená gesta, zvláště na dynamické pohyby ruky, mohou být vyvinuta v té nejjednodušší formě. Raným příkladem je Theramin, elektronický hudební nástroj z 20. let 19. století. Prostřednictvím dvou senzorů měřící délku (vertikální a horizontální) reaguje na pohyb ruky. Vertikální senzor kontroluje výšky, horizontální hlasitost. Theramin je úspěšný díky tomu, že přímo mapuje pohyby ruky na spojitou odezvu, umožňuje tak uživateli rychle vytvořit mentální model, jak zařízení používat.

Videoplace Myrona Kruegera je dalším systémem reagujícím na přirozená gesta (Krueger 1991). Používá real-time zpracování obrazů z živého videa uživatele. Pro vytvoření siluety uživatele a identifikaci relevantních příznaků je použito odečítání pozadí a detekce hran. Samotné rozpoznávání je natolik dokonalé, že rozliší mezi rukou a prsty, jestli jsou prsty natažené nebo skrčené nebo i to, o který prst se jedná. S těmito schopnostmi byl systém naprogramován k vykonání řady interakcí, z nichž mnohé blízce odpovídaly každodennímu používání gest.



Obrázek 2.4: Rozpoznání ukazování, špička prstu použita jako kreslicí zařízení (zdroj [9])

### 2.4.3 Rozpoznávání symbolických gest

VideoPlace a Theramin reagují na přirozená gesta, avšak rozhraní s širším spektrem příkazů mohou vyžadovat symbolická gesta. V tomto případě jsou určité příkazy spojeny s přetřenoaným tvarem gesta. Tyto rozhraní jsou obvykle použita v imersivních<sup>2</sup> virtuálních prostředích, ve kterých uživatel není schopen vidět v reálném světě na vstupní zařízení. V tomto případě je sada předtřenoaných gest použitých pro navigaci skrze toto prostředí a interakci s virtuálními objekty. Například ve virtuálním rozhraní Rubber Rocks je uživatel schopen zvednout virtuální kameny dlaňovým gestem a hodit a uvolnit je dalším gestem (Codella 1992). Symbolická gesta mohou mít spoustu výhod (Baudel a Beaudouin-Lafon, 1993):

- přirozená interakce: gesta jsou přirozená forma interakce a jednoduše se používají
- stručné a silné: jedno gesto může být použito pro specifikaci příkazu i jeho parametrů
- přímá interakce: ruka jako vstupní zařízení eliminuje potřebu prostředníků

Ovšem je zde i řada nevýhod. Uživatele může unavit provádění gest v prostoru a samotné rozhraní není samopopisné, uživatel je nucen se naučit řadu gest, kterým systém rozumí. S rostoucím počtem gest je pak problém s jejich zapamatováním, a další problémy.

Vše čerpáno a přeloženo z [9].

---

<sup>2</sup> [http://en.wikipedia.org/wiki/Immersion\\_\(virtual\\_reality\)](http://en.wikipedia.org/wiki/Immersion_(virtual_reality))

## Kapitola 3

### Zpracování obrazu

Účelem této části je podat teoretický základ pro návrh systému. Obsahem je popis kamery, její nedokonalosti a způsob korekce. Dále co je to homografie, jež je použita pro přepočtení souřadnic a nakonec barevný prostor HSV.

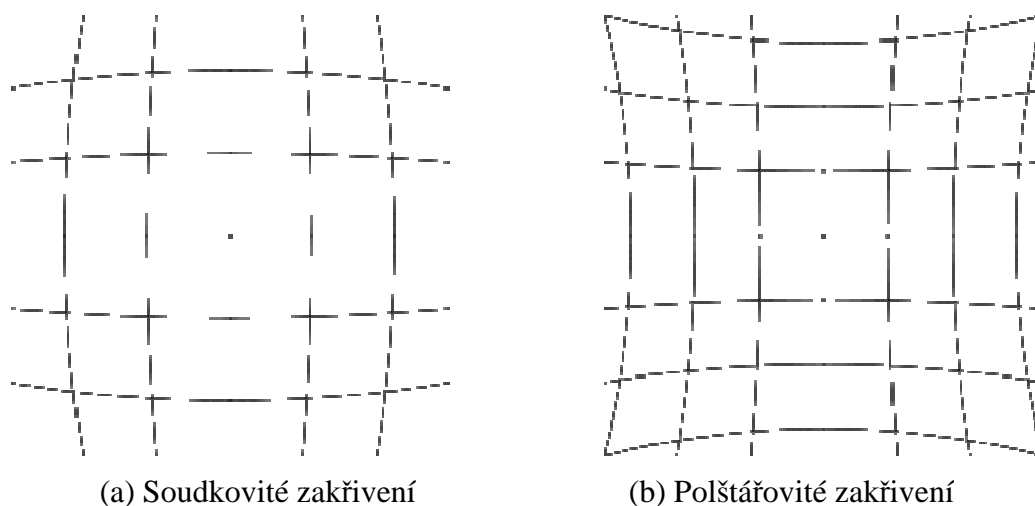
#### 3.1 Kalibrace kamery

Jedná se o nezbytnou součást snímající jednotlivé gesta, která jsou posléze zpracována a klasifikována. Její kvalita ovlivňuje výsledné zpracování. Pro správnou činnost je potřeba se vypořádat s klasickými problémy (např. zakřivení, popsáno níže).

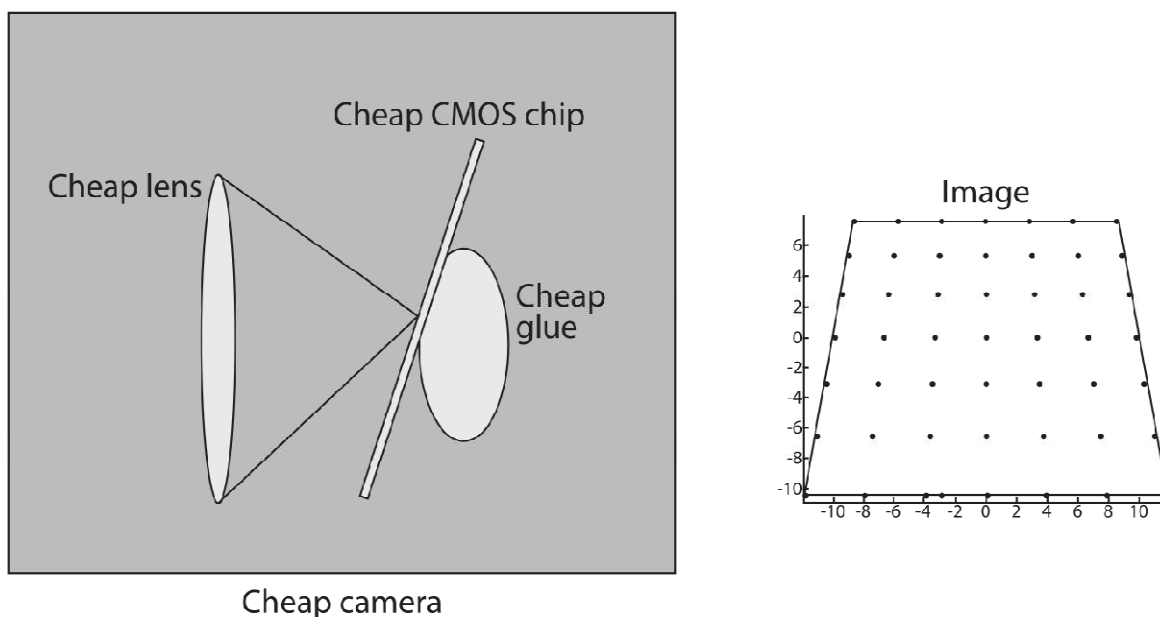
##### 3.1.1 Zkreslení obrazu

Teoreticky je možné definovat čočku, která nemá žádné zkreslení. V praxi není ovšem žádná čočka ideální. Hlavní příčinou je výrobní proces; je mnohem jednodušší vytvořit „sférickou“ čočku, než vyrobit více matematicky ideální „parabolickou“ čočku. Také je náročné mechanicky přesně zarovnat čočku a snímač (přeloženo z [1]). Rozlišujeme dva typy zakřivení, radiální (dáno tvarem čočky) a tečné (výrobním procesem).

Radiální zakřivení můžeme rozdělit nejčastěji na soudkovité (barrel) a polštářovité (pincushion). Viz obr. 3.1(a) a 3.1(b). U soudkovité jsou body vzdáleny více od středu, než ve skutečnosti. U polštářovitého je tomu naopak.



Obrázek 3.1: Radiální zakřivení (převzato z [2])



Obrázek 3.2: Tečné zakřivení (převzato z [1])

Tečné zakřivení (obr. 3.2) vzniká v případě, kdy čočka není úplně paralelní k promítací ploše; v levných kamerách je toto chování způsobeno v případě, kdy je snímač přilepen zezadu kamery (přeloženo z [1]).

K opravě těchto „vad“ (kalibraci) lze využít několika speciálních funkcí knihovny OpenCV. Nejprve je potřeba spočítat dvě matice: intrinsics a distortion (matice parametrů). K tomu je potřeba použít metodu *cvFindChessboardCorners*, které „předhodíme“ několik snímků šachovnice (různě natočená jak do stran, tak i do vzdálenosti). Její výstup je dále zpracován a pomocí *cvCalibrateCamera2* jsou spočítány dané matice. Pro korekci snímku je následně použita metoda *cvInitUndistortMap* pro výpočet transformace, voláním *cvRemap* následně získáme opravený obraz. Více o zakřivení čočky a opravě obrazu je možné se dozvědět z [1], [2].

### 3.2 Homografie

Homografie je invertibilní transformace z reálné projekční plochy<sup>3</sup> na projekční plochu, která mapuje přímé úsečky na přímé úsečky. Synonyma jsou kolineace, projektivní transformace a projekce (přeloženo z [3]). Nezachovává velikosti a úhly, zachovává incidenci<sup>4</sup> útvarů a dvojpoměr<sup>5</sup>. Více o homografii např. v [3].

<sup>3</sup> Více informací na [http://en.wikipedia.org/wiki/Real\\_projective\\_plane](http://en.wikipedia.org/wiki/Real_projective_plane)

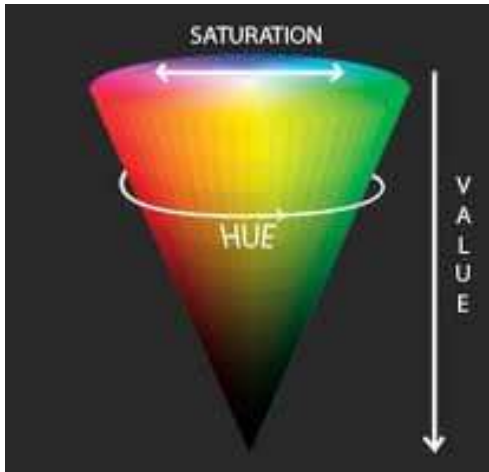
<sup>4</sup> Více na [http://en.wikipedia.org/wiki/Incidence\\_\(geometry\)](http://en.wikipedia.org/wiki/Incidence_(geometry))

<sup>5</sup> Více na <http://en.wikipedia.org/wiki/Cross-ratio>

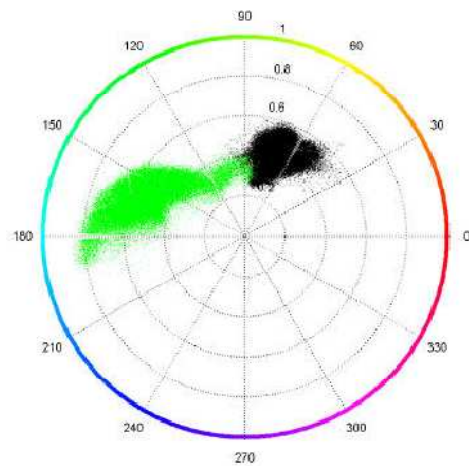
### 3.3 Barevný model HSV

Definuje typ barevného prostoru. Podobný RGB a CMYK modelům. Má tři základní komponenty: odstín (hue), nasycení (saturation) a světelnost (value), obr. 3.3. Odstín reprezentuje barvu (odpovídá úhlu), nasycení rozsah šedé v barevném prostoru, světelnost (množství světla). Více ve [4].

Pro účely práce je důležitá spojitost modelu s barvou lidské kůže. Ta, jak je vidět na obr. 3.4, tvoří ohraničenou oblast, kterou lze od okolních barev oddělit určením rozsahu hodnot jednotlivých barevných složek modelu.



Obrázek 3.3: HSV model (zdroj [4])



Obrázek 3.4: Segmentace barvy lidské kůže (zdroj [6])

## Kapitola 4

### Návrh řešení rozpoznávání gest a prohlížeče fotek

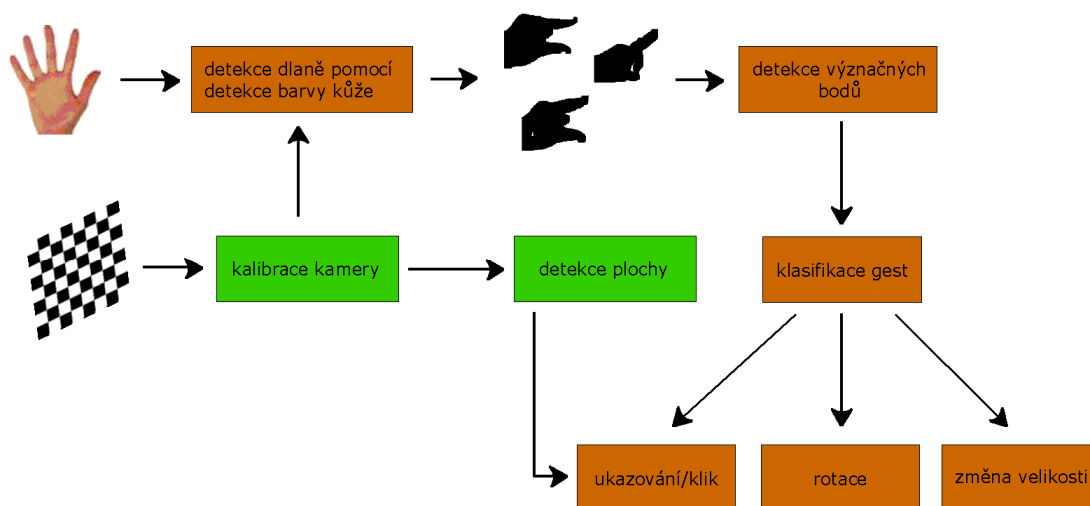
V této části jsou popsány použité metody, omezení a postupy nalezení ruky, její následné klasifikace (rozpoznání gesta) a použití v aplikaci. Na začátku jsou nastíněny cíle, ve kterých je popsáno, čím se práce zabývá. Dále jak pomocí webkamery detekovat plochu. Seznámení s jednoduchou metodou detekce dlaně (zpracování obrazové informace základními operacemi) a její rozpoznání (použití primitivní metody založené na vzájemné poloze význačných bodů). A nakonec použité knihovny, které ulehčují vývoj aplikace a návrh uživatelského rozhraní.

#### 4.1 Cíle práce

Záměrem je vytvořit jednoduchý prohlížeč fotek, na kterém je demonstrováno ovládání počítače gesty ruky, která jsou snímána prostřednictvím externí webkamery. Nejedná se o nahrazení stávajících periférií (myš, klávesnice, touchpad, aj.), nýbrž o jejich rozšíření.

Z důvodu použité metody pro klasifikaci (popsáno dále) se rozpoznávají gesta pro rotaci (doleva, doprava), změnu velikosti (zvětšení, zmenšení) fotky a její výběr (ukazování + klik). Uživatel je tedy schopen si fotku přiblížit/vzdálit, otočit či zobrazit jinou podle potřeby.

#### 4.2 Diagram systému



Obrázek 4.1: Diagram systému, přehled jednotlivých kroků

#### 4.3 Detekce plochy

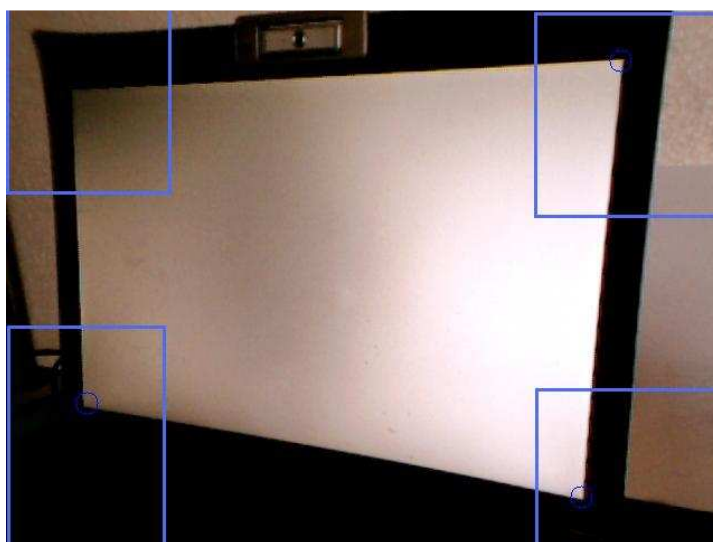
První metoda detekce plochy byla založena na rozdělení snímku na regiony, ve kterých se nachází dané rohy (obr. 4.2 a), b)). Jejich rozměry a souřadnice byly zvoleny tak, aby vyhovovali pozici kamery zprava (vyvstává omezení pro uživatele). Každý tento region byl následně zpracován převedením na stupně šedi a provedením morfologických operací (několikrát dilatace a eroze). V takto



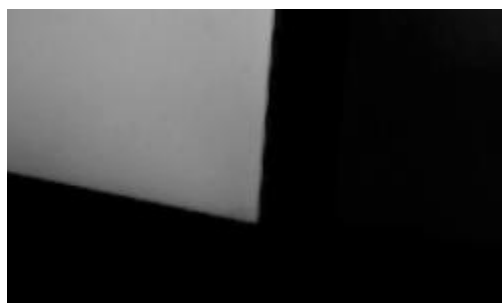
zpracovaném regionu se našly rohy. Mohlo se stát, že daný roh neodpovídal rohu plochy (odstavec níže), proto byla ještě provedena kontrola, jestli se opravdu o něj jedná (vzal se výřez jeho okolí o čtvercovém poloměru 20 pixelů, naprahoval se, našly se další body a podle jejich vzájemné polohy se kontrolovalo, obr. 4.2 c)).

Často se stávalo, že byl detekován jiný bod, než roh (čočka kamery, výstupky na obalu monitoru/obrazovce). Volbou jiného osvětlení nastával stejný případ. Řešením bylo volit po předem daných intervalech jiné metody prahování a docílit tak adaptibilní detekce (adaptibilní prahování). Tento postup ale nepřinesl žádné zlepšující výsledky, naopak došlo ke zhoršení. Jelikož plocha obsažená ve snímku zabírala téměř celý rozsah stupňů šedi, nešlo prahování použít.

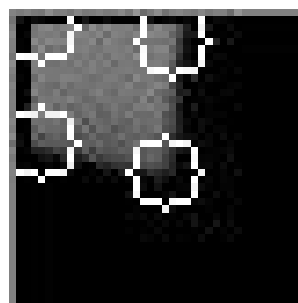
Nevýhodou tohoto postupu bylo, že jakmile se ruka přiblížila příliš blízko levému okraji plochy, stávalo se nemožné detekovat dlaň (byla mimo záběr), která je nutná ke klasifikaci. Nebylo tedy možné využít celou plochu a uživatel byl omezen na pravou část. Navíc byl problém při umísťování kamery, kdy stačilo, aby daný roh nebyl ve vymezeném regionu a detekce selhala. Řešení bylo fixování na umístění kamery zprava, nešlo zvolit jinou polohu. I osvětlení ovlivňovalo správnou detekci (příliš velký jas způsobil nalezení jiných pozic.) Proto jsem se rozhodl pro jinou metodu.



Obrázek 4.2 a): Detekce rohů plochy



Obrázek 4.2 b): Výřez rohu



Obrázek 4.2 c): Kontrola rohu

Druhá metoda je založena na odečítání dvou snímků plochy, kdy na každém z nich má plocha jinou barvu. Prvotním pokusem byly zvoleny barvy červená, modrá a zelená, tedy tři snímky (obr. 4.3 a, b),

c)). Předpokládal jsem, že tyto barvy budou mít na získaném snímku přibližně stejnou hodnotu. Potom by stačilo použít prahování pro každou barevnou složku. Ukázalo se však, že tomu tak není díky nerovnoměrnému osvětlení (na obrázcích 4.3 lze vidět přechod barvy od tmavší ke světlejší). Nepomohla ani změna pozice kamery, popř. jiné světelné podmínky.



Obrázek 4.3 a): červená plocha



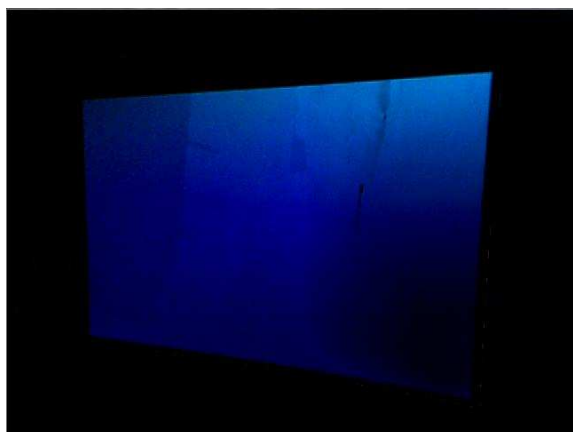
Obrázek 4.3 b): zelená plocha



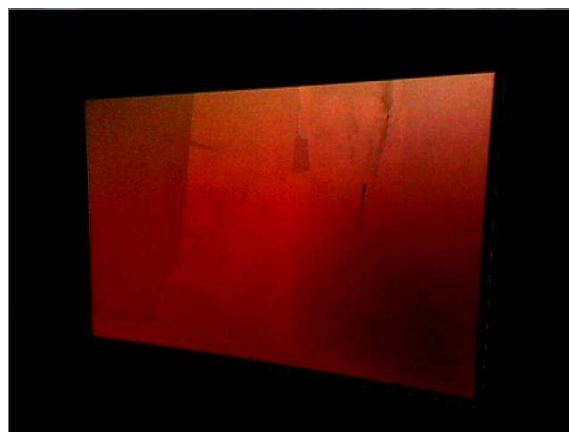
Obrázek 4.3 c): modrá plocha

Proto jsem se inspiroval metodou založenou na odečítání pozadí, kdy nepohybující nebo neměnní body mají stejnou hodnotu a jejich vzájemným odečtením se vyruší. V mém případě je konstantní pozadí a mění se barva na ploše. Odečtením jednotlivých snímků potom dostaneme samostatnou plochu (obr. 4.4 a), b), c)). Na obrázcích 4.4 a), b) jsou vidět tmavší oblasti u pravého spodního rohu. Opět dáno daným osvětlením. Na obrázku 4.4 c) je ale vidět, že se zde tato oblast nenachází. Dává nejlepší výsledky z daných rozdílů a navíc se barva tolik nemění, proto byl zvolen pro další zpracování. Dalším krokem je prahování (práh zvolen na hodnotu 10), kdy dostaneme obrys plochy, jak je vidět na obr. 4.5. V tomto stavu je hrana pilovitá, proto je provedeno několikrát rozmazání pro odstranění výrazných přechodů. Následně se detekují rohy pomocí vhodných metod.

Rozdílem oproti první metodě (rozdělení na regiony) je možnost libovolné pozice plochy na snímku. Co se týče vzdálenosti, je zde větší variabilita, ovšem je potřeba dávat pozor na přílišné oddálení od plochy, kdy z důvodu chybné detekce je zvolena minimální vzdálenost (200 pixelů) dvou libovolně detekovaných rohů.



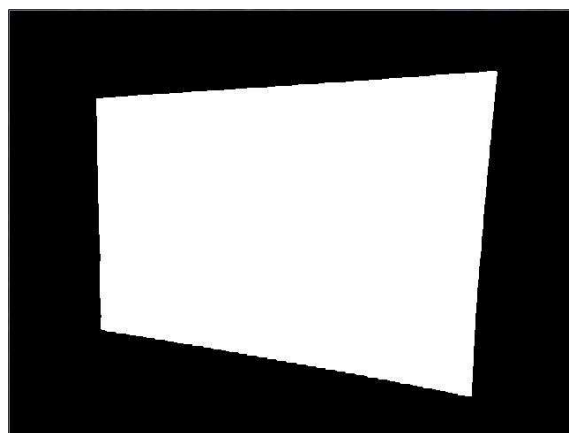
Obrázek 4.4 a): Odečtení modré a zelené



Obrázek 4.4 b): Odečtení červené a zelené



Obrázek 4.4 c): Odečtení červené a modré



Obrázek 4.5: Prahování rozdílu č. a m. barvy

## 4.4 Detekce dlaně a rozpoznání gesta

Existuje řada způsobů, jak nalézt ruku, např. Haarovy příznaky, HOG (histogram of oriented gradients), SIFT, SURF. Zvolil jsem detekci založenou na barevném modelu HSV, jelikož je to metoda jednoduchá na implementaci a např. knihovna OpenCV poskytuje potřebné vybavení pro její realizaci. Vyjmenované možnosti jsou časově náročné či je potřeba natrénovat vhodné klasifikátory (např. Haarovy příznaky), avšak mohou dávat lepší výsledky. Dále rozeberu postup, který jsem použil.

### 4.4.1 Použití barevného modelu HSV pro detekci dlaně

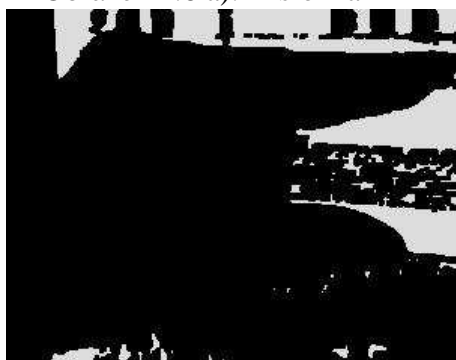
Experimentováním se ukázalo, že postačující hodnoty pro detekci obdélníku ohraničující ruku leží v intervalu:  $H <0, 20>$ ,  $S <30, 150>$ ,  $V <80, 255>$ . Jejich zvětšení, popř. zmenšení vede k detekci i jiných objektů, než ruky. Vybráním těchto intervalů v jednotlivých barevných složkách získáme oblast, na které provedeme sekundární segmentaci. Ta je založena na principu zpracování každé složky zvlášť, kdy jsou provedeny např. morfologické operace či rozmazání pro odstranění šumu, konečným prahováním získáme bitmapu, obr. 4.6(a-c). Nad takto vzniklými bitmapami provedeme logický průnik a další zpracování (opět morfologické operace), čímž dostaneme segmentované gesto pro následné rozpoznání, obr. 4.6(d).



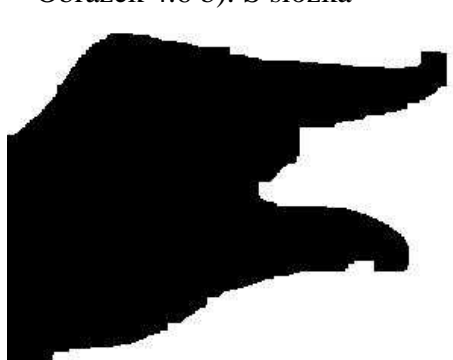
Obrázek 4.6 a): H složka



Obrázek 4.6 b): S složka



Obrázek 4.6 c): V složka



Obrázek 4.6 d): log. součin složek

Nevýhodou tohoto přístupu je závislost složek na osvětlení. Nedostatečné či přílišné způsobuje useknutí či deformaci částí ruky, které jsou potřebné pro správné rozpoznání (např. konečků prstů).

#### 4.4.2 Statická gesta

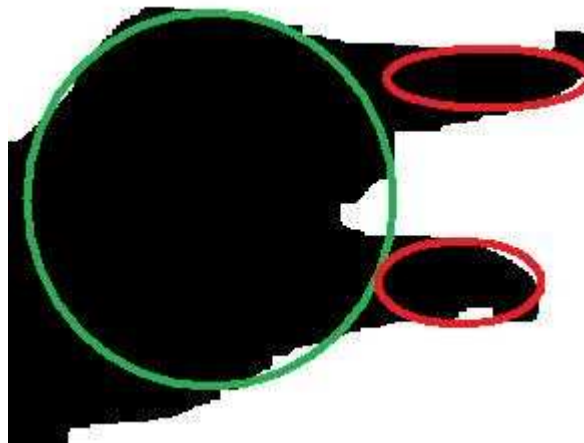
Jednou z možností je použít n-gramy<sup>6</sup>, kdy dané gesto znormalizujeme na předem danou velikost a porovnáme se vzory (např. rozdělením na mřížku čtverců a určení rozdílu od vzoru). Hlavním kritériem pro výběr rozpoznávací metody byla potřeba invariance vůči vzdálenosti prstů mezi sebou (gesto pro změnu velikosti, rotace prstů). Např. rotace může být u každého jednotlivce vyjádřena jinou vzdáleností palce a ukazováčku. Proto jsem se zaměřil na vzájemnou pozici prstů, podle které se rozpoznává. Zvolené prsty je možné nahradit jinými, pokud splňují požadované kritéria. Avšak musí být dosti zřetelné, který prst je natažen a který ne. Zde vlastnosti, kterými se od sebe jednotlivé gesta liší:

- **změna velikosti:** dva prsty; x-ová souřadnice konce ukazováčku a palce se od sebe liší minimálně (zvolená konstanta). Vzdálenost pak určuje, jestli se jedná o zmenšení nebo zvětšení.
- **rotace:** dva prsty, x-ové souřadnice konce ukazováčku a palce se od sebe liší více, než předem zvolená konstanta. Podle znaménka difference x-ových souřadnic se rozlišuje rotace vlevo nebo rotace vpravo.

<sup>6</sup> <http://www.springerlink.com/content/054045227701341q/>

- **ukazování:** jeden prst.

Jedná se tedy o metodu klasifikace založenou na detekci význačných bodů a určení jejich polohy vzhledem ke středu dlaně. Princip hledání těchto bodů je založen na rozdělení ruky na prsty a dlaň. Prsty můžeme definovat jako útvary velice odlišné od pravidelného  $n$ -úhelníku (kde  $n$  je velké číslo, lze nahradit elipsou), dlaň jako útvar velmi podobný tomuto  $n$ -úhelníku (lze nahradit kružnicí), viz. Obr 4.7.



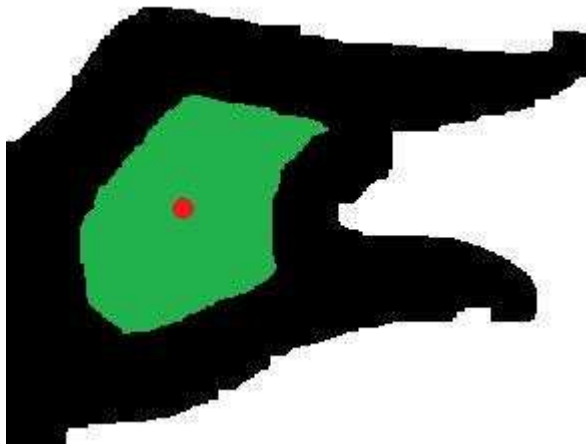
Obrázek 4.7: Podobnost dlaně, resp. prstů s kružnicí, resp. s elipsou

Za význačné body lze zvolit např. střed dlaně, středy prstů a jejich koncečky. Jako další lze zvolit body přechodu mezi dlaní a prsty (jejich detekce je problematická), popř. body na obvodu dlaně. Prvotním pokusem bylo detekovat body tvořící obrys ruky, rozdělit je na skupinky tvořící dlaň, resp. prsty a provést jejich aproximaci kružnicí, resp. přímkou metodou nejmenších čtverců<sup>7</sup>. Ovšem tento směr byl zavrhnut, jelikož výpočet aproximace je časově náročný (lze nahradit výběrem několika náhodných bodů dané skupiny a spočítáním průměru středu a poloměru, resp. průměrné směrnice). Ukázalo se jednodušší nalézt konturu kolem dané skupiny a určit střed, z toho důvodu se rozpoznávají první tři výše uvedené význačné body.

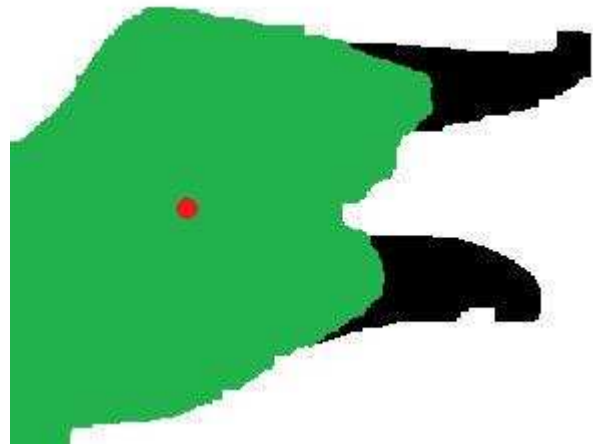
Nalezení samotné dlaně spočívá v eliminaci prstů. Dále je potřeba znát efekt operace eroze a hodnoty pixelů, na které působí. „Eroze obecně zmenšuje velikost objektů a odstraňuje malé anomálie odečtením objektů s poloměrem menším, než strukturní element. Na šedotónových obrázcích redukuje světlost světlých objektů na tmavém pozadí odebráním minima okolí při průchodu strukturního elementu obrázkem“, citováno z [7]. Efektem eroze je tedy ořezávání pixelů s hodnotou 0 na okrajích ruky (dále také úzkých části či výstupky, jelikož jejich tloušťka je malá). Toho je využito právě pro eliminaci prstů, kdy několikanásobnou aplikací postupně ořezáváme prsty, dokud nezmizí (počet erozí byl experimentálně určen na 60), jak je vidět na obr. 4.8 a). Při volbě většího počtu erozí dochází ke

<sup>7</sup> [http://en.wikipedia.org/wiki/Least\\_squares](http://en.wikipedia.org/wiki/Least_squares)

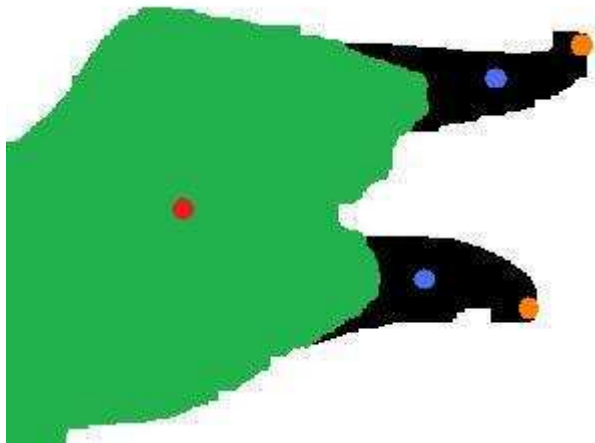
ztrátě dlaně (případy špatně osvětlené ruky, popř. příliš vzdálené od kamery), při menším počtu se objevují malé objekty (šum z okolí).



Obrázek 4.8 a): ořezání prstů ruky, zůstává ořezaná dlaně (zelená barva), červená tečka značí střed



Obrázek 4.8 b): odečtení dilatované dlaně od segmentované ruky, černě zbylé prsty



Obrázek 4.8 c): středy prstů označeny modře, konečky oranžově

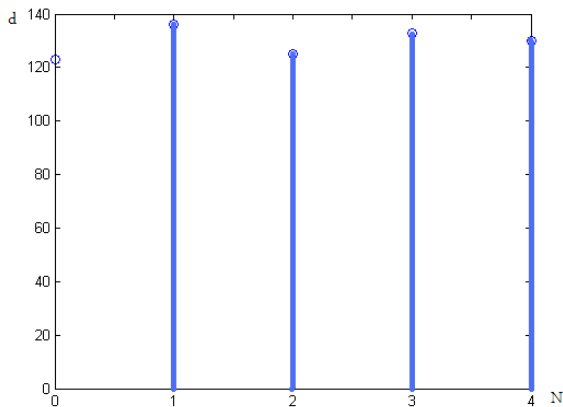
Protože jedno z rozpoznávaných gest je ukazování místa na ploše prohlížeče, je třeba určit rohy monitoru, resp. obrazovky (pro určení relativní polohy ukazovaného místa) a polohu místa samotného (souřadnice pozice odpovídají souřadnicím na snímku, proto je nutné jej převést na souřadnice plochy). Pro jeho přepočítání je použita planární homografie, kdy je vytvořena matice transformace a její aplikací na každou takto nalezenou polohu ve snímku.

#### 4.4.3 Dynamická gesta

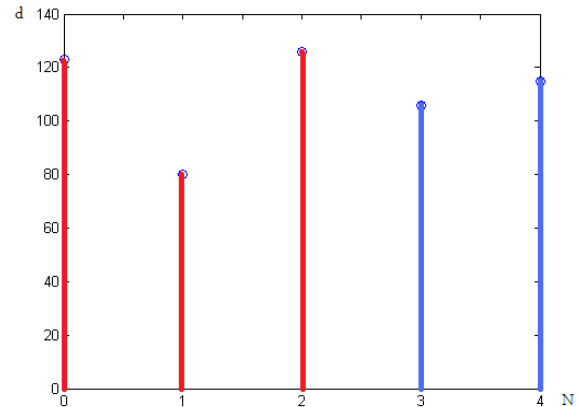
Zde se jedná pouze o rozpoznání kliku. Původní koncept byl zaznamenávat si posledních  $n$  (např. 5) pozic ukazovaného místa (souřadnice středu a konečku prstu). Experimentováním se ukázalo, že pokud je pohyb při kliku dostatečně rychlý, konec prstu začíná být pro kameru průhledný, dochází tedy ke zmenšení vzdálenosti mezi středem a konečkem prstu. Postupem tedy bylo vypočítat si tuto vzdálenost. Tato metoda byla ale náchylná k chybným detekcím, kdy stačila prudká změna pozice



konečku. Proto jsem se zaměřil na změnu vzdálenosti y-ových souřadnic těchto dvou bodů. Na obr. 4.9 a) a 4.9 b) lze vidět grafy vzdáleností pro ukazování a klik.



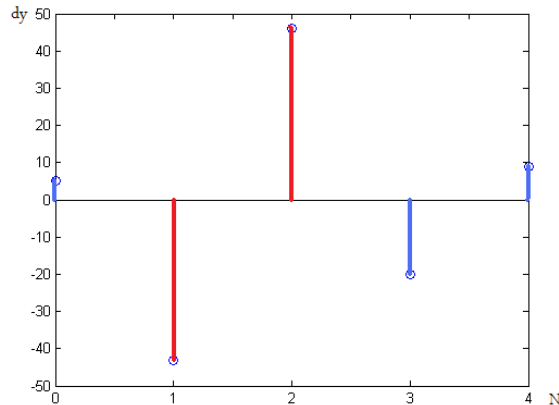
Obrázek 4.9 a): vzdálenosti středu a koncečku prstu při plynulém pohybu



Obrázek 4.9 b): vzdálenosti středu a koncečku prstu při kliku

Ukazování lze specifikovat plynulým pohybem nebo minimálním rozdílem vzdálenosti (obr. 4.9 a)). Naopak pro klik je specifická náhlá změna (obr. 4.9 b)). Je potřeba si přitom dávat pozor, jestli je tato změna způsobena rychlým svislým pohybem nebo opravdu klikem. Avšak v obou těchto případech nelze přesně určit, o který se jedná, jelikož není dostatek potřebných informací (bylo by potřeba zvolit jinou metodu detekce dlaně). Tento nedostatek lze omezit volbou meze maximální a minimální diference. Samotná diference pro klik je znázorněna na obrázku 4.9 c). Ten můžeme definovat jako posloupnost dvou diferencí, kde první je záporná (pohyb prstu dolů) a druhá kladná (pohyb prstu nahoru).

Jelikož je klik součástí ukazování pozice, dochází k její změně. Vystává tím problém určit, na které místo je klik aplikován. Je možné brát pozice, ve které byla detekována kladná či záporná diference. Ukázalo se však, že i toto řešení je nedostatečné. Proto jsem zvolil výběr pozice, která následuje za ukončením kliku a vytvořením kruhu kolem této pozice, který definuje oblast, ve které se hledá příjemce kliku (více v implementační části).



Obrázek 4.9 c): Diference vzdáleností při kliku

#### 4.4.4 Stabilizátory náhlé změny gesta, pozice ukazovaného místa

Daná metoda rozpoznání je závislá na kvalitě segmentace ruky a ta je závislá na daném osvětlení. Při špatných světelných podmínkách dochází k chybné klasifikaci, a proto je nutné odstranit náhle změny gesta (např. změna velikosti přejde na ukazování vlivem špatného osvětlení palce). Stabilizace je založena na ukládání historie posledních gest. Ta gesta, která se vyskytují izolovaně nebo tvoří posloupnost předem zvolené délky, jsou nahrazena posledním rozpoznávaným stabilním gestem. Tím však dochází jakoby k zamrznutí (může nastat případ, kdy máme za sebou několik totožných rozpoznání). Proto je potřeba volit maximální počet odlišných gest po sobě s opatrností.

Pro ukazování je analogický problém. Jeho zdrojem je drobné chvění ruky, proto se provádí průměrování několika posledních pozic.

Samotnou stabilizaci vysvětlím na příkladě. V tabulce níže lze vidět sekvenci rozpoznávaných gest (0 žádné gesto, 1 ukazování, 2 škálování, 3 rotace):

původní sekvence	stabilizovaná sekvence	
0	0	uložení prvního gesta jako stabilní (0)
0	0	aktuální gesto stejné jako stabilní (0), žádná změna
2	0	aktuální gesto různé od stabilního, nahradíme stabilním, první změna
2	0	aktuální gesto různé, nahradíme stabilním, druhá změna
2	2	aktuální gesto různé, stabilní nahradíme aktuálním (2)
2	2	aktuální gesto stejné jako stabilní (2)
2	2	aktuální gesto stejné jako stabilní (2)
2	2	aktuální gesto stejné jako stabilní (2)
3	2	aktuální gesto různé od stabilního, nahradíme stabilním, první změna
2	2	aktuální gesto stejné jako stabilní (2)
2	2	aktuální gesto stejné jako stabilní (2)
<b>3</b>	2	aktuální gesto různé od stabilního, nahradíme stabilním, první změna
<b>1</b>	2	aktuální gesto různé, nahradíme stabilním, druhá změna
1	1	aktuální gesto různé, stabilní nahradíme aktuálním (1)
1	1	aktuální gesto stejné jako stabilní (1)
1	1	aktuální gesto stejné jako stabilní (1)
0	1	...
0	1	



Principem je ukládat si poslední stabilní gesto (na začátku zvoleno jako první, tím je nejčastěji žádné gesto). Jakmile narazíme na jiné gesto, než poslední stabilní, nahradíme jej stabilním a zaznamenáme si, že došlo k nahrazení. S následujícím gestem děláme totéž. Pokud jsme takto nahradili dvě gesta a stále jsme nenarazily na stabilní, nahradíme stabilní aktuálním (uživatel vykonává jiné gesto). Tímto způsobem odfiltrujeme všechny náhlé změny způsobené špatně klasifikovaných gestem. Nevýhodou je právě zamrznutí (modré gesta v tabulce), kdy se opakuje totéž gesto třikrát po sobě. Při malé hodnotě maximálního počtu různých gest od stabilního je tento stav zanedbatelný, jakmile je ale tato hranice příliš vysoká, ovládnutí se začíná „sekat“.

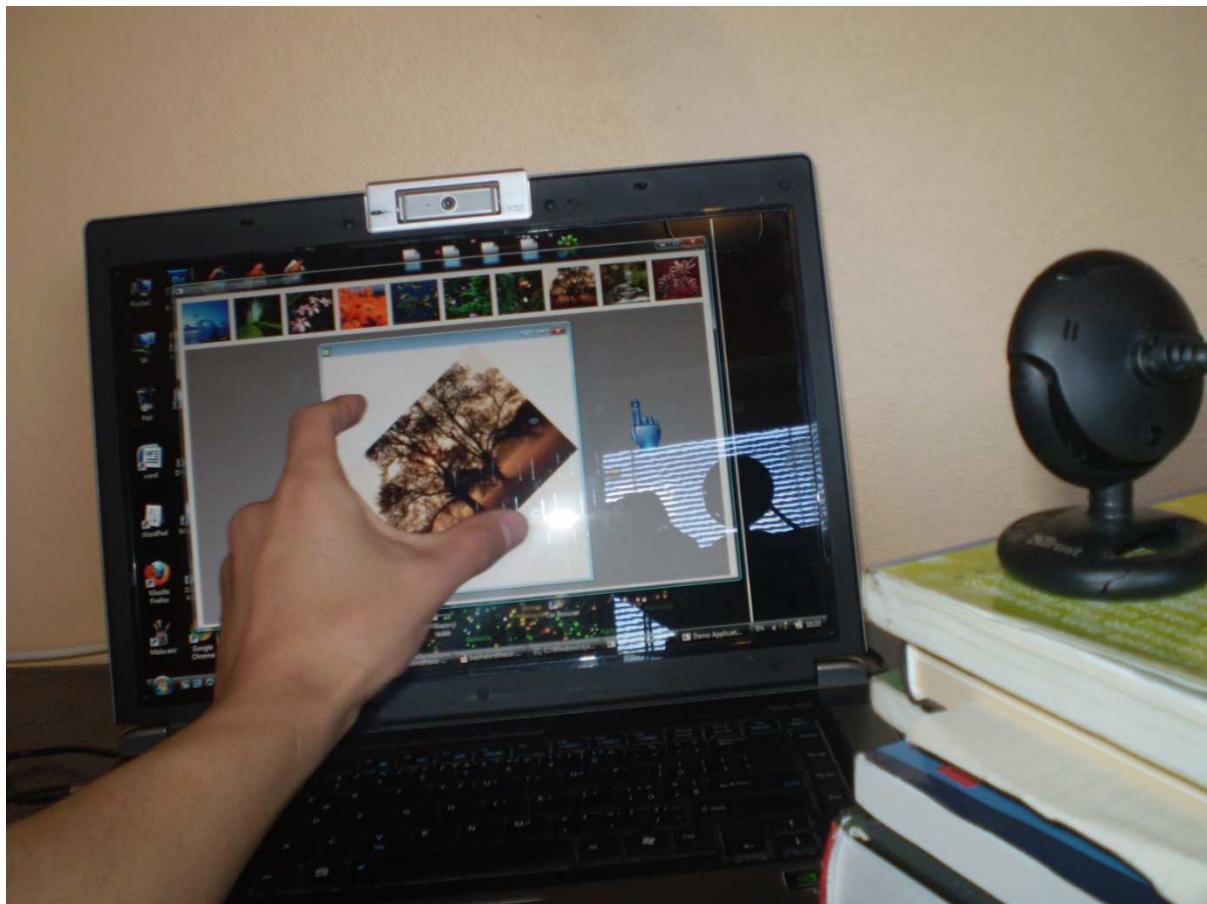
## Kapitola 5

### Implementace rozpoznávání gest a prohlížeče fotek

Na základě návrhu jsou vytvořeny jednotlivé třídy starající se o kalibraci kamery, detekci plochy, rozpoznání gesta a třídy uživatelského rozhraní obstarávající propojení se signály a vzájemnou komunikaci mezi sebou. Kalibrace kamery využívá standardních metod knihovny OpenCV, detekce plochy poskytuje informaci o umístění, samotné rozpoznávání gest pak tvoří největší část kapitoly.

#### 5.1 Koncept rozhraní

Formu komunikace uživatele s rozhraním lze vidět na obrázku 5.1. Kamera snímá ruku a podle rozpoznávaného gesta provádí operace (na obrázku konkrétně rotace obrázku doleva).

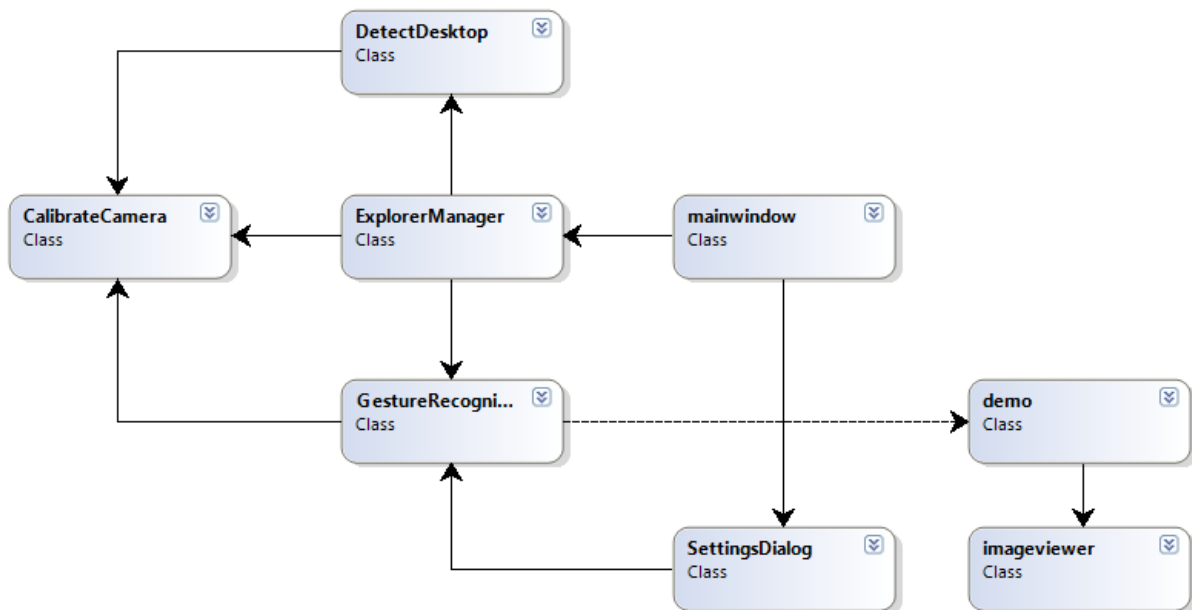


Obrázek 5.1: Koncept rozhraní

#### 5.2 Hierarchie tříd

Na obrázku 5.2 lze vidět hierarchii tříd systému. *ExplorerManager* se stará o správu detekce plochy, kalibrace a rozpoznávání gest. *CalibrateCamera* poskytuje parametry pro úpravu obrazu při detekci plochy a rozpoznávání gest. Třída *mainwindow* volá metody *ExplorerManageru*, spouští dialog pro změnu nastavení aplikace (parametry gest pro třídu *Gesture Recognition*) a spouští demo

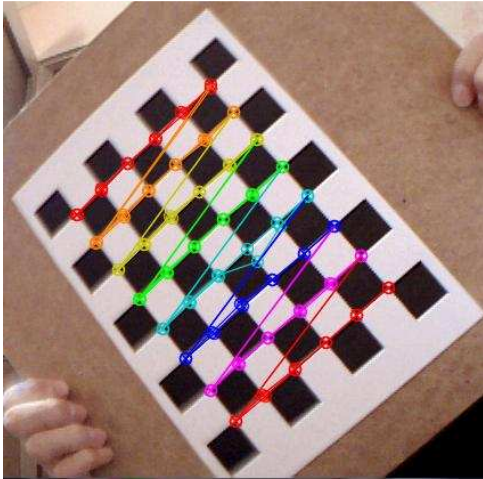
aplikaci. Třída *demo* sestává z několika objektů třídy *imageviewer* a jsou ji zasílána rozpoznaná gesta (přerušovaná čára).



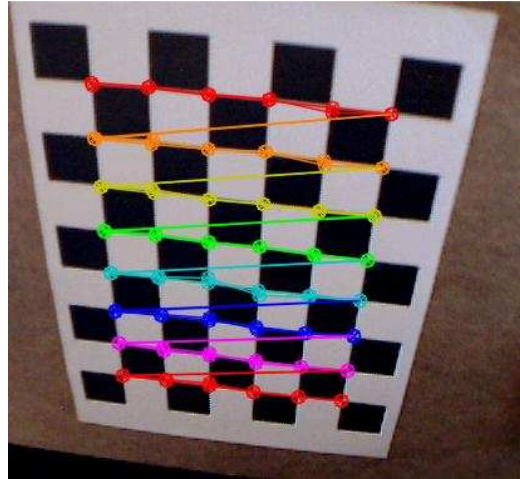
Obrázek 5.1: Hierarchie tříd

### 5.3 Třída CalibrateCamera

Jejím účelem je vytvoření matic pro opravu obrazu (distortion, intrinsics). Jak bylo zmíněno v návrhu, knihovna OpenCV je pro tyto účely vybavena sadou metod. První implementace byla založena na vykreslování šachovnice o rozměrech 8x6 na plochu obrazovky a její otáčení. V daném časovém intervalu se nasnímal šachovnice a voláním *cvFindChessboardCorners* se našly její body. Kamera byla umístěna z boku a její pozice ovlivňovala výsledné vypočítané matice. Velmi často se stávalo, že obraz se ještě více zakřivil. Důvodem byl nedostatek různých souřadnic bodů. Opravou bylo zavedení snímání externí šachovnice, obr. 5.3. V tomto případě je možné provádět i jiné transformace, než otáčení. Různou vzdálenost hran šachovnice se získávají další souřadnice.



Obrázek 5.3(a): Detekce bodů



Obrázek 5.3(b): Změna vzdálenosti hran

Těchto snímků je pořízeno celkem 8, přičemž je brána každá 4. úspěšně detekovaná šachovnice (decimace). Uživatel tak nemusí rychle měnit polohy. Vypočítané souřadnice jsou pro každý snímek uloženy do matice. Ty se předají do metody `cvCalibrateCamera2`, jejíž výstup jsou již požadované matice parametrů pro opravu zakřivení. Uloží se do XML soubory, který je použit dalšími třídami pro opravu obrazu.

## 5.4 Třída `DetectDesktop`

Slouží pro detekování rohů obrazovky. Napřed jsou z XML souborů načteny matice pro úpravu obrazu. Před samotnou detekcí je časový úsek, ve kterém se kamera „zahřívá“, tj. adaptuje na dané osvětlení (v případě zobrazení červené nebo modré barvy se stalo, že výsledné snímky obsahovaly nízké osvětlení, proto je potřeba na začátku zobrazit zelenou, která obsahuje více jasu). Poté dochází k snímání nejdříve červené, modré a nakonec zelené plochy. Pro tyto změny je vytvořena v Qt šablona pomocí `QCreator`, přičemž pak měníme jen barvu daného layoutu. Interval mezi jednotlivými změnami je důležitý. Při rychlém snímání kamera nemusí stíhat snímat všechny snímky, při pomalém je uživatel nucen dlouho čekat, proto jsem zvolil 300 ms intervaly.

Po získání všech tří snímků dochází k odečítání červené a modré barvy. Původně se odečítala navíc ještě červená od zelené a modrá od zelené. Ukázalo se však, že tyto kroky nejsou potřebné. Plocha je i tak detekována. Zelená se dále nepoužívá, ale zůstává pro potřeby, kdyby první odečtení nestačilo a bylo nutné doimplementovat více detekcí. Získaný obraz je dále naprahován s mezí 10, čímž dostaneme vysegmentovanou plochu, erodován, dilatován a rozmazán (pro otupění hran, jak je uvedeno v návrhu). Rohy nadetkovány pomocí metody `cvGoodFeaturesToTrack` a seřazeny.

Se samotnou detekcí se získá také rozlišení plochy. Jelikož pro zobrazení barvy na ploše se používá layout v režimu plná obrazovka, stačí zjistit jeho rozměry a tím získáme i rozměry plochy. Původně se zjišťovali voláním metody knihovny `windows.h`, což ale omezovalo použití pouze na platformy `windows`.

V konečné fázi se všechny detekované rohy zobrazí uživateli, ten potom může upravit pozici plochy ve snímku nebo opakovat detekci, pokud nastaly chyby (špatně detekované rohy, špatná pozice plochy, ...).

## 5.5 Třída `GestureRecognition`

Zde je jádro samotného rozpoznávání. Proces můžeme rozdělit do několika fází:

1. V první fázi je úprava obrazu před samotnou detekcí.
2. Druhá fáze je zaměřena na nalezení obdélníku ohraničující ruku, ve kterém provádíme detekci částí ruky. Tato fáze nám umožňuje zmenšit plochu hledání a tím zrychlit detekci.
3. Ve třetí fázi upřesňujeme intervaly složek a hledáme jednotlivé části ruky. Výstupem je seznam bodů popisující ruku.
4. Čtvrtá fáze je typická klasifikací jednotlivých gest na základě polohy bodů získaných ve třetí fázi. Součástí je eliminace bodů, které znepřesňují klasifikaci. Výstupem pro každé gesto je jeho typ, popř. jeho pozice.
5. Předposlední fází je stabilizace klasifikovaných gest, kdy nahrazujeme osamocená gesta okolními gesty a tím odstraňujeme náhlé změny v ovládní. Součástí je historie posledních gest.
6. V poslední fázi je pro každé gesto vyvolána daná akce, kterou přijme posluchač (demo) a na jejím základě provede danou operaci.

Jako ve třídě `DetectDesktop`, dochází k načtení parametrů pro opravu obrazu, na jejichž základě upravíme zakřivení snímku aplikací transformačních matic. Snímek převedeme do HSV barevného prostoru, pro nalezení obdélníku nejdříve provedeme vybrání pásem pro každou složku (danými intervaly). Tím dostaneme bitmapu, ve které se nachází ruka, popř. další objekty, které leží ve stejných intervalech. Abychom odfiltrovali šum a jiné nečistoty (např. malé částečky), provedeme morfologické operace s následným vyhlazením a prahováním. Tím dostaneme binární reprezentaci ruky.

Obdélník nalezneme pomocí kontur (`OpenCV` je vybavena metodou `cvFindContours`). Vyhledáme kontury všech útvarů a provedeme selekci na základě jejich rozměrů a obsahů ploch (objekty s malou plochou nebo podobné čtverci lze odfiltrovat). Zůstane nám ruka, kolem které určíme obdélník (metoda `cvBoundingRect`), jehož souřadnice a rozměry použijeme pro výřez části snímku obsahující námi hledané gesto.

Ukazuje se, že pokud bychom provedli detekci na takto získaný snímek, mohlo by se stát, že části ruky budou chybět (experimentálně ověřeno), jinými slovy získané gesto neobsahuje dostatek informací pro kvalitní klasifikaci, proto se provádí sekundární segmentace (viz. návrh). Každá složka je zpracována zvlášť a poté proveden logický průnik podle vzorce:

$$diff = |H - S - V| \quad (1)$$

$$\Pi(H, S, V) = \begin{cases} 0, & diff = 0 \\ 255, & diff \neq 0 \end{cases} \quad (2)$$

**Vysvětlení vzorce:** Segmentovaná ruka je v každé složce reprezentována hodnotou 0. Ostatní složky hodnotou různou od nuly tak, že každá složka má svoje specifické číslo (zvoleno tak, aby rozdíl dvou nenulových hodnot libovolných složek nevyšel roven 0). Jejich odečtením dostaneme nulu nebo nenulovou hodnotu. Nulová potom reprezentuje ruku, nenulová ostatní části.

Tímto postupem dochází k malé ztrátě informace o gestu, jelikož každá složka obsahuje mírně odlišnou segmentaci ruky a průnikem dojde k vybrání jen těch společných částí. Proto se každá tato složka zpracovává odlišně s cílem co nejvíce omezit tuto ztrátu. Největším problémem je odstínová složka, která je náchylná na změny osvětlení, navíc kopečky či hrbolky na povrchu ruky na straně zdroje osvětlení jsou jinak osvětleny a při segmentaci se ztrácí, proto je speciálně pro tento případ přidán další interval segmentace a takto získaná část snímku je přičtena k odstínové složce. Nehty nejsou výjimkou (lesknou se a jejich hodnoty neleží v žádném rozumném intervalu složky jasu). Nenalezl jsem způsob, jak je zahrnout do ruky, takže v případě velkého osvětlení nejsou vidět, s čímž dochází k problémům při detekci palce, ve výsledku potom dochází k častější špatné klasifikace gesta pro rotaci a škálování (s tímto problémem se částečně vypořádávají stabilizátory gest, viz níže). Na získaný logický průnik provedeno další zpracování pro vyhlazení a odstranění šumu (viz obrázky 4.6).

V další fázi začíná hledání význačných bodů. Podle návrhu nalezneme dlaň (erodujeme segmentovanou část ruky, čímž odstraníme prsty). Na takto získaný útvar aplikujeme metodu *cvGoodFeaturesToTrack*, čímž získáme 8 bodů ležících na obvodu. Samotný střed pak jako průměr těchto bodů:

$$S = \frac{1}{N} \sum_{i=1}^N B_i \quad (3)$$

kde  $B_i$  jsou souřadnice  $[x, y]$  bodů na obvodu,  $S$  počítaný střed

Nevýhodou takového průměrování je náchylnost na posun tohoto středu (pohybem ruky se mění tvar erodované dlaně a tím body na obvodu, takže dostáváme pokaždé jiný střed). Mírným potlačení tohoto chování je výběr většího počtu bodů, avšak tento způsob se neukazuje příliš účinný, proto jsem zůstal u volby 8 bodů. Jiným řešením by bylo použití váženého průměrování:

$$S = \frac{1}{N} \sum_{i=1}^N \omega_i B_i \quad (4)$$

kde  $B_i$  jsou souřadnice  $[x, y]$  bodů na obvodu,  $S$  počítaný střed,  $\omega_i$  váha pro bod  $B_i$ . Touto cestou jsem se však nevydal, jelikož určení daných vah (experimentálně, popř. heuristika) vyžaduje kvalitnější segmentaci ruky, informace o orientaci prstů a heuristiky.

Jakmile ze segmentované ruky známe část obsahující dlaň, nalezení prstů je již snadná záležitost. Stačí odečíst dlaň, kterou nejdříve upravíme na původní velikost (obr. 4.3 b)). Prsty potom nalezneme

pomocí kontur, přičemž odstraňujeme objekty, které mají malý obsah. Detekce význačných bodů jednotlivého prstu se skládá ze dvou kroků (obr 4.3 c)):

1. nalezení středů
2. nalezení konečků prstů.

Za koneček prstu je považován bod na obvodu, který je nejdál. K určení vzdálenosti je použita Manhattanova vzdálenost dána vztahem:

$$d(A, B) = |A_x - B_x| + |A_y - B_y| \quad (5)$$

Nejvzdálenější bod potom určíme na základě vztahu:

$$F_i = \max_j [d(C_i, B_{ij})] \quad (6)$$

kde  $F_i$  je koneček prstu  $i$ ,  $C_i$  střed prstu  $i$ ,  $B_{ij}$  body tvořící hranici prstu  $i$ . Kontura nám poskytuje velký počet bodů a nemá smysl počítat vzdálenost pro každý z nich, proto se vybírá každý  $n$ -tý.

Problém přichází, pokud jsou dva prsty příliš blízko u sebe. V tomto případě dochází k detekci více prstů, které téměř splývají. Význačných bodů je více a komplikují klasifikace. Pro odstranění je použita kontrola na úhel svíraný mezi těmito prsty. Protože dopředu neznáme, o které se jedná, kontroluje každý s každým a v případě, že svíraný úhel je menší, než minimum, vybere se ten prst, který byl dříve detekován. Úhel spočítáme pomocí směrnice přímků vytvořených ze středu a konečku prstů podle vztahu, kdy bereme absolutní hodnotu tohoto úhlu:

$$\tan\varphi = \frac{k_2 - k_1}{1 - k_2 k_1} \quad (7)$$

Vlivem šumu se také objevují malé objekty nalevo dole od středu dlaně, proto dochází k jejich odfiltrování kontrolou jejich souřadnic. Obdélník kolem ruky rozdělíme na 3x3 šachovnici. Pokud se detekovaný prst (jeho střed) nachází v nejlevějším nejspodnějším políčku, je zahozen. Tento postup si můžeme dovolit v případě, kdy žádné z detekovaných gest nemá v tomto políčku význačné body.

Máme střed dlaně, středy a konečky prstů reprezentující dané gesto. Můžeme tedy pokračovat ke klasifikaci.

Klasifikace gesta je, jak již několikrát zmíněno, založena na vzájemné poloze bodů. Jako základní bod je zvolen střed dlaně (dále počátek). Konkrétní gesto je určeno podle kritérií již dříve zmíněných. To je označeno jednoznačným číslem. U ukazovacího se navíc ukládá jeho pozice. Tato čísla se ukládají do fronty a tvoří historii. Na začátku rozpoznávání je fronta prázdná, takže dokud se nenaplní, neprovádí se kontrola historie. Jakmile je fronta naplněna, začíná se procházet prvek po prvku a kontrolují se s předchozími položkami. Každé gesto je uloženo do vlastní proměnné a slouží jako nahrazení místo nestabilního.

Ovládání aplikace je potom řízené právě stabilními gesty. Ty volají signály, které jsou napojeny na dané sloty ovládající rotaci, změnu velikosti, ukazování a klik. Ukazovaný bod je ještě potřeba přepočítat na souřadnice na ploše. K tomu slouží homografie podle rovnic:

$$X = \frac{xh_0 + yh_1 + h_2}{xh_6 + yh_7 + h_8}, Y = \frac{xh_3 + yh_4 + h_5}{xh_6 + yh_7 + h_8} \quad (8)$$

Kde  $H = \begin{pmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{pmatrix}$  je transformační matice homografie,  $X, Y$  jsou přepočítané souřadnice,  $x, y$  původní souřadnice.

Detekce gesta klik je založena na zjišťování diferencí svislých vzdáleností mezi středem a konečkem prstu. Z tohoto důvodu uchováváme do fronty tyto vzdálenosti. Jakmile fronta nebude požadované délky (v našem případě 5), dochází k jejímu procházení, kdy se počítají difference. Jelikož se jedná o frontu, nastává pro jeden klik více výskytů (dokud daný klik nevypadne z fronty, v našem případě se 3 krát opakuje). Řešením může být proměnná, která nám bude udávat, nastal-li klik, v kladném případě se po dobu 3 cyklů nebude provádět klik. V aplikaci to ovšem ničemu nevádí, dojde jen k trojnásobnému provedení zobrazení nového obrázku. Z takto získaných diferencí sestavíme pole a hledáme po sobě zápornou a kladnou diferenci. Pokud mají obě minimální diferenci, detekovali jsme klik a můžeme vyvolat signál, která zachytí posluchač a vyvolá patřiční akce. Procházením prvního a druhého prvku v poli zabráníme násobnému kliku. Stále ale může nastat případ, kdy budou po sobě následovat dva kliky (vlivem šumu).

## 5.6 Třídy SettingsDialog, Demo, imageviewer

SettingsDialog umožňuje uživateli měnit parametry gest. Po jejich případné změně dochází k znovu uložení daných parametrů do XML souboru. Ten je vytvořen ihned po spuštění úvodního nastavení v případě, že ještě neexistuje.

Demo tvoří samotný prohlížeč fotek, obsahuje 10 ukázkových fotografií, se kterými může uživatel testovat aplikaci. Každý obrázek je instance třídy imageviewer a obsahuje metody pro změnu velikosti a rotaci (součástí Qt knihovny pro práci s QLabel, rotace složením jednodušších operací).

## 5.7 Třída ExplorerManager

Spravuje v sobě instance tříd pro kalibraci kamery, detekci plochy, rozpoznávání gest a aplikace. Stará se o spojení jednotlivých akcí s danými metodami. Od třídy DetectDesktop přijímá souřadnice rohů a rozlišení plochy a předává je třídě GestureRecognition. Ta komunikuje s třídou Demo a ta s jednotlivými instancemi třídy imageviewer.

Další součástí jsou metody pro přesun parametrů gest z/do GestureRecognition a Demo.

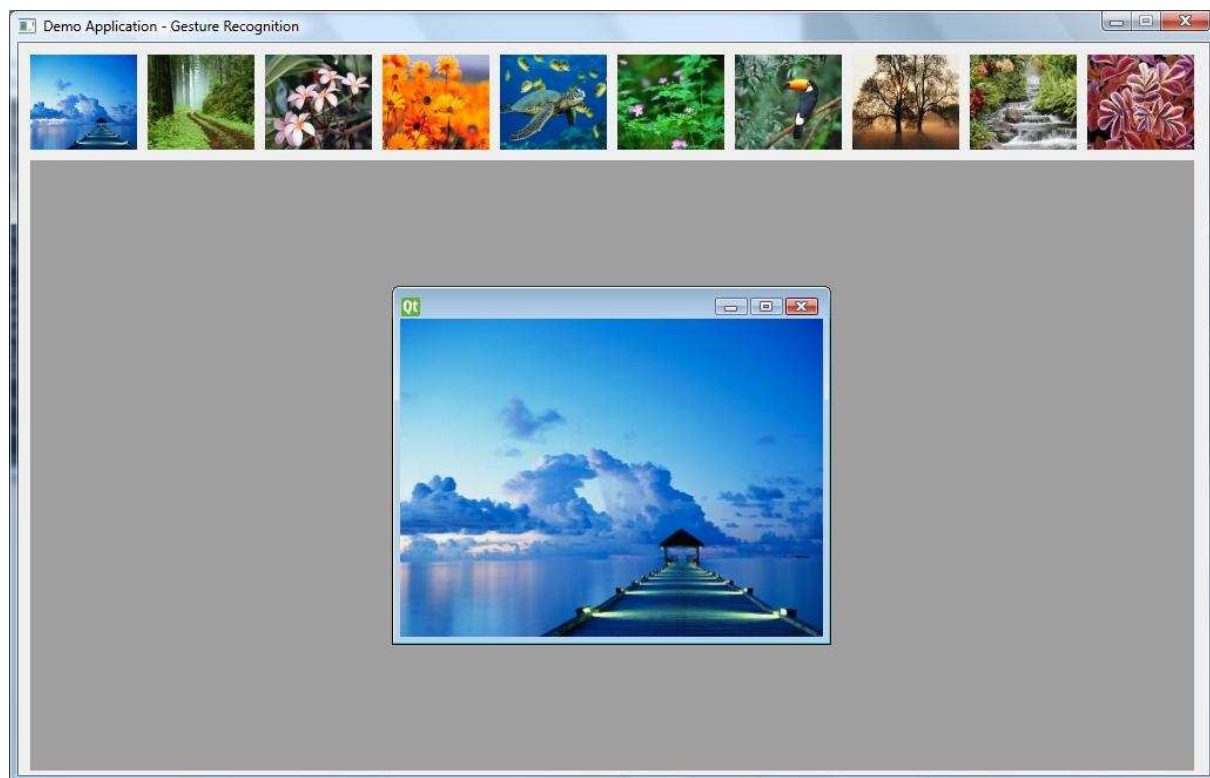


## 5.8 Grafické uživatelské rozhraní

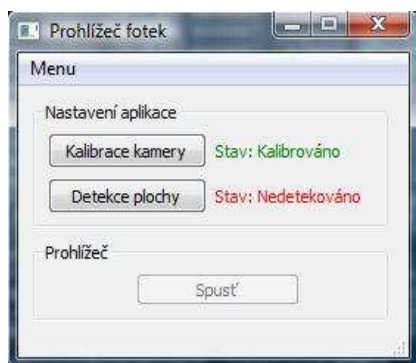
Jádrem celého systému je systémový manažer, který umožňuje kalibraci kamery, detekci plochy, klasifikaci gest a správu uživatelského rozhraní. Před spuštěním samotné aplikace je potřeba nejprve nakalibrovat kameru. Uživatel musí být proto opatřen vytištěnou šachovnicí, kterou ukazuje kameře. Ta si vybere několik snímků a podle nich vypočítá potřebné matice. Výsledek je viditelný ihned po skončení. Pokud se stane, že obraz nemá požadovanou kvalitu, je možné tento proces opakovat. Druhým krokem je detekci rohů plochy na monitoru. Ta závisí na daném osvětlení a dalších aspektech zmíněných výše. Pokud jsou nalezeny, jsou do výsledného snímku zakresleny. Uživatel má možnost provést detekci opakovaně, jedná-li se o chybnou. Poté je možné spustit prohlížeč a začít pracovat.

### 5.8.1 Demo aplikace

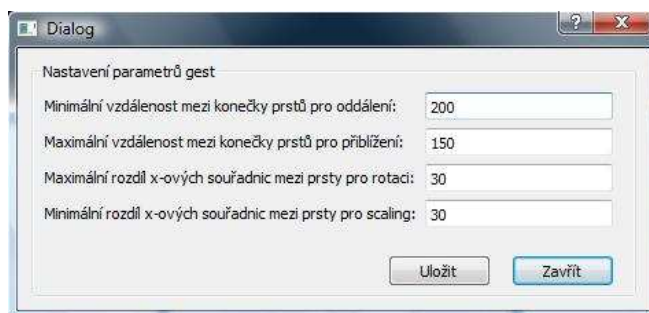
Samotná aplikace sestává ze vstupního rozhraní (obr. 5.4 b), obr. 5.4 c)) a prohlížeče (obr. 5.4 a)). Vstupní rozhraní umožňuje uživateli intuitivně kalibrovat kameru, detekovat plochu a spustit prohlížeč. Stav je potom signalizovaný jak barvou, tak textovou formou pro lepší přehlednost. Nastavení parametrů umožňuje adaptaci na různé velikosti prstů a kontrolu přechodů mezi gesty (ukazování na rotaci a zpět). Rozpoznání gest pak běží na pozadí spuštěné aplikace, ve které je možno výběru z několika fotografií.



Obrázek 5.4 a): demo aplikace



Obrázek 5.4 b): Vstupní rozhraní



Obrázek 5.4 c): Nastavení parametrů gest

## 5.9 Použité vývojové prostředky

### 5.9.1 OpenCV

Pro zpracování obrazových dat je použita open source knihovna OpenCV<sup>8</sup> (Open Source Computer Vision). Obsahuje řadu metod, které umožňují provádět základní obrazové operace (ostření, rozmazávání, morfologické operace, atd.), sledování pohybu objektů, nalezení význačných bodů, čtení videa z webkamery aj. Seznámení se základními metodami a jednoduchými příklady lze např. z knihy [1].

### 5.9.2 Qt

Uživatelské rozhraní je napsáno za pomoci knihovny Qt<sup>9</sup> v C++, která umožňuje např. tvorbu formulářů, dialogových oken, aplikačních oken, aj. Rozpozná gesta pak volají metody/sloty, které vyvolávají signály ovládající danou aplikaci.

## Shrnutí

Implementace podle návrhu byla doprovázena řadou problémů. Jedním z nich byla správná kalibrace, bez které by nebyla možnost rozumně detekovat plochu. V počátcích vývoje jsem používal dva monitory, na jeden byla namířena kamera, na druhém se zobrazovaly její snímky, protože nebylo možné zároveň kalibrovat a zároveň sledovat, na kterých snímcích byla nalezena šachovnice. Problémem bylo, že takto vytvořená kalibrace byla pro uživatele skryta. Ten neviděl, jestli probíhá nebo jestli jenom svítí obrazovka. Přejít na snímání externí šachovnice tento problém vyřešil a uživatel tak může s klidem kontrolovat celý proces.

Dalším problémem bylo získání testovacích dat, které by umožňovaly vytvořit základní klasifikační algoritmus. Nahrál jsem několik videosekvencí pohybu ruky pro polohu kurzoru, zvětšování a rotaci, přičemž se nahrávky děly za příznivých světelných podmínek. Nevýhodou pak byla specializace

<sup>8</sup> <http://opencv.willowgarage.com/wiki/>

<sup>9</sup> <http://qt.nokia.com/products/>

rozpoznávání právě na tyto gesta a dané podmínky. Jakmile jsem se pokusil o reálné snímání gest, aplikace nebyla schopna rozpoznání, jen ve výjimečných případech. Najít správné parametry a meze pro zpracování snímků a klasifikaci si vyžádali řadu testování a ovlivnili tak výslednou implementaci. Jedním z nevyřešených problémů však stále zůstává správně osvětlení, kdy nedostatek světla způsobuje špatnou segmentaci a gesto se tak stává nerozpoznatelné.

Na druhou stranu při kvalitních světelných podmínkách se podařilo implementovat rozpoznání všech gest, které jsem si vytyčil v návrhu. Čas od času dojde k chybnému určení gesta a aplikace pak dělá něco jiného, než uživatel zamýšlí. Navíc co uživatel, to jiný tvar ruky, jiná délka prstů. Proto je možné v omezené míře nastavit vzdálenost mezi konečky prstů, popř. úhel pro natočení pro změnu velikosti a rotaci.

## Kapitola 6

### Návrh testování a vyhodnocení rozpoznávání gest

Testování lze rozdělit do dvou částí. První část je zaměřena na samotnou aplikaci, tj. její schopnost detekovat plochu a rozpoznat gesta za proměnlivých světelných podmínek a různých poloh kamery. Druhá část je zaměřena na vyhodnocení interakci s uživatelem, jak intuitivní je ovládání, jak obtížně se daná gesta provádí, nedochází-li ke zmatení, je-li únavné držet ruku před plochou.

#### 6.1 Detekce plochy

Danou metodou detekce je detekování robustní, jediným kritériem je viditelnost celé plochy v jakémkoliv místě snímku. Omezením je potom minimální poměr obsahu plochy k celkovému obsahu snímku. Zaměřením se na různé pozice kamery (zleva, zprava, shora aj.) zvýšíme uživateli volnost jejího umístění. Součástí testů je tedy několik nahrávek plochy z různých míst. Avšak je možné využití real-time detekce plochy, které je jednodušší nepotřebou videosekvencí.

#### 6.2 Rozpoznávání gest

Pro každé rozpoznávané gesto je potřeba vytvořit desítky nahrávek od různých uživatelů pro otestování, jestli daný systém je schopen jej detekovat. Jednak se zaměřit na pravé detekce a jednak na falešné detekce. Postupným testováním se ukázalo, že kritický je přechod mezi gestem pro změnu velikosti a ukazování, kdy vlivem špatného osvětlení dochází k přebliknutí gest (ukazování se mění na změnu velikost a naopak). Proto je nutné vytvořit sadu testů a vyladit systém tak, aby při této změně nedocházelo k nečekaným změnám ovládání. Dále je potřeba se zaměřit na gesto klikání, které se lehce stává ukazováním, pokud se neprovede náhlá změna pohybu konce ukazováčku. Danou metodou detekce kliku se může stát, že dojde k falešnému detekování, z toho důvodu je i proto dané rozpoznání striktnější. V celku se tedy jedná o záznamy obsahující rotaci, změnu velikosti, ukazování, klikání a dále falešné gesta či destruovaná jako např. ukazování s dvěma spojenými prsty s cílem detekce gesta s jedním prstem, nikoliv dvěma. Ve fázi, kdy je systém schopen rozpoznávat gesta a podle nich provádět příslušné akce se jeví výhodnější provádět real-time testování. Je možné měnit jednak světelné podmínky, jednak mírně polohu kamery (spíše vzdálenost) a různé polohy ruky, které se v testovacích videosekvencích nevyskytly.

Úspěšnost detekce byla testována na pěti sadách, každá pro jedno gesto. Každá sada byla složena z 10 videosekvencí různých uživatelů. Tabulka 6.1 udává jednotlivé úspěšnosti:

gesto	Úspěšnost [%]
rotace vlevo	86,5
rotace vpravo	77,9
zvětšení	85,3
zmenšení	80,5
klik	42,6

Tabulka 6.1: úspěšnost detekce gest

Samotná úspěšnost se počítala podle vzorce:

$$\text{úspěšnost} = \frac{\text{počet správně detekovaných výskytů daného gesta v posloupnosti vzorků}}{\text{celkový počet detekovaných výskytů všech gesta v posloupnosti vzorků}}$$

kde posloupnost vzorků znamená posloupnost detekovaných gest, ve které se provádělo dané gesto. Nízká hodnota gesta klik je způsobena malým rozdílem po sobě jdoucích diferencí (je problém zvolit vhodnou rychlost pohybu konečku prstu), bylo by potřeba jiným způsobem určovat diferenci a zpřesnit určování průběhu změny polohy konečku prstu.

Schopnost správně detekovat gesto je silně ovlivněna světelnými podmínkami. Při ideálním podmínkách (Příloha 1), kdy je dlaň s prsty rovnoměrně osvětlena (nejlépe venkovní denní světlo), se dostávají dobré výsledky. Pokud je šero, stín, málo světla či umělé osvětlení (Příloha 2), výsledky se zhoršují, v horším případě detekce nefunguje vůbec.

### 6.3 Testování uživatelem

Jelikož samotné ovládání gestem není natolik rozšířené v povědomí uživatelů, je potřeba zpočátku uživatele navést, jakým způsobem se provádí ovládání. Je celkem pravděpodobné, že samotný uživatel bude pohybovat rukou před plochou, snažit se ovládat aplikaci a přitom se neuvědomí, že jeho ruka není snímána kamerou. Je to způsobena právě nevědomostí nových přístupů a jejich principů. Důležitou součástí testů je přístup uživatele a jeho schopnost rychle si vytvořit mentální model, jak to vše vlastně funguje. Testuje se tedy intuitivnost a reakce. Z důvodu nedokonalosti rozpoznání, kdy místy dochází před odezvou na gesto k menší časové prodlevě, může být uživatel nedočkavý, popř. jiným způsobem nervózní, čímž se dané ovládání stává obtížné či neočekávané. Proto je nutné zaznamenat chování a reakce uživatele a omezit tyto situace na minimum.

Součástí každého vyhodnocení musí být i seznam věcí, které uživateli chybí, které by změnil, popř. které považuje za zbytečné.

Samotné testování bylo provedeno na 10 lidech, zde některé reakce:

- „Nevěděla jsem, že je něco takového vůbec možné. Ze začátku jsem nevěděla, co mohu a nemohu, ale za chvíli jsem se do toho dostala a prohlížení mi připadalo celkem intuitivní“
- „Čas od času aplikace zamrzne nebo trvá chvíli, než např. otočím obrázek. Někdy když chci otočit obrázek doleva, otočí se doprava“
- „Mám problémy s kliknutím, moc se mi to nedaří. Ale otáčení a zvětšování je pro mě příjemná záležitost, docela jsem si pohrála“

Dále si uživatelé stěžovali, že nevidí, co se právě děje, jestli je kamera vidí nebo ne, jestli správně provádí gesto. Jako budoucí rozšíření se proto jeví dobré zobrazovat uživateli např. obrázek, který bude zobrazovat stav rozpoznávání. Tři uživatelé měli přání rozšířit aplikaci o gesto pro přesun mezi fotkami (filmový pás).

Na vybrané skupině osob bylo testováno, jak rychle jsou schopni vykonat předem zadanou posloupnost operací. Testování proběhlo na následující sekvenci: *výběr 5. fotky, přiblížení, oddálení, rotace vlevo, výběr 9. fotky, rotace vlevo, rotace vpravo*. Pro porovnání s jinými aplikacemi prohlížení fotek jsem zvolil programy *Windows Fotogalerie* a *Microsoft Office Picture Manager* (dále MO Office Picture). Tabulka 6.2 udává srovnání dosažených časů. V porovnání s výše zmíněnými aplikacemi je vidět, že vykonání dané sekvence trvá delší dobu. Nejvíce se uživatel zdržel při provádění kliku, dále je zde malá časová prodleva před odezvou na gesto.

<b>program</b>	<b>Window Fotogalerie</b>	<b>MO Office Picture</b>	<b>má aplikace</b>
<b>číslo měření</b>	<b>čas [s]</b>		
1.	14,4	17,9	24,3
2.	16,8	20,4	28,7
3.	15,4	18,5	25,5
4.	13,7	18,8	50,4
5.	16,9	21,4	25,3
<b>průměrný čas</b>	15,5	19,4	30,8

Tabulka 6.2: Měření časů dosažených při provádění sekvenci příkazů

Táž skupina byla požádána o srovnání aplikace s výše zmíněnými prohlížeči na stupnici od 0 (výrazně horší ovladatelnost), 5 (stejná) až do 10 (razantně lepší), viz tabulka 6.3. Je vidět, že uživatelům se ovládání líbilo více než konvenční přístup. Nejvíce se však líbila právě možnost ovládat aplikaci gesty. Na druhou stranu si nejvíce stěžovali na klik.

<b>program</b>	<b>Window Fotogalerie</b>	<b>MO Office Picture</b>
<b>respondent</b>	<b>srovnání</b>	
1.	7	8
2.	8	8
3.	5	6
4.	4	5
5.	5	6

Tabulka 6.3: srovnání aplikací

## **Shrnutí**

Omezení přesnosti klasifikace je bráno s ohledem na kvalitu dané webkamery (mohou být levné, nekvalitní čočka, apod.), daného osvětlení (snímaná plocha není rovnoměrně osvětlena, některé části ruky mohou mít jiný odstín, následkem toho jsou tyto části nedetekovány a ztrácí se tak informace o gestu) a vzdálenost (čím dále, tím menší objem ruky a horší rozpoznávací schopnost – prsty se mohou ztrácet). Jedním z předpokladů je pozice kamery zprava (možné i zleva, ovšem klasifikace na tuto možnost není rozšířena, proto může dojít k chybnému rozpoznání).

## Kapitola 7

### Závěr

Tato práce byla zaměřena na implementaci jednoduchého rozhraní pro ovládání počítače gesty. Mnou zvolená metoda je jedna z používaných přístupů, jak lze provádět klasifikaci gest. Jejím porovnáním s podobnými postupy lze najít spoustu nedostatků, avšak mohu tvrdit, že i tak dává dobré výsledky. Gesta pro rotaci, změnu velikosti a ukazování dávají dobré výsledky. Problém nastává u kliku, kdy není možné jednoznačně určit, kdy nastal. Tento nedostatek je způsoben převážně modelem použitým pro klasifikaci. Nalezené body oscilují kolem své základní polohy a znemožňují tak přesné určení změny polohy konečku prstu. K nevýhodám patří nemožnost přidávání nových gest a nízká škálovatelnost (lze rozšířit pouze na gesta obsahující čtyři nebo pět prstů).

Samotné testování bylo značným způsobem ovlivněno světelnými podmínkami, proto v případě použití v jiných lokalitách budou testy dávat jiné výsledky. Avšak tyto nedostatky se dají odstranit nebo zmírnit přesnější segmentací obrazu a výběrem většího počtu příznaků popisující dané gesto (viz kapitola o návrhu).

Co se týče budoucí práce, určitě se zaměřit na lepší metody segmentace obrazu, např. vyhledáním nejčastěji se vyskytujících pixelů barvy lidské kůže nebo zvolit šablony pro detekci ruky v různých polohách. Dále pro klasifikace gest lze použít informace o zakřivení prstů, algoritmy pro předpověď budoucí polohy prstů či vytvořit trojdimenzionální model ruky a ten posléze pasovat do příznaků snímku. Aktuální klasifikace je vytvořena pro jednu ruku, jako rozšíření lze přidat detekci druhé ruky a rozšířit tak počet rozpoznávaných gest. Pro dynamická gesta použít např. HMM.



## Literatura

- [1] Bradski G. R., K. A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008, ISBN 978-0-596-51613-0.
- [2] Wikipedia. *Distortion (optics)* [online]. 4/2011 [citováno dne 2011-04-30]. Dostupné z: <[http://en.wikipedia.org/wiki/Distortion\\_\(optics\)](http://en.wikipedia.org/wiki/Distortion_(optics))>
- [3] Wikipedia. *Homography* [online]. 4/2011 [Citováno dne 2011-04-30]. Dostupné z: <<http://en.wikipedia.org/wiki/Homography>>
- [4] Tech-Faq. *HSV* [online]. [Citováno dne 2011-04-30]. Dostupné z: <<http://www.tech-faq.com/hsv.html>>
- [5] Oliveira, Victor. *Skin Detection using HSV color space* [online]. 2009 [citováno dne 2011-04-30]. Dostupné z: <<http://www.matmidia.mat.puc-rio.br/sibgrapi2009/media/posters/59928.pdf>>
- [6] Španěl, Michal. *Klasifikace a rozpoznávání: Aplikace* [online]. 2009 [citováno dne 2011-04-30]. Dostupné z: <[https://www.fit.vutbr.cz/study/courses/IKR/public/prednasky/08\\_clustering/IKR\\_Clustering.pdf](https://www.fit.vutbr.cz/study/courses/IKR/public/prednasky/08_clustering/IKR_Clustering.pdf)>
- [7] The IDL Astronomy User's Library. *Eroding and Dilating Image Objects* [online]. 6/2007 [citováno dne 2011-04-30]. Dostupné z: <[http://idlastro.gsfc.nasa.gov/idl\\_html\\_help/Eroding\\_and\\_Dilating\\_Image\\_Objects.html](http://idlastro.gsfc.nasa.gov/idl_html_help/Eroding_and_Dilating_Image_Objects.html)>
- [8] Wikipedia. *Virtual reality* [online]. 4/2011 [Citováno dne 2011-04-30]. Dostupné z: <[http://en.wikipedia.org/wiki/Virtual\\_reality](http://en.wikipedia.org/wiki/Virtual_reality)>
- [9] Billinghurst, Mark. *Gesture based interaction* [online]. 5/2009 [Citováno dne 2011-04-30]. Dostupné z: <<http://www.billbuxton.com/input14.Gesture.pdf>>
- [10] Wikipedia. *Gesture recognition* [online]. 4/2011 [Citováno dne 2011-04-30]. Dostupné z: <[http://en.wikipedia.org/wiki/Gesture\\_recognition](http://en.wikipedia.org/wiki/Gesture_recognition)>
- [11] Wikipedia. *Kinect* [online]. 4/2011 [Citováno dne 2011-04-30]. Dostupné z: <<http://en.wikipedia.org/wiki/Kinect>>
- [12] TechEncyclopedia. *Kinect* [online]. [Citováno dne 2011-04-30]. Dostupné z: <<http://www.answers.com/topic/project-natal>>

## Seznam obrázků

Obrázek 2.1	Ovládání her bez použití herního ovladače (zdroj [12]) .....	3
Obrázek 2.2	Rukavice CyberGlove pro přenos informací pro rozpoznání Gesta (zdroj [9]) .....	4
Obrázek 2.3	Zařízení Theramin pro ovládání zvuku (zdroj [9]).....	4
Obrázek 2.4	Rozpoznání ukazování, špička prstu použita jako kreslící Zařízení (zdroj [9]).....	6
Obrázek 3.1	Radiální zakřivení (převzato z [2]) .....	7
Obrázek 3.2	Tečné zakřivení (převzato z [1]) .....	8
Obrázek 3.3	HSV model (zdroj [4]) .....	9
Obrázek 3.4	Segmentace barvy lidské kůže (zdroj [6]) .....	9
Obrázek 4.1	Diagram systému, přehled jednotlivých kroků .....	10
Obrázek 4.2 a)	Detekce rohů plochy .....	11
Obrázek 4.2 b)	Výřez rohu .....	11
Obrázek 4.2 c)	Kontrola rohu .....	11
Obrázek 4.3 a)	červená plocha .....	12
Obrázek 4.3 b)	zelená plocha .....	12
Obrázek 4.3 c)	modrá plocha .....	12
Obrázek 4.4 a)	Odečtení modré a zelené .....	13
Obrázek 4.4 b)	Odečtení červené a zelené .....	13
Obrázek 4.4 c)	Odečtení červené a modré .....	13
Obrázek 4.5	Prahování rozdílů č. a m. barvy .....	13
Obrázek 4.6 a)	H složka .....	14
Obrázek 4.6 b)	S složka .....	14
Obrázek 4.6 c)	V složka .....	14
Obrázek 4.6 d)	log. součin složek .....	14
Obrázek 4.7	Podobnost dlaně, resp. prstů s kružnicí, resp. s elipsou .....	15
Obrázek 4.8 a)	ořezání prstů ruky, zbývá ořezaná dlaň (zelená barva), červená tečka značí střed .....	16
Obrázek 4.8 b)	odečtení dilatované dlaně od segmentované ruky, černě zbylé prsty .....	16
Obrázek 4.8 c)	střed prstů označeny modře, konečky oranžově .....	16
Obrázek 4.9 a)	vzdálenosti středu a konečku prstu při plynulém pohybu .....	17
Obrázek 4.9 b)	vzdálenosti středu a konečku prstu při kliku .....	17
Obrázek 4.9 c)	Diference vzdáleností při kliku .....	18
Obrázek 5.1	Koncept rozhraní .....	20

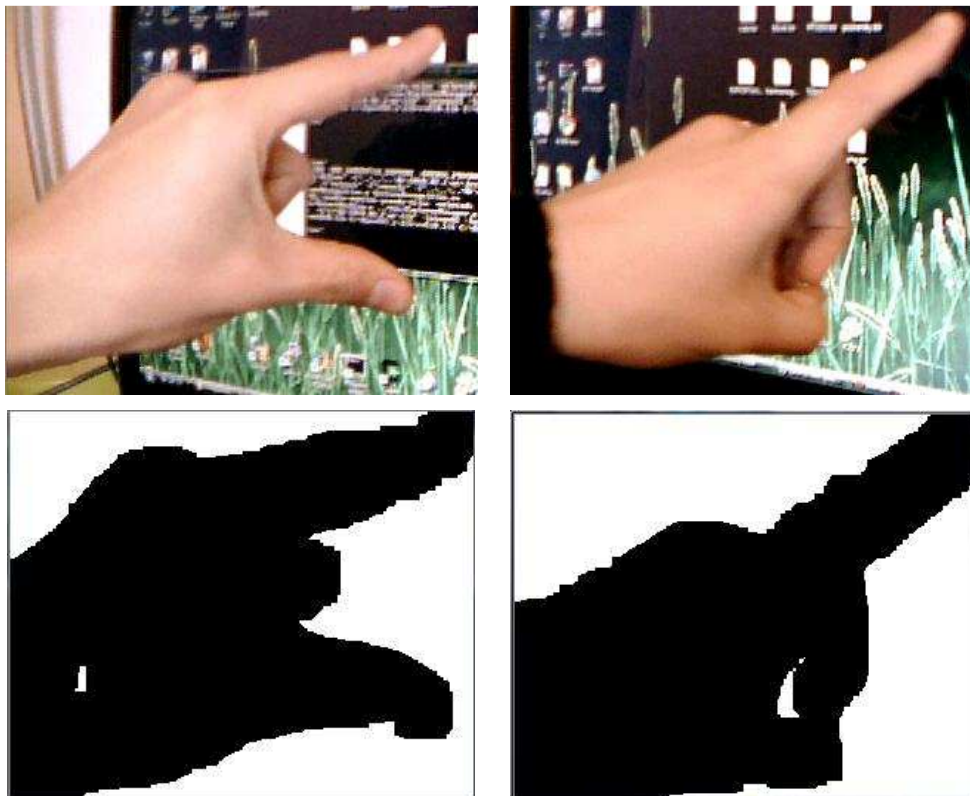
Obrázek 5.2	Hierarchie tříd .....	21
Obrázek 5.3(a)	Detekce bodů .....	22
Obrázek 5.3(b)	Změna vzdálenosti hran .....	22
Obrázek 5.4 a)	demo aplikace .....	27
Obrázek 5.4 b)	Vstupní rozhraní .....	28
Obrázek 5.4 c)	Nastavení parametrů gest .....	28

## Dodatek A

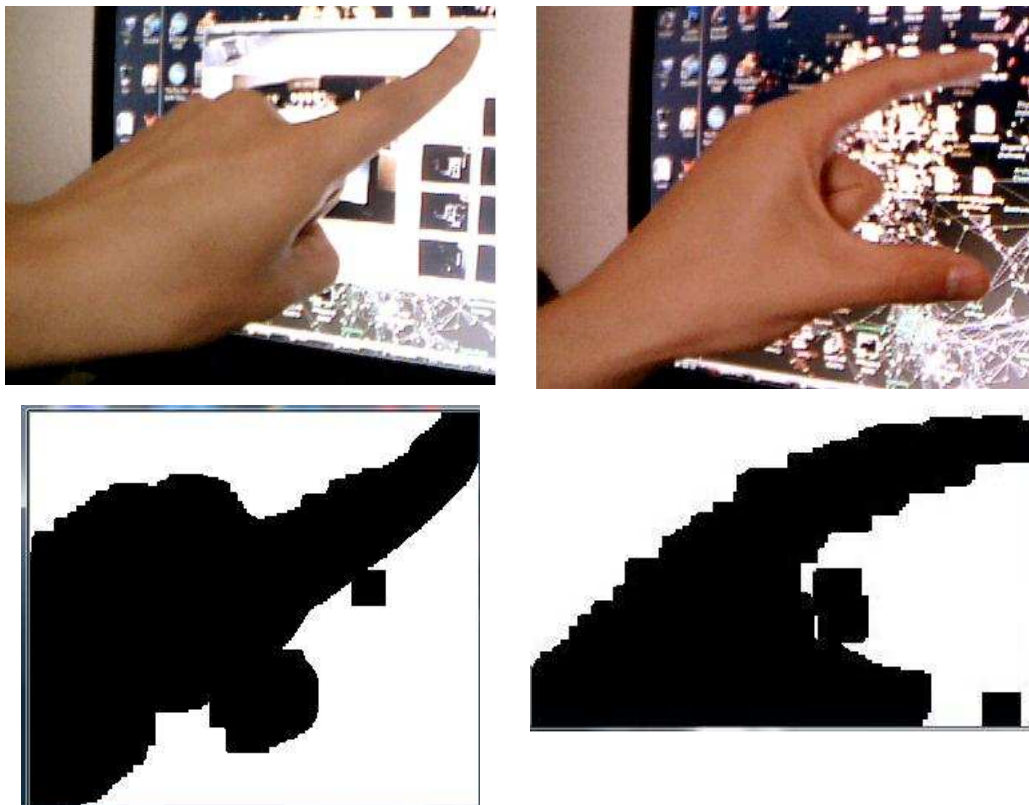
### Obsah CD

- složka **run**:
  - složka **config**: konfigurační soubory a aplikační obrázky
  - složka **demoimages**: obrázky pro prohlížeč fotek
  - spustitelná aplikace
- složka **doc**:
  - složka **src**: zdrojové texty dokumentace ve formátu doc
  - složka **pdf**: dokumentace ve formátu pdf
- složka **src**:
  - zdrojové kódy aplikace
- složka **test**:
  - testovací soubory pro demonstraci aplikace rozpoznávat gesta

## Příloha



Příloha 1: ideální světelné podmínky, segmentace dodává dostatek informací pro rozpoznání gesta



Příloha 2: špatné osvětlení, částí ruky chybí, nedostatek informací pro klasifikaci