

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

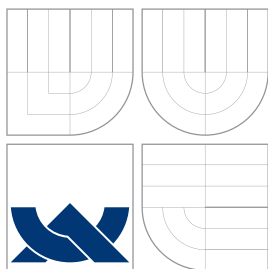
DETEKCE POHYBU RUKY PRO OVLÁDÁNÍ APLIKACÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

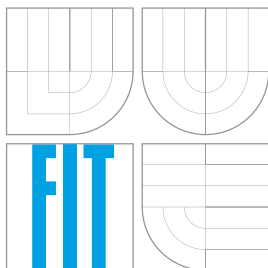
AUTOR PRÁCE
AUTHOR

Bc. JURAJ BLAHO

BRNO 2009



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

DETEKCE POHYBU RUKY PRO OVLÁDÁNÍ APLIKACÍ

HAND MOTION RECOGNITION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JURAJ BLAHO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL HRADIŠ

BRNO 2009

Abstrakt

Účelem této diplomové práce je návrh a implementace nového počítačového rozhraní založeného na detekci a sledování ruky v obrazu z jedné kamery. Vytvořené rozhraní nemá žádné speciální požadavky na hardware a je použitelné i na běžném počítači a s obyčejnou webkamerou. Implementované rozhraní bylo použito k vytvoření aplikace, která umožňuje syntetizovat události vstupů z klávesnice a myši a tak ovládat existující programy bez nutnosti zásahu do jejich zdrojového kódu. Dalším přínosem této práce je představení inovativní metody automatického získávání trénovacích dat pro detektory ruky. Použitím této metody je možné získat tisíce trénovacích příkladů v průběhu několika málo hodin.

Abstract

The aim of this work is to design and implement a novel computer interface based on detection and tracking of a hand in an image from a single camera. The created interface doesn't require any special hardware and it is possible to use it on a common computer with standard web-camera. The implemented interface was used to create an application, which is able to synthesize keyboard and mouse input events and this way it is able to control existing programs without the need to change their source code. Another contribution of this work is a novel method of automatic data acquisition for training of hand detectors. By using this method it is possible to collect thousands of training examples in a few hours.

Klíčová slova

počítačové rozhraní, ruka, detekce, sledování pohybu, počítačové vidění

Keywords

computer interface, hand, detection, motion detection, computer vision

Citace

Juraj Blaho: Detekce pohybu ruky pro ovládání aplikací, diplomová práce, Brno, FIT VUT v Brně, 2009

Detekce pohybu ruky pro ovládání aplikací

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše.

.....

Juraj Blaho
13. mája 2009

Poděkování

Děkuji svému vedoucímu Ing. Michalovi Hradišovi za poskytnutí odborné pomoci při vypracování této práce.

© Juraj Blaho, 2009.

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	3
2	Existujúce prístupy	5
2.1	Metódy detekcie ruky a rozpoznanie pózy	5
2.1.1	Normalizovaná korelácia	6
2.1.2	Detekcia farby kože	7
2.1.3	Metódy založené na kontúre	8
2.1.4	AdaBoost	8
2.1.5	Ďalšie metódy	10
2.2	Adaptácia klasifikátorov	11
2.2.1	MAP adaptácia	11
2.3	Metódy sledovania pohybu	11
2.3.1	Flocks of Features	12
2.3.2	Sledovanie pomocou kontúry	12
2.4	Rozpoznávanie dynamických gest	13
2.4.1	Skryté Markovove modely	13
3	Navrhnuté rozhranie pre ovládanie počítača rukami	14
3.1	Adaptívny model farby kože	15
3.2	Detekcia tváre	17
3.3	Detekcia rúk a rozpoznanie pózy	17
3.4	Automatické získavanie trénovacích dát pre detektory rúk	18
3.5	Sledovanie pohybu	20
3.6	Rozpoznanie dynamických gest	23
3.7	Generovanie udalostí	27
4	Implementácia rozhrania a demonštračnej aplikácie	28
4.1	Interpretácia pohybu ruky a rozpoznania póz	30
4.2	Interpretácia dynamických gest	32
4.3	Generovanie udalostí v operačnom systéme Windows	32
5	Testy	34
5.1	Modely farby kože	34
5.2	Úspešnosť detektorov rúk	35
5.3	Výpočtová náročnosť rozhrania	37
5.4	Efektivita práce s rozhraním	37
6	Záver	41
6.1	Námety na budúce pokračovanie práce	42

Literatúra	43
A Zoznam netlačených príloh	45
B Odvodenie akceleračnej funkcie	46

Kapitola 1

Úvod

V súčasnosti sú najčastejšie spôsoby ovládania počítača klávesnica a myš. Tieto sú však neprirodzené a človeku dlho trvá kým si na ne zvykne. Spomeňme napríklad čas potrebný na naučenie sa písania na klávesnici. Čo sa týka prirodzenosti ovládania sú na tom výrazne lepšie postupne sa rozširujúce dotykové displeje, kde užívateľ môže priamo manipulovať so zobrazenými objektmi. Avšak spoločnou nevýhodou všetkých spomenutých rozhraní je potreba bezprostredného kontaktu s nimi, či dokonca nutnosť pre ich efektívne použitie napríklad sedieť pri stole. To výrazne obmedzuje ich použiteľnosť v situáciách, keď človek okrem ovládania počítača potrebuje vykonávať ešte inú činnosť, prípadne sa potrebuje voľne pohybovať.

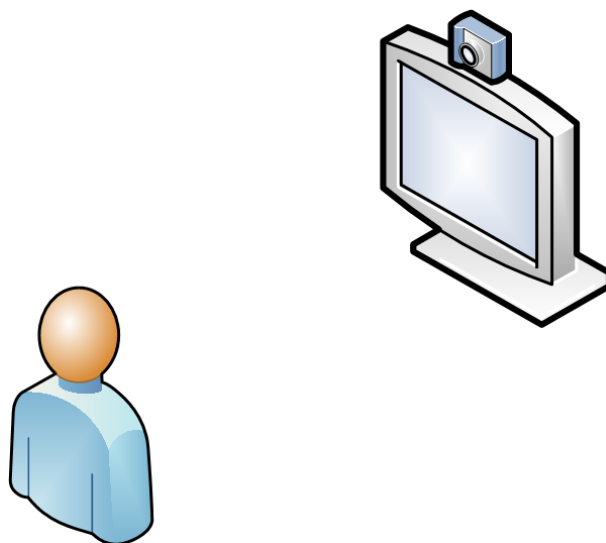
Bolo by preto vhodné vytvoriť nejaké lepšie komunikačné rozhranie medzi človekom a strojom. Jednou z možností riešenia tohto problému je inšpirovať sa metódami komunikácie, ktoré používajú ľudia medzi sebou. Najbežnejším spôsobom komunikácie je hovorená reč, avšak veľmi dôležitá je aj neverbálna komunikácia ako napríklad gestá tváre a rúk.

Zadávanie rečových príkazov počítaču je v súčasnosti do značnej miery zvládnuté a je využívané napríklad v nových verziách operačných systémov alebo ako hlasové vytáčanie v mobilných telefónoch.

Naproti tomu rozhrania založené na gestách zatiaľ príliš rozšírené nie sú, i keď existujú práce, ktoré sa touto problematikou zaoberajú. Podobne ako majú gestá a reč iné využitie v komunikácii medzi ľuďmi, bude odlišné aj ich uplatnenie pri ovládaní počítača. Výhodou ovládania počítača gestami rúk oproti reči je možnosť ich použitia v hlučných prostrediach alebo naopak v prostrediach, kedy je potrebné zachovať ticho. Naproti tomu ich nevýhodou je, že ruky už nebude možné použiť na inú činnosť ako napríklad písanie na klávesnici, riadenie auta, atď.

Rozhrania založené na detekcii a sledovaní rúk v obraze z kamery môžu byť použité na diaľkové ovládanie počítača, či domácej elektroniky. Taktiež by sa dali nasadiť do informačných terminálov a tabúl, kde by užívateľ pohybom ruky mohol napríklad posúvať zobrazenú stránku, či mapu. Uplatnenie môžu nájsť aj v medicíne pri chirurgických operáciách, kedy je nutné zabezpečiť bezdotykovú komunikáciu doktora s počítačom, ktorý zobrazuje dáta o pacientovi [17]. Ďalšie využitie takýchto rozhraní je napríklad na dekódovanie znakov reči, či výuku bojových umení [9].

Účelom tejto práce je práve návrh a implementácia nového užívateľského rozhrania umožňujúceho ovládať počítač na základe sledovania rúk v obraze. Obraz je získavaný z jednej kamery, ktorá je umiestnená pred užívateľom v blízkosti monitora prípadne plátna tak, ako je to znázornené na obrázku 1.1. Vytvorené rozhranie je schopné pracovať na bežnom počítači a nevyžaduje žiadne špeciálne značky na rukách užívateľa. Ovládanie prebieha



Obr. 1.1: Kamera je v navrhnutom rozhraní umiestnená pred užívateľom v blízkosti monitora prípadne plátna.

na základe sledovania pohybu ruky v obraze a rozpoznávania rôznych póz. Rozhranie je tiež schopné detekovať niekoľko jednoduchých dynamických gest. V rámci tejto práce bola tiež implementovaná demonštračná aplikácia založená na vytvorenom rozhraní umožňujúca ovládanie existujúcich programov pomocou generovania udalostí klávesnice a myši. Ďalším prínosom tejto práce je predstavenie inovatívnej metódy automatického získavania tréningových dát pre detektory rúk.

V tomto dokumente budú často spomínané pojmy póza a gesto. Póza bude znamenať statickú konfiguráciu ruky, pričom gesto bude označovať pohyb ruky.

Tento dokument je rozdelený do niekoľkých kapitol. Za úvodom nasleduje kapitola popisujúca problematiku ovládania počítača rukami. Táto kapitola sa zaoberá existujúcimi metódami detekcie rúk, rozpoznávaním rôznych póz a sledovaním objektov. Kapitola 3 popisuje navrhnuté užívateľské rozhranie a jeho jednotlivé časti. V tejto kapitole je tiež predstavená metóda automatického získavania tréningových dát pre detektory rúk. Nasledujúca kapitola s číslom 4 pojednáva o implementácii navrhnutého rozhrania a demonštračnej aplikácie. Táto kapitola tiež vysvetľuje spôsob použitia implementovaného rozhrania v demonštračnej aplikácii. V predposlednej kapitole sa nachádzajú výsledky testov dôležitých častí rozhrania ako aj výsledky testov efektivity práce s rozhraním. V závere je zhodnotené dosiahnutie stanovených cieľov a sú uvedené námety na budúce možné vylepšenie rozhrania a demonštračnej aplikácie.

Táto diplomová práca nadväzuje na semestrálny projekt, v rámci ktorého boli skúmané existujúce prístupy k detekcii rúk v obraze a metódy sledovania pohybu. Táto diplomová práca vychádza z návrhu rozhrania vytvoreného v rámci semestrálneho projektu. Tento návrh bol však čiastočne upravený na základe poznatkov získaných pri implementácii.

Kapitola 2

Existujúce prístupy

Užívateľské rozhranie založené na vizuálnom sledovaní rúk musí v reálnom čase riešiť niekoľko problémov. V prvom rade je potrebné získať obraz z kamery, prípadne niekoľko obrazov z rôznych kamier a predspracovať ich. Ďalšou úlohou je detekcia rúk v získanom obraze a rozpoznanie póz. Rozhranie tiež musí sledovať pohyb detekovaných rúk. Nakoniec musia byť získané detekcie rúk a ich pohyb interpretované.

Rozhranie môže vystačiť s jedným obrazom. V prípade viacerých obrazov je však možné použiť aj informáciu o vzdialenosti rúk, či dokonca určiť presnú polohu alebo aj tvar ruky v priestore. Dve kamery boli použité napríklad v prácach [18, 10, 7]. Predspracovanie obrazu môže zahŕňať napríklad odstránenie vplyvov šumu alebo osvetlenia. Následne je v získanom obraze detekovaná poloha a prípadne konkrétne pózy, jednotlivých rúk.

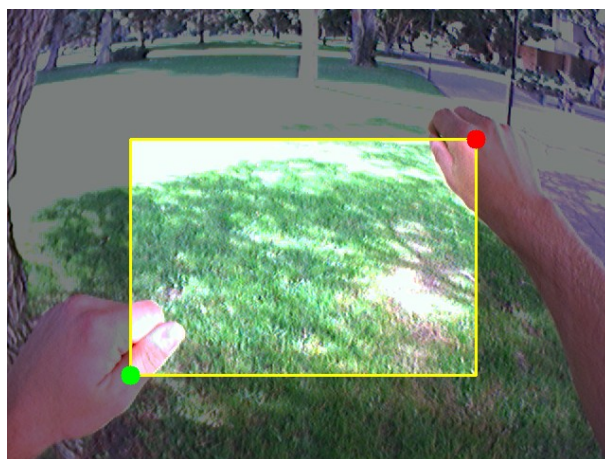
Sledovanie pohybu je potrebné na určenie súvislostí medzi detekciami v po sebe nasledujúcich snímkoch. To je nevyhnutné pre vyhodnotenie dynamických gest alebo pre udržanie zamerania (focus) na určitú ruku. Sledovanie tiež môže výrazne urýchliť celý systém. Princíp urýchlenia spočíva v tom, že nebude potrebné vždy prehľadávať celé obrazy, aby sa našli ruky, ale bude stačiť hľadať iba na pozícii predpovedanej z predošlých snímkov [8, 4].

Poslednou, ale veľmi dôležitou súčasťou rozhrania je interpretácia získaných detekcií a ich využitie pre ovládanie aplikácií. Jednoduché rozhranie môže fungovať tak, že na základe detekcie konkrétnych póz vyvolá zvolené akcie. Tento typ interpretácie je použitý v systéme popísanom v [11]. Prirodzeným spôsobom interpretácie je simulovať ovládanie klasickou myšou [8]. Takéto rozhranie bude využívať detekovanú polohu ruky pre umiestnenie kurzora a na základe rozpoznania pózy vyvolávať akcie podobne ako pri stláčaní tlačidiel myši. Vhodným rozšírením tohoto konceptu je sledovanie oboch rúk užívateľa. Polohami dvoch rúk sa dá napríklad vymedziť obdĺžniková oblasť ako je to naznačené na obrázku 2.1. Určite by bolo možné nájsť viac spôsobov využitia sledovania oboch rúk, a dokonca niektoré systémy, ako je rozpoznávanie znakov reči, sa bez neho nezaobídu.

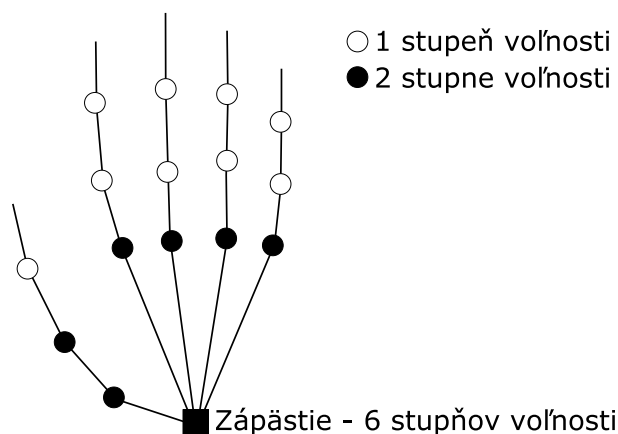
2.1 Metódy detekcie ruky a rozpoznanie pózy

Problém rozpoznania rúk v obraze sa dá riešiť veľkým množstvom metód používaných na detekciu všeobecných objektov. V tejto sekcii sú popísané niektoré metódy aplikované konkrétne na detekciu rúk.

Hlavným problémom pri detekcii rúk v obraze je veľká variabilita ich vzhľadu. Nielenže ruky rôznych užívateľov vyzerajú inak, ale aj ruka jedného užívateľa môže zaujímať veľké množstvo póz. Štyri prsty ruky majú 3 kĺby, ktoré každému z nich dávajú 4 stupne voľnosti. Palec má dokonca 5 stupňov voľnosti. Pokiaľ započítame aj možnosť rotácie a zmeny pozície



Obr. 2.1: Využitie oboch rúk užívateľa pre vymedzenie obdĺžnikovej oblasti. (prevzaté z [8])



Obr. 2.2: Model ruky. Približne 27 stupňov voľnosti

celej ruky, dostaneme celkovo 27 stupňov voľnosti [15]. Model ruky s vyznačenými stupňami voľnosti je znázornený na obrázku 2.2.

2.1.1 Normalizovaná korelácia

Pravdepodobne najjednoduchšou metódou detekcie objektov je výpočet korelácie vzorového obrázka hľadaného objektu s každou rovnako veľkou časťou vstupného obrazu. Aby výsledky neboli ovplyvnené nerovnomerným rozložením jasu vo vstupnom obraze, je nutné ich normalizovať. Predstavme si vzorový obrázok hľadaného objektu s šírkou W a výškou H ako vektor $\vec{a}$ dimenzie $W \cdot H$. Aktuálne skúmanú časť vstupného obrazu si predstavme ako vektor $\vec{b}$ rovnakej dimenzie. Potom výsledok normalizovanej korelácie je popísaný vzťahom 2.1, ktorý je možné tiež interpretovať ako kosínus uhla medzi vektormi $\vec{a}$ a $\vec{b}$.

$$\frac{\vec{a} \cdot \vec{b}}{\sqrt{(\vec{a} \cdot \vec{a})(\vec{b} \cdot \vec{b})}} \quad (2.1)$$

Normalizovaná korelácia bola použitá na detekciu rúk v práci [5] publikovanej už v roku 1998. Účelom systému bolo diaľkové ovládanie televízora. Kamera sledovala užívateľa spredu. Ten otvorenou rukou televízor zapol a rukou zaťatou v päšť vypol. Pohybom otvorenej ruky

sa dali prepínať kanály a meniť hlasitosť. Pre zlepšenie výsledkov metódy bolo odstraňované pozadie pomocou plávajúceho priemeru (running average).

2.1.2 Detekcia farby kože

Farba kože má charakteristický a relatívne malý rozsah farebných odtieňov. Vďaka tomu sa dá dobre detekovať a môže byť použitá k vyhľadaniu častí ľudského tela a konkrétne rúk v obraze.

Rozhodovanie, či daná farba predstavuje kožu, sa dá vykonávať samostatne pre každý pixel a je možné použiť ľubovoľný klasifikátor. Klasifikátor obvykle modeluje dve triedy: farbu kože a farbu pozadia. Tieto triedy môžu byť modelované histogramom, prípadne jednou alebo niekoľkými Gaussovými funkciami. Je vhodné zvoliť farebný priestor, ktorý dokáže aspoň do určitej miery potlačiť vplyvy osvetlenia. Kvôli možnosti rýchleho výpočtu je často používaný normalizovaný RG priestor, ktorého dve zložky sú dané vzťahmi:

$$r = \frac{R}{R + G + B} \quad (2.2)$$

$$g = \frac{G}{R + G + B} \quad (2.3)$$

Iným vhodným priestorom je odtieň a sýtosť z HSV farebného modelu. Ďalšie modely používané na detekciu kože, v ktorých sa dá priamo zanedbať jasová zložka, sú napríklad YCbCr a YUV.

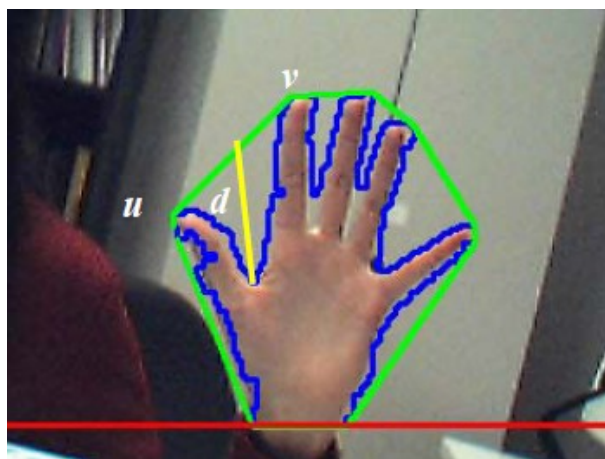
V práci [10], sú detekované spojené zhluky zodpovedajúce farbe kože, pričom najväčšie zhluky sú považované za ruky. Je používaný RGB farebný priestor a trieda farby kože aj trieda farby pozadia sú modelované 3D histogramom. Na základe kontúry detekovaných zhlukov je potom rozhodnuté, či sa jedná o ľavú alebo pravú ruku a v akej sa nachádza póze. Pre potlačenie problémov s objektmi farebne podobnými farbe kože, je pred detekciou kože z obrazu odstránené pozadie.

Ďalší systém bezvýhradne sa spoliehajúci na detekciu farby kože je popísaný v práci [11]. Tento systém je použitý na ovládanie hier. Používa HS farebný priestor, pričom nemodeluje triedu pozadia. Trieda farby kože je modelovaná jednou 2D Gaussovou funkciou. Systém segmentuje vstupný obraz na základe farby kože. Podľa kontúry nájdeného zhluku je rozpoznávaných niekoľko póz, ktoré sú interpretované ako príkazy. Farba kože sa pred použitím systému natrénuje tak, že užívateľ umiestni ruku na požadované miesto v obraze.

V [4] je detekcia kože použitá ako predspracovanie pre detektor založený na metóde AdaBoost. Model kože je adaptovaný na základe spoľahlivého detektora tváre.

Ako pomocné kritérium je farba kože použitá pri sledovaní objektov v [8]. Táto metóda sledovania je stručne predstavená v sekcii 2.3.1 na strane 12.

Nedostatkom rozoznávania farby kože pre detekciu rúk je, že ostatné časti tela majú tiež farbu kože. A taktiež iné, napríklad drevené objekty môžu mať veľmi podobnú farbu. Aj v prípade vhodne zvoleného farebného priestoru závisí farba kože na osvetlení a použitej kamere. Z týchto dôvodov vyplýva, že detekciu kože nie je vhodné používať samostatne, ale radšej v kombinácii s inými metódami. Ďalšou možnosťou využitia informácie o farbe kože je urýchlenie existujúcich systémov tak, že nebude nutné hľadať ruky v celom obraze, ale bude stačiť hľadať iba v miestach, kde bola detekovaná koža.



Obr. 2.3: Princíp rozpoznania pózy ruky na základe kontúry predstavený v práci [11]. Dva susedné body kontúry ležiace na konvexnej obálke sú označené u a v . Maximálna vzdialenosť kontúry od konvexnej obálky medzi týmito bodmi je označená d . Póza ruky je určená podľa hodnoty $\bar{d}$ označujúcej priemer maximálnych vzdialeností. Obrázok je prevzatý z [11].

2.1.3 Metódy založené na kontúre

Kontúry objektov je možné zo vstupného obrazu získať segmentáciou. V prípade detekcie rúk sa na segmentáciu dá použiť napríklad farba kože.

Klasifikácia pózy ruky na základe kontúry je použitá v práci [11]. Metóda najskôr určí konvexnú obálku získanej kontúry. Medzi každými dvomi susednými bodmi kontúry, ktoré ležia na konvexnej obálke, sa vypočíta maximálna vzdialenosť d_i kontúry od konvexnej obálky. Spriemerovaním získaných hodnôt dostaneme príznak $\bar{d}$, ktorý slúži na rozpoznanie pózy. Metóda je na základe tohto jediného príznaku schopná rozpoznať otvorenú ruku s oddelenými prstami, otvorenú ruku so spojenými prstami a ruku zatatú v päšť. Predpokladá sa totiž, že napríklad pre ruku s oddelenými prstami bude hodnota $\bar{d}$ vyššia ako pre ruku so spojenými prstami. Systém okrem rozpoznania pózy dokáže určiť aj natočenie ruky. Princíp tejto metódy je znázornený na obrázku 2.3.

Metóda popísaná v [10] pre rozpoznanie rôznych póz využíva počet výrazných výstupkov a priehlbín na kontúre ruky tak, ako je to znázornené na obrázku 2.4. Metóda rozlišuje pózy ukazovanie a štipanie (pinching). Ukazovanie je rozpoznané v prípade, že bol nájdený len jeden výstupok. Nájdený výstupok je považovaný za ukazovák a je určený jeho smer. V prípade dvoch výstupkov je póza rozpoznaná ako štipanie a výstupky sú považované za palec a ukazovák.

Iné pokročilejšie metódy rozpoznávajú ruku a pózy na základe vhodnej reprezentácie kontúry, ktorá je použitá napríklad ako vstup pre skrytý Markovov model (Hidden Markov Model – HMM). Spoločným nedostatkom všetkých metód založených na kontúre je fakt, že ignorujú vnútornú štruktúru objektov.

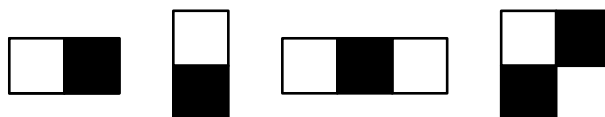
2.1.4 AdaBoost

AdaBoost je metóda spájania takzvaných slabých klasifikátorov za účelom získania silného klasifikátora, ktorý má vyššiu úspešnosť ako pôvodné slabé klasifikátory. Medzi výhody tejto metódy patrí, že dobre generalizuje a nemá sklony k pretrénovaniu.

V článku [14] Paul Viola a Michael Jones popísali použitie metódy AdaBoost pre detekciu tváre. Ako základ pre slabé klasifikátory použili Haarove príznaky. Dobrou vlastnosťou



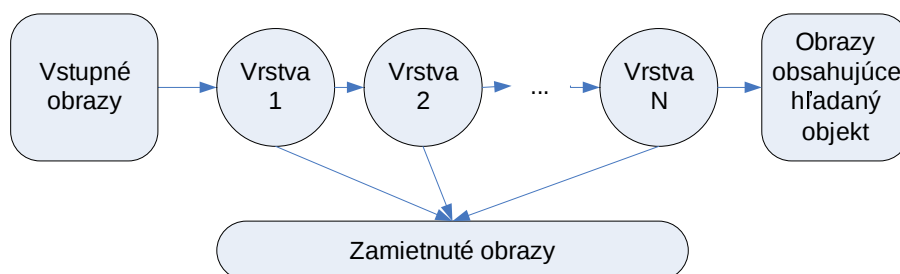
Obr. 2.4: Detekcia výstupkov a priehlbni na kontúre ruky. Nie všetky priehlbne sú detekované. Póza je určená len na základe počtu výstupkov. Kontúra je získaná z obrazu, z ktorého bolo najskôr odstránené pozadie a následne bola detekovaná farba kože. Prevzaté z [10].



Obr. 2.5: Štyri typy Haarových príznakov slúžiacie ako základ slabých klasifikátorov pre metódu AdaBoost, ktoré boli použité v práci [14].

týchto príznakov je, že ich hodnota nie je príliš ovplyvnená osvetlením. Podľa mien autorov sa táto metóda AdaBoost v kombinácii s Haarovými príznakmi často označuje ako *Viola&Jones detektor*. Typy príznakov použité v [14] sú ukázané na obrázku 2.5.

Hodnota príznaku sa určí jeho umiestnením nad obraz a následným odčítaním súčtu hodnôt pixelov pod bielymi obdĺžnikmi od súčtu hodnôt pixelov pod čiernymi obdĺžnikmi. Pomocou integrálneho obrazu je možné tento výpočet vykonať v konštantnom čase. Vďaka tomu je výsledný klasifikátor rýchly. Pre ešte výraznejšie zrýchlenie býva klasifikácia rozdelená do niekoľkých vrstiev, pričom v každej vrstve je časť vstupných obrazov zamietnutá a do ďalšej vrstvy ich potom vstupuje menší počet. Takže pre väčšinu obrazov klasifikátor nemusí byť vyhodnotený celý, ale stačí vyhodnotiť iba niekoľko vrstiev, kým obraz nie je zamietnutý. Princíp je znázornený na obrázku 2.6.



Obr. 2.6: Kaskáda klasifikátorov. V každej vrstve sú niektoré zo vstupných obrazov zamietnuté. Výstupom poslednej vrstvy sú obrazy s hľadaným objektom.

Viola&Jones detektor bol na detekciu rúk použitý v práci [8]. Na rozpoznanie konkrétnej pózy boli detektory usporiadané do dvoch úrovní. Prvá úroveň zistí, či sa na danej pozícii nachádza ruka a prípadne upresní jej polohu. V druhej úrovni je paralelnou aplikáciou niekoľkých klasifikátorov rozpoznaná póza. Pre lepšie rozpoznanie pózy sú pri tréningu klasifikátorov druhej úrovne použité ako negatívne vzorky príklady rúk z ostatných tried. Spomínaná práca tiež predstavuje nové typy príznakov, kde sa jednotlivé obdĺžniky nemusia dotýkať a prípadne sa môžu aj prekrývať.

Ďalšie použitie Viola&Jones detektora na rozpoznanie rúk je popísané v [4]. Systém je veľmi podobný predošlému avšak klasifikátor je aplikovaný na obraz segmentovaný podľa farby kože.

Spomenuté práce ukázali, že Viola&Jones detektor je vhodný na detekciu rúk. Nie všetky pózy sú však rozpoznané rovnako dobre. Podľa [8] je najlepšie rozpoznaná otvorená ruka so spojenými prstami. Pózy ako päšť a ukazovanie hore sú často klasifikátormi zamieňané. Detekcia pomocou spomenutej metódy je do určitej miery invariantná voči natočeniu. Podľa testov vykonaných v [8] je kvalita rozpoznania rúk ovplyvnená len veľmi málo, pokiaľ je rotácia tréningových a taktiež rotácia testovacích vzorov v rozmedzí 15 stupňov.

Metóda AdaBoost je primárne určená na rozlíšenie práve dvoch tried. Pokiaľ je potrebné klasifikovať do viacerých tried, je možné aplikovať paralelne niekoľko klasifikátorov [4], prípadne klasifikátory usporiadať do stromu [8].

Nevýhodou metódy AdaBoost je dlhý čas tréningovania. Horný odhad tréningovania a výberu jedného slabého klasifikátora založeného na Haarovom príznaku je obvykle $O(NT)$, kde N je počet vzorov a T je počet možných Haarových príznakov. To pre bežný počet niekoľko tisícov vzorov a niekoľko desiat tisícov príznakov predstavuje rádovo minúty. Tréningovanie celého klasifikátora tak na bežnom počítači zaberie niekoľko dní. Existuje niekoľko prác zaoberajúcich sa aspoň čiastočným odstránením tohto nedostatku. Metóda popísaná v článku [12] redukuje čas tréningovania slabého klasifikátora na $O(Nd^2 + T)$, kde d je počet pixelov vzoru. Toto je pre uvedené počty vzorov a príznakov v rádoch iba niekoľkých sekúnd.

2.1.5 Ďalšie metódy

Na rozpoznanie rúk je okrem doteraz spomenutých metód možné použiť tiež neurónové siete. Ich veľkou výhodou je schopnosť klasifikovať vstup do niekoľkých tried. Takže na rozpoznanie pózy nie je potrebné aplikovať paralelne niekoľko klasifikátorov ako napríklad v prípade metódy AdaBoost. Nevýhodou neurónových sietí je ich náchylnosť k pretrénovaniu, teda, že sa príliš dobre naučia dáta z tréningovej množiny a stratia schopnosť správne zobecňovať. Ďalšou nevýhodou neurónových sietí je podobne ako pri metóde AdaBoost dlhý čas tréningovania.

Úplne odlišný prístup k sledovaniu a detekcii rúk reprezentuje trieda metód založených na 3D geometrickom modeli ruky. Metóda popísaná v správe [13] je schopná zo sekvencie snímok jednej kamery odhadnúť 3D konfiguráciu ruky. Metóda definuje štruktúrally a kinematický model ruky, kde články prstov a dlaň sú považované za pevné objekty konštantnej veľkosti navzájom spojené v kĺboch. Medzi týmito objektmi je pomocou pravdepodobností definovaných niekoľko typov obmedzení. Kinematické obmedzenie definuje možnosti rotácie jednotlivých kĺbov. Štruktúralne obmedzenie zabezpečuje, že jednotlivé časti sa v 3D priestore neprekrývajú a časové obmedzenie rieši súvislosť aktuálneho stavu s predchádzajúcim stavom. Metóda v obraze sleduje hrany a pravdepodobnosť farby kože a na základe týchto údajov upravuje pravdepodobnosť aktuálneho stavu ruky. Výsledky ukázali, že metóda dokáže presne odhadnúť a sledovať 3D konfiguráciu ruky. Bohužiaľ zatiaľ ju nie je

možné použiť pre potreby interaktívneho ovládania počítača, pretože spracovanie jedného snímku trvá rádovo niekoľko sekúnd až minút.

2.2 Adaptácia klasifikátorov

Predstavme si situáciu, že máme na základe veľkého množstva dát dobre natrénovaný klasifikátor. Chceme ho ale použiť v novom alebo špecifickejšom prostredí (nová kamera, zmena osvetlenia, nový užívateľ, atď.). Klasifikátor potrebujeme prispôbiť novým podmienkam pomocou malého množstva adaptačných dát.

2.2.1 MAP adaptácia

MAP adaptácia je metóda používaná na adaptáciu klasifikátorov, ktoré modelujú pravdepodobnostné rozloženia tried. Podrobný popis metódy a odvodenie vzťahov je možné nájsť v [1]. Princíp spočíva v tom, že parametre Θ modelu sa riadia pravdepodobnostným rozložením $p(\Theta)$. Toto rozloženie nemusí byť rovnomerné, teda niektoré parametre sú viac pravdepodobné ako iné. Rozloženie pravdepodobnosti parametrov odhadujeme na základe natrénovaného modelu.

Pre dané adaptačné dáta $\mathbf{X}$ hľadáme:

$$\hat{\Theta} = \arg \max_{\Theta} p(\Theta|\mathbf{X}) \quad (2.4)$$

kde

$$p(\Theta|\mathbf{X}) \propto p(\mathbf{X}|\Theta)p(\Theta) \quad (2.5)$$

$$\propto \prod_{k=1}^N p(x_k|\Theta)p(\Theta) \quad (2.6)$$

Predpokladajme, že máme natrénovaný model určený normálnym rozložením hustoty pravdepodobnosti s parametrami μ_{model} , σ_{model}^2 . Chceme adaptovať strednú hodnotu μ , ktorej rozloženie hustoty pravdepodobnosti je dané funkciou $p(\mu) = \mathcal{N}(\mu, \mu_{param}, \sigma_{param}^2)$. Výsledný rozptyl σ^2 budeme považovať za daný. Maximalizovaním vzťahu $p(\mu|\mathbf{X})$ a aplikovaním zjednodušenia $\sigma_{param}^2 = \frac{\tau}{\sigma^2}$, $\mu_{param} = \mu_{model}$ dostaneme:

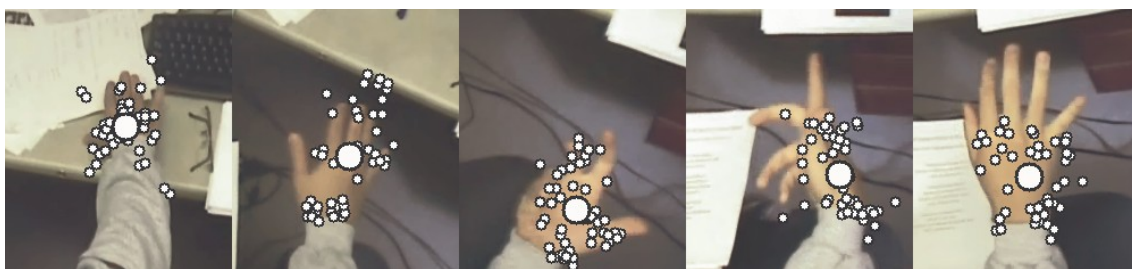
$$\hat{\mu} = \frac{\sum_{k=1}^N x_k + \tau \mu_{model}}{N + \tau} \quad (2.7)$$

Vzťah 2.7 je možné interpretovať tak, že pôvodne natrénovaný model je chápaný ako τ hodnôt, ktoré spolu so získanými adaptačnými dátami slúžia na nový odhad parametrov. Na základe tejto myšlienky sa potom dá odvodiť vzťah 2.8 pre adaptáciu rozptylu alebo tiež adaptácia histogramu.

$$\hat{\sigma}^2 = \frac{(\sigma_{model}^2 + \mu_{model}^2)\tau + \sum_{k=1}^N x_k^2}{\tau + N} - \hat{\mu}^2 \quad (2.8)$$

2.3 Metódy sledovania pohybu

Veľká variabilita vzhľadu ruky nie je problém len pri jej detekcii, ale aj pri sledovaní jej pohybu. Táto kapitola popisuje niekoľko metód sledovania pohybu, ktoré je možné aplikovať na ruky.



Obr. 2.7: Princíp metódy Flocks of Features. Malé krúžky reprezentujú jednotlivé príznaky. Jeden veľký krúžok značí polohu sledovaného objektu. Obrázok je prevzatý z [8].

2.3.1 Flocks of Features

Metóda sledovania pohybu rúk s názvom *Flocks of Features* bola predstavená v práci [8]. Táto metóda je schopná sledovať rýchlo sa pohybujúce a veľmi variabilné objekty bez potreby poznať ich model. Princíp spočíva v sledovaní veľkého množstva jednoduchých príznakov na povrchu objektu. Výsledná poloha objektu je určená ako medián polôh jednotlivých príznakov.

Počas sledovania sú v každom kroku najskôr nezávisle na sebe aktualizované polohy jednotlivých príznakov. Následne je požadované, aby boli príznaky navzájom dostatočne vzdialené. Druhá požiadavka vynucuje, aby vzdialenosť príznakov od ich mediánu bola pod určenou hranicou. Príznaky, ktoré tieto podmienky nespĺňajú sú nahradené novými. Nové príznaky sú vytvárané na miestach s vysokou pravdepodobnosťou farby kože. Model farby kože je získavaný dynamicky.

Výhodou metódy Flocks of Features je jej jednoduchosť a rýchlosť. Metóda je tiež odolná voči pohybu kamery a šumu na pozadí. Medzi nevýhody patrí, že objekt musí byť dostatočne veľký, aby sa na jeho povrchu našlo dostatočné množstvo príznakov. Za ďalšiu nevýhodu je možné považovať, že získaná poloha nie je ťažiskom sledovaného objektu, ale sa voľne pohybuje po jeho povrchu.

2.3.2 Sledovanie pomocou kontúry

Metódy sledovania pohybu pomocou kontúry reprezentujú objekt krivkou, ktorá často avšak nie nevyhnutne leží na hranici tohto objektu. Jedným zo zástupcov tejto triedy metód sú aktívne kontúry. Tieto fungujú na princípe minimalizácie energie krivky, ktorá závisí na tvare krivky a na veľkosti gradientu v obraze pod krivkou.

Inou metódou založenou na kontúrach je algoritmus CONDENSATION predstavený v článku [6]. Jeho autormi sú Michael Isard a Andrew Blake. Objekt je reprezentovaný jednou spline krivkou, ktorú je možné popísať obmedzeným počtom parametrov. Na začiatku spracovania každého snímku algoritmom CONDENSATION je známe rozloženie hustoty pravdepodobnosti parametrov krivky. Toto rozloženie je navzorkované tak, že tam kde je jeho hodnota vyššia, sa generuje viac vzorkov ako tam, kde je nižšia. Na základe modelu sledovaného objektu potom jednotlivé vzorky podstúpia dva predikčné kroky. V prvom kroku sú posunuté deterministicky a v druhom nedeterministicky. Nakoniec sú vzorkom na základe nájdenia príznakov v spracovávanom snímku priradené váhy. Takto váhované vzorky reprezentujú rozloženie hustoty pravdepodobnosti parametrov krivky reprezentujúcej sledovaný objekt v danom snímku.

Autori metódy CONDENSATION demonštrovali jej použitie aj na sledovaní otvorenej ruky, kde sa prsty mohli hýbať nezávisle na sebe. Toto použitie je znázornené na obrázku 2.8.



Obr. 2.8: Sledovanie otvorenej ruky na zložitom pozadí pomocou metódy CONDENSATION. Obrázky sú prevzaté z [6].

Metóda vyzerá byť vhodná pre sledovanie objektov na zložitom pozadí. Avšak kvôli vysokej variabilite ruky môže byť zložité odhadnúť parametre modelu, podľa ktorých prebiehajú predikčné kroky. Problém býva tiež s inicializáciou metódy, ale v prípade sledovania rúk sa dá použiť napríklad inicializácia na základe segmentácie podľa farby kože. Tento spôsob inicializácie bol použitý v rámci diplomovej práce [18], kde bol algoritmus CONDENSATION aplikovaný na sledovanie jednej konštantnej pózy. Konkrétne sa jednalo o pózu ukazovania kolmo na kameru tak, aby bol v obryse viditeľný ukazovák.

2.4 Rozpoznávanie dynamických gest

Na základe pohybu ruky je možné rozpoznávať dynamické gestá. Dĺžka trvania gesta sa môže meniť, preto je potrebné na rozpoznávanie použiť techniku, ktorá je schopná spracovávať sekvencie vstupov rôznych dĺžok. Takýmito technikami sú napríklad konečný automat alebo skrytý Markovov model.

2.4.1 Skryté Markovove modely

Skrytý Markovov model (Hidden Markov Model – HMM) je určený množinou stavov S , maticou prechodových pravdepodobností medzi stavmi A a množinou možných pozorovaní O . Pre každý stav $s \in S$ je určená jeho počiatočná pravdepodobnosť p_{init} a vysielacia pravdepodobnosť $p(o)$ pre každé možné pozorovanie $o \in O$.

Na vyhodnotenie HMM sa používa Viterbiho algoritmus alebo veľmi podobný takzvaný dopredný (forward) algoritmus. Vstupom vyhodnotenia HMM je sekvencia pozorovaní. Podľa jednotlivých pozorovaní sa v závislosti od aktuálnej pravdepodobnosti stavov, prechodových a vysielacích pravdepodobností vyhodnotí nová pravdepodobnosť stavov. Na základe spracovania celej sekvencie je možné určiť pravdepodobnosť, že táto sekvencia pozorovaní bola vygenerovaná daným modelom, alebo tiež najpravdepodobnejšiu sekvenciu stavov, ktorá by generovala takéto pozorovania.

Skryté Markovove modely sa používajú na rozpoznávanie ručne písaného textu, gestá myši alebo v spracovaní reči. Viac o skrytých Markovových modeloch je možné nájsť v [16, 2].

Kapitola 3

Navrhnuté rozhranie pre ovládanie počítača rukami

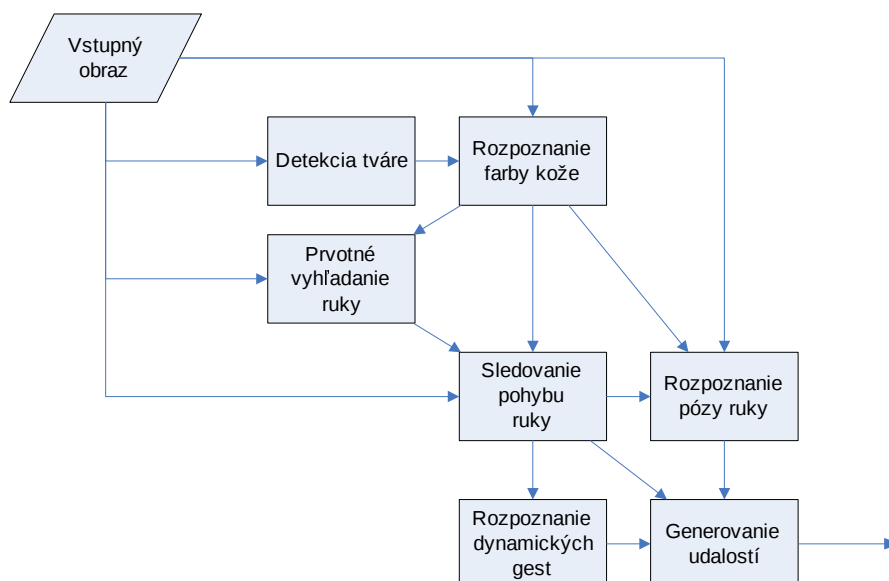
Táto kapitola obsahuje návrh počítačového rozhrania založeného na detekcii a sledovaní rúk v obraze. Implementácia rozhrania a jeho použitie v demonštračnej aplikácii je popísaná v nasledujúcej kapitole s číslom 4.

Navrhnuté rozhranie sleduje ruky pomocou jednej kamery, ktorá je umiestnená v statickej polohe pred užívateľom v blízkosti monitora prípadne plátna. Takéto umiestnenie je zvolené z dôvodu, aby bolo možné rozhranie použiť aj na notebookoch, kde je kamera štandardne na vrchnom okraji displeja. Ďalšou výhodou tejto polohy je možnosť súčasného sledovania pozície hlavy a rúk. Navrhnuté rozhranie nevyžaduje umiestnenie žiadnych špeciálnych značiek na rukách užívateľa. Farebné alebo reflexné rukavice prípadne značky by výrazne zjednodušili detekciu a sledovanie rúk, na druhú stranu by ale znižovali prirodzenosť ovládania. V súčasnej verzii je kvôli zjednodušeniu systému sledovaná iba pravá ruka, ale do budúcnosti bude možné rozhranie rozšíriť o sledovanie oboch rúk.

Užívateľ začne svoju prácu so systémom vložením ruky do záberu kamery. Ruka musí určitý čas nehybne zotrvať v požadovanej póze. Potom ju rozhranie začne sledovať. Na základe jej pohybu, rozpoznania rôznych póz alebo dynamických gest sú generované udalosti, ktoré môže ovládaná aplikácia rôznym spôsobom interpretovať. Práca s rozhraním skončí, keď užívateľ prejde so sledovanou rukou mimo záber kamery.

Princíp činnosti rozhrania je znázornený na obrázku 3.1. Na základe detekcie tváre vo vstupnom obraze je adaptovaný model farby kože. Rozpoznanie farby kože slúži ako pomocné kritérium pre detekciu rúk, sledovanie ich pohybu a rozpoznanie póz. Po detekcii ruky v obraze je inicializované jej sledovanie. Na základe pohybu ruky sú rozpoznávané dynamické gestá. Pre sledovanú ruku je tiež určená jej póza. Poslednou časťou systému je generátor udalostí. Ten na základe výstupov z ostatných častí systému vytvára správy, podľa ktorých ovládaná aplikácia riadi svoju činnosť.

Prirodzeným požiadavkom na rozhranie je minimalizácia nárokov na výpočtový výkon. Nestačí totiž, aby rozhranie bolo natoľko efektívne, že dokáže bežať v reálnom čase, teda aby dokázalo spracovať niekoľko snímkov za sekundu. Je nutné, aby na danom počítači okrem samotného rozhrania bežala minimálne ešte ovládaná aplikácia. Ďalším dôvodom redukcie výpočtových nárokov je znižovanie spotreby elektrickej energie. Toto je obzvlášť dôležité u prenosných zariadení pracujúcich na batériu.



Obr. 3.1: Princíp činnosti rozhrania. Vstupom je obraz z kamery, výstupom sú udalosti riadiace činnosť ovládanej aplikácie.

3.1 Adaptívny model farby kože

Farba kože v rozhraní slúži ako pomocné kritérium hlavne pre sledovanie pohybu, ale taktiež pre detekciu rúk, rozpoznanie pózy. Aby bolo rozpoznanie kože kvalitné za rôznych svetelných podmienok, pri použití rôznych kamier a pre rôznych užívateľov, je vhodné použiť model, ktorý sa bude adaptovať počas behu systému.

Kvôli efektívnosti výpočtu a nezávislosti na intenzite osvetlenia bol na parametrizáciu farby zvolený normalizovaný RG priestor. Výpočet jeho zložiek je uvedený v sekcii 2.1.2 v rovniciach 2.3 a 2.2.

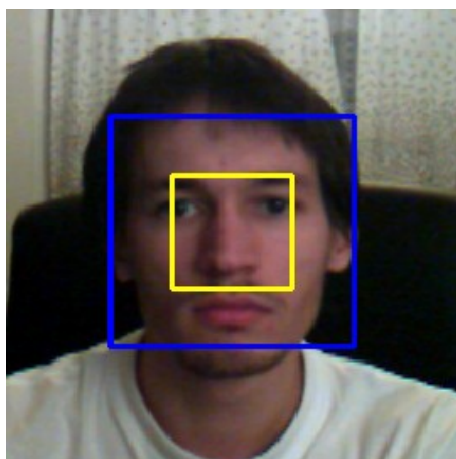
Model farby kože pozostáva z dvoch tried: koža a pozadie. Pre tieto triedy sú definované rozloženia hustoty pravdepodobností $p(x|skin)$ a $p(x|background)$ a apriórne pravdepodobnosti $P(skin)$ a $P(background)$. Výsledná pravdepodobnosť, že nejaký pixel je koža sa vypočíta z Bayesovho vzorca:

$$P(skin|x) = \frac{p(x|skin)P(skin)}{p(x|skin)P(skin) + p(x|background)P(background)} \quad (3.1)$$

Model môže byť natrénovaný dopredu, ale počas behu systému sa adaptuje na základe detekcie tváre vo vstupnom obraze. Obdĺžnik predstavujúci výstup z detektora tváre je zmenšený, pričom ostáva zachovaný jeho stred. Všetky pixely vnútri zmenšeného obdĺžnika sú použité na adaptáciu rozloženia hustoty pravdepodobnosti farby kože. Princíp adaptácie na základe detekcie tváre je znázornený na obrázku 3.2.

Ďalším zdrojom dát pre triedu farby kože je oblasť sledovanej ruky. Podobne ako pri tvári sú použité pixely v strede tejto oblasti. Tieto dáta sú však menej kvalitné v porovnaní s tými, získanými z tváre. Hlavným dôvodom je, že ruka je oveľa variabilnejšia ako tvár a nie je možné určiť časť sledovanej oblasti, v ktorej by sa vždy nachádzala iba koža a nikdy žiadne pozadie. Ďalším dôvodom horšej kvality týchto dát je nedokonalosť detekcie a sledovania ruky.

Za farbu pozadia sú považované úplne všetky pixely obrázka. Do pozadia je teda zahrnutá aj samotná tvár a taktiež oblasti ako krk alebo ruky, ktoré majú farbu kože. Ale vzhľadom



Obr. 3.2: Adaptácia rozloženia hustoty pravdepodobnosti farby kože na základe detekcie tváre. Vonkajší obdĺžnik predstavuje výstup detektora tváre, vnútorný ohraničuje pixely použité na adaptáciu.

na malú plochu týchto oblastí v porovnaní s celým obrazom je ich vplyv na kvalitu modelu nevýznamný. Bolo by tiež možné zvoliť obmedzenie, že iba pixely, ktoré sa nenachádzajú v detekovaných tvárach a rukách sú pixely pozadia. Prípadne by sa tiež dalo použiť učenie bez učiteľa.

V modeli farby kože sa adaptujú iba rozloženia hustôt pravdepodobností. Apriórne pravdepodobnosti tried sú konštantne nastavené na hodnotu 0,5. Na adaptáciu sa využívajú všetky dovtedy získané dáta s tým, že staršie dáta majú nižšiu váhu. Váha konkrétnych dát sa vypočíta podľa vzťahu $w = r^{age}$, kde w je výsledná váha, r je konštanta v rozsahu 0 až 1 ovplyvňujúca rýchlosť zabúdania a age vyjadruje počet snímok, ktoré uplynuli od získania daných dát. Takže pokiaľ boli dáta získané v aktuálnom snímku, je hodnota age rovná 0, a teda ich váha je 1. Popísaný spôsob určovania váh je možné efektívne implementovať tak, že váhy všetkých adaptačných vzorkov sú po spracovaní aktuálneho snímku pre násobené hodnotou r .

Pre triedy navrhnutého adaptívneho modelu farby kože existuje niekoľko možností reprezentácie rozloženia hustoty pravdepodobnosti:

- Normálne rozloženie reprezentované 2D Gaussovou funkciou.
- Gaussian Mixture Model (GMM) – váhovaná suma niekoľkých 2D Gaussových funkcií.
- Normalizovaný 2D histogram.

Normálne rozloženie dobre generalizuje, ale je vhodné iba pre triedu farby kože. Pre triedu pozadia by sa jednalo o príliš veľké zjednodušenie. GMM je pravdepodobne použiteľný na reprezentáciu rozloženia hustoty pravdepodobnosti oboch tried. Jeho nevýhodou oproti ostatným spomenutým reprezentáciám je však vyššia výpočtová a implementačná zložitosť. Normalizovaný histogram je podobne ako GMM vhodný na reprezentáciu oboch tried.

Z uvedených dôvodov bol na reprezentáciu pozadia zvolený normalizovaný 2D histogram. Na reprezentáciu triedy farby kože bolo experimentované s normálnym rozložením a histogramom. Na základe pozorovania, ktorého výsledky sú uvedené v sekcii 5.1, bola nakoniec aj pre kožu zvolená reprezentácia histogramom. Histogram a normálne rozloženie dosahujú pri dobrom osvetlení podobné výsledky, ale histogram lepšie zvláda presvetlené oblasti kože.

Čo sa týka spôsobu adaptácie modelu farby kože, máme opäť niekoľko možností:

- Natrénovať model dopredu a počas behu systému už neadaptovať.
- Kompletne pretrénovať model, keď sú k dispozícii nové dáta.
- Použiť MAP adaptáciu.

Z testov uvedených v sekcii 5.1 vyplynulo, že adaptácia počas behu systému má výrazný vplyv na kvalitu rozpoznania kože. Bola zvolená adaptácia kompletným pretrénovaním modelu, pretože je oproti MAP adaptácii implementačne aj výpočtovo menej náročná a vzhľadom na dostatočné množstvo kvalitných adaptačných dát dosahujú obe metódy podobné výsledky. Ďalšou veľkou výhodou adaptácie kompletným pretrénovaním je, že odpadá nutnosť trénovať model pred spustením systému.

3.2 Detekcia tváre

Účelom detekcie tváre v rozhraní je adaptácia modelu farby kože, ktorý je popísaný v sekcii 3.1. Na samotné vyhľadávanie tváre bol zvolený Viola&Jones detektor, ktorý je stručne predstavený v sekcii 2.1.4. Dôvodom tejto voľby je dostupnosť takéhoto detektora v rámci knižnice OpenCV, a taktiež jeho rýchlosť a vysoká úspešnosť.

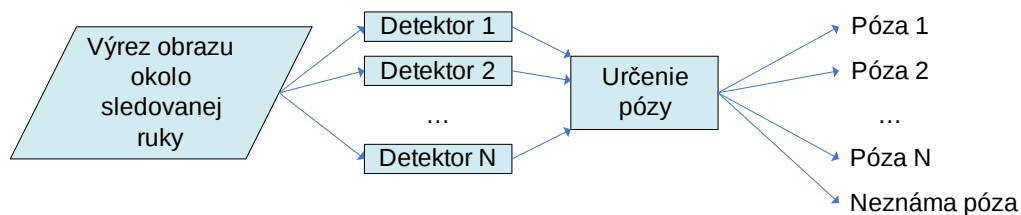
Pre zníženie výpočtových nárokov vyhľadávania tváre vo vstupnom obraze je možné použiť niektorý z nasledujúcich prístupov:

- Na základe sledovania pohybu aplikovať detektor iba v okolí predpokladaného výskytu hlavy.
- Aplikovať detektor iba raz za niekoľko snímkov. Nevýhodami tohto prístupu je nárazové zafarbovanie procesora a zníženie množstva adaptačných dát pre detektor kože. Veľkou výhodou tohto prístupu je jednoduchosť jeho implementácie.
- Rozdeliť obraz na niekoľko oblastí a v každom snímku prehľadávať iba jednu oblasť. Nedostatkami tejto metódy je obmedzenie veľkosti detekovanej tváre rozmermi prehľadáwanej oblasti. Ďalším problémom je, že pokiaľ sa tvár nachádza na hranici dvoch oblastí, tak nie je nájdená, lebo neleží ani v jednej z nich. Tento problém by ale bolo možné čiastočne riešiť prekrývaním oblastí. Podobne ako pri predchádzajúcej metóde dochádza k zníženiu množstva adaptačných dát pre detektor kože.
- Približne rovnomerne rozdeliť všetky možné prehľadávané pozície v obraze do skupín a v každom snímku prehľadávať iba pozície z jednej skupiny. Prehľadávané pozície by sa dali rozdeliť napríklad podľa veľkosti. Opäť dochádza k zníženiu množstva adaptačných dát.

Z dôvodu jednoduchosti implementácie a iba malým nedostatkom oproti ostatným prístupom bola zvolená druhá možnosť, teda vyhľadávať tvár len raz za niekoľko snímkov.

3.3 Detekcia rúk a rozpoznanie pózy

Rovnako ako na vyhľadávanie tváre je aj na rozpoznanie rúk zvolený Viola&Jones detektor. Rozhranie využíva niekoľko detektorov, z ktorých každý je natrénovaný na vyhľadávanie inej pózy. Je určená jedna, takzvaná inicializačná póza. Rozhranie vyhľadáva túto pózu v obraze, a keď ju nájde, spustí sa sledovanie danej ruky a užívateľ môže začať pracovať. Pre zníženie výpočtových nárokov je inicializačná póza vyhľadávaná len raz za niekoľko



Obr. 3.3: Rozpoznávanie pózy sledovanej ruky. Detektory jednotlivých póz sú paralelne aplikované v oblasti okolo sledovanej ruky.

snímkov, podobne ako je to pri detekcii tváre. Aby sa však minimalizoval dopad nárazovej záťaže, je ruka vždy vyhľadávaná v inom snímku ako tvár. Ak si určíme vyhľadávací interval ako malé kladné celé číslo t , potom tvár je vyhľadávaná v snímkoch $i_{face} \in \{t, 3t, 5t, 7t, \dots\}$ a inicializačná póza v snímkoch $i_{hand} \in \{2t, 4t, 6t, \dots\}$.

Princíp rozpoznávania pózy sledovanej ruky je znázornený na obrázku 3.3. V oblasti okolo sledovanej ruky sú paralelne aplikované detektory jednotlivých póz. Najväčšia detekcia určuje výslednú pózu. V prípade, že v oblasti nie je žiadna detekcia, je póza označená ako neznáma. Aby došlo k zmene pózy, musí byť nová póza rozpoznaná v niekoľkých po sebe idúcich snímkoch po stanovenú minimálnu dobu. To zabezpečí, že nedochádza k príliš častým zmenám pózy spôsobenými napríklad nedokonalosťou detekcie.

Navrhnuté usporiadanie detektorov je výhodné v tom, že je možné ľubovoľne nezávisle na sebe pridávať a odoberať rozpoznávané pózy. Z pohľadu výpočtových nárokov je ale vhodné len pre malý počet póz. Pre väčší počet by bolo lepšie detektory usporiadať napríklad do stromu.

Pre zníženie počtu falošných detekcií je pred aplikáciou detektorov rúk obraz z kamey vymaskovaný farbou kože. Je však nutné zabezpečiť kvalitné vymaskovanie obrazu tak, aby neboli odstránené pixely z rúk. Inak by tieto ruky nemuseli byť nájdené.

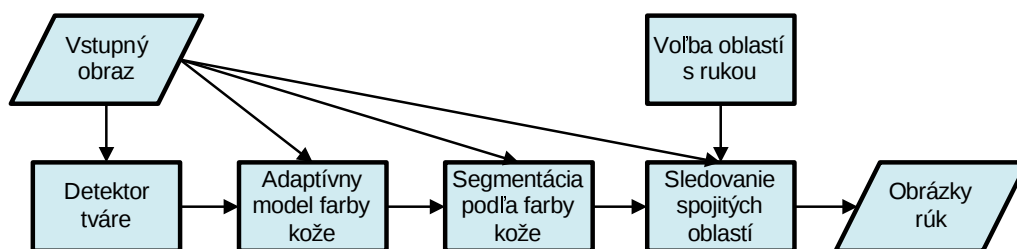
3.4 Automatické získavanie trénovacích dát pre detektory rúk

Na trénovanie detektorov rúk je potrebné nazbierať veľké množstvo pozitívnych a negatívnych príkladov. Negatívnych príkladov je možné získať takmer neobmedzené množstvo ako náhodné výrezy z obrázkov neobsahujúcich ruky. Pozitívne príklady sa dajú zaobstarať napríklad manuálnou anotáciou dát. Toto je však veľmi časovo náročné. Z tohto dôvodu je potrebné navrhnuť metódu, ktorá by proces získavania pozitívnych príkladov do značnej miery automatizovala.

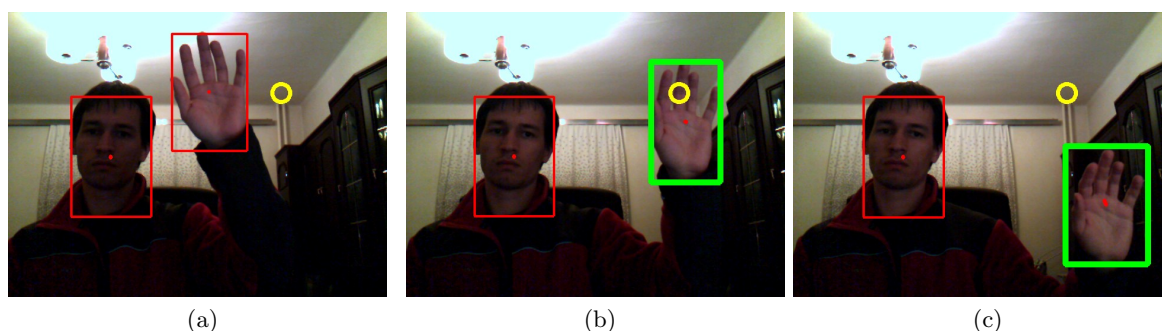
Navrhnutá metóda automatického získavania trénovacích dát využíva skutočnosť, že farba rúk je podobná farbe tváre, a pritom kvalitné detektory tváre sú voľne dostupné¹.

Princíp navrhutej metódy je znázornený na obrázku 3.4. Vstupom je obraz z kamery. V tomto obraze je detekovaná tvár a na základe jej pozície je adaptovaný model farby kože. Podľa tohto modelu je potom segmentovaný vstupný obraz a sú sledované všetky dostatočne veľké spojené oblasti s farbou kože. Užívateľ zabezpečí výber oblastí obsahujúcich ruku. Pozície zvolených oblastí potom môžu byť použité na anotáciu vstupného obrazu, avšak

¹Napríklad v rámci knižnice OpenCV je dostupných niekoľko natrénovaných Viola&Jones detektorov tváre.



Obr. 3.4: Princíp metódy automatického získavania trénovacích dát. Vstupom je obraz z kamery a výstupom sú trénovacie príklady pre detektor rúk.



Obr. 3.5: Spôsob aktivácie oblastí s rukou. (a) Na začiatku je oblasť neaktívna. (b) Po jej umiestnení na zvolený bod v obraze sa aktivuje. (c) Vďaka sledovaniu ostane oblasť aktívna, aj keď opustí daný bod.

z dôvodu redukcie veľkosti ukladaných dát je vhodnejšie ukladať len výrezy zo vstupného obrazu určené pozíciou daných oblastí.

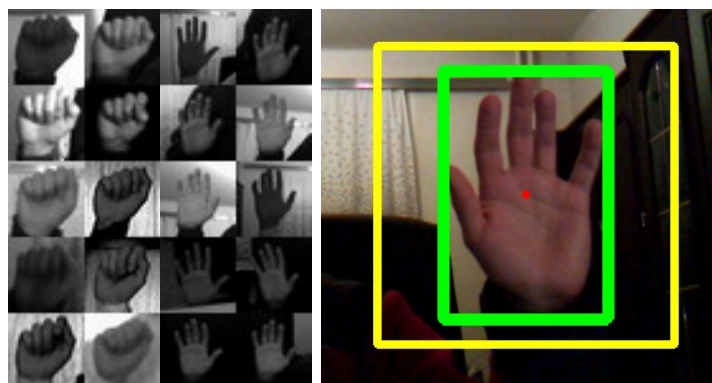
Metóda využíva adaptívny model farby kože popísaný v sekcii 3.1. Trieda kože je adaptovaná len na základe detekovanej tváre. Za pozadie sú považované všetky pixely mimo tvár.

Korešpondencia oblastí v dvoch po sebe nasledujúcich snímkoch je určená na základe ich vzdialenosti. Bolo by možné využiť aj lepšie metódy korešpondencie, prípadne dokonalejšie metódy sledovania, avšak tento spôsob je dostatočný.

Nie všetky sledované oblasti farby kože obsahujú ruku. Preto má každá oblasť priradený príznak aktivity. Ten určuje, či je oblasť považovaná za ruku alebo nie, a teda, či budú na základe jej pozície získavané trénovacie dáta. Na začiatku sú všetky oblasti neaktívne. Oblasť je aktivovaná, keď prejde cez zvolený bod v obraze. Toho užívateľ dosiahne umiestnením ruky na dané miesto, tak ako je to znázornené na obrázku 3.5. Vďaka sledovaniu ostane oblasť aktívna, aj keď opustí toto miesto. Tiež by sa dali oblasti aktivovať napríklad kliknutím myši, avšak to by znížilo mieru automatizácie metódy.

Poloha aktívnych oblastí nie je na anotáciu obrazu, prípadne určenie výrezov, použitá priamo. Okolo každej aktívnej oblasti je vytvorený obaľujúci obdĺžnik, ktorý má potrebný pomer strán. Tento obdĺžnik je následne zväčšený, aby na okrajoch vzniklo voľné miesto okolo ruky. Takto pripravené dáta je potom možné jednoducho zmenšiť na veľkosť potrebnú pre trénovanie detektora. Na obrázku 3.6 je ukážka výslednej oblasti určujúcej ukladaný výrez, a taktiež príklady získaných trénovacích dát.

Navrhnutá metóda umožňuje získavať niekoľko trénovacích príkladov za sekundu. To v praxi znamená, že za niekoľko málo hodín sa dajú nazbierať tisíce trénovacích príkladov.



Obr. 3.6: Ukážky výstupov metódy automatického získavania tréningových dát. Na obrázku vpravo je ukázaný princíp získania výslednej ukladanej oblasti. Vnútorňý obdĺžnik predstavuje sledovanú oblasť farby kože a vonkajší obdĺžnik je výsledný ukladaný výrez s potrebným pomerom strán.

Tieto, ako ukázali testy uvedené v sekcii 5.2, stačia na natrénovanie dobre použiteľných detektorov rôznych póz ruky.

Ďalšie zlepšenie kvality tréningových príkladov by bolo pravdepodobne možné dosiahnuť ich zarovnaním. Toto však zatiaľ nebolo otestované. Princípy použité v predstavenej metóde je možné aplikovať taktiež na získavanie tréningových dát pre iné objekty ako ruky.

3.5 Sledovanie pohybu

Na sledovanie pohybu ruky je použitý upravený algoritmus Flocks of Features. Úpravy algoritmu spočívajú v lepšej schopnosti vysporiadať sa so zmenou veľkosti objektu a v použití detekcie ruky na korekciu pozície. Presný popis pôvodného algoritmu je možné nájsť v práci [8]. Pôvodný algoritmus je tiež veľmi stručne predstavený v sekcii 2.3.1. Algoritmus Flocks of Features bol vybraný z dôvodu, že podľa demonštrácie autorov je schopný sledovať rýchlo sa pohybujúce a značne deformovateľné objekty akými sú aj ruky. Ďalšími jeho výhodami sú, že nevyžaduje model sledovaného objektu a je výpočtovo nenáročný.

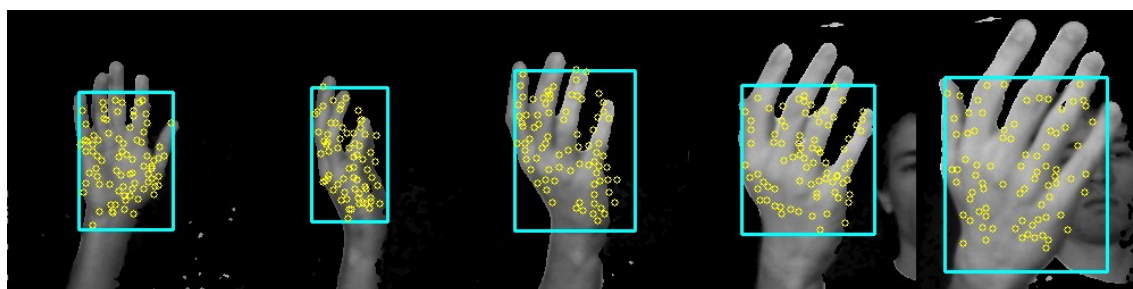
Bola tiež snaha použiť aktívne kontúry. Tie ale neboli schopné ruku spoľahlivo sledovať. Na sledovanie rúk na zložitom pozadí sa zdá byť vhodná metóda CONDENSATION stručne popísaná v sekcii 2.3.2. Tá však vyžaduje model sledovaného objektu. Takýto model môže byť pre ruku veľmi ťažké získať vzhľadom na veľkú variabilitu jej tvaru. Na druhú stranu by v prípade použitia metódy CONDENSATION na sledovanie ruky mohlo byť oveľa jednoduchšie rozpoznávanie pózy, keďže by bola známa jej kontúra.

Vstupmi použitého algoritmu sledovania sú šedotónový obraz vymaskovaný oblasťami kože a obraz predstavujúci pravdepodobnosť farby kože. Oba tieto obrazy sú získané na základe vstupného snímku z kamery pomocou adaptívneho modelu farby kože. Ukážka vstupov je na obrázku 3.7.

Princíp použitého algoritmu spočíva v sledovaní rádovo desiatok jednoduchých príznakov rozmiestnených na povrchu ruky, tak ako je to znázornené na obrázku 3.8. V každom snímku sú pomocou výpočtu optického toku tieto príznaky nezávisle na sebe sledované. Pre niektoré príznaky môže sledovanie zlyhať, ale z tých ktoré zostanú je vypočítaný medián. Je vytvorený obdĺžnik obaľujúci 90 percent príznakov, ktoré sú najbližšie k mediánu. To znamená, že pre výpočet obdĺžnika je zanedbaných 10 percent najvzdialenejších príznakov. Tento obdĺžnik je



Obr. 3.7: Vstupy pre algoritmus sledovania: (a) šedotónový obraz z kamery vymaskovaný farbou kože, (b) obraz predstavujúci pravdepodobnosť farby kože.



Obr. 3.8: Sekvencia obrázkov demonštrujúcich sledovanie ruky vo vymaskovanom snímku. Malé krúžky predstavujú pozície príznakov. Obdĺžnik reprezentuje sledovanú oblasť ruky. Obrázky boli získané vyrezaním oblastí 200x250 pixelov z pôvodného snímku s rozlíšením 640x480. Je možné si všimnúť, že oblasť ruky mení podľa potreby svoju veľkosť.

následne mierne zväčšený, aby bolo kompenzované prípadné zanedbanie príznakov ležiacich na ruke. Výsledný obdĺžnik predstavuje oblasť sledovanej ruky.

V ďalšom kroku sú odstránené všetky príznaky, ktoré ležia mimo túto oblasť. Vzhľadom na to, že pôvodný obaľujúci obdĺžnik bol zväčšený, nemusia byť odstránené všetky príznaky, ktoré boli pri výpočte obaľujúceho obdĺžnika zanedbané.

Následne sú odstránené príznaky, ktorých vzdialenosť od niektorého iného príznaku je menšia ako minimálna povolená vzdialenosť d_{min} .

Aby nebolo sledované zbytočne veľké množstvo príznakov je ich počet pre nasledujúci snímok vypočítaný na základe veľkosti sledovanej oblasti podľa vzťahu:

$$c = \min(c_{min} + \frac{s}{k \cdot d_{min}^2}, c_{max}) \quad (3.2)$$

kde c je výsledný počet, c_{min} a c_{max} označujú minimálny a maximálny počet, s je obsah plochy, d_{min} je minimálna vzdialenosť príznakov a k je nejaké malé kladné číslo, ktorého hodnota je rádovo v jednotkách. Potom sú podľa potreby v oblasti ruky vytvorené nové príznaky. Tie sú vždy vytvárané náhodne tak, že tam, kde je vyššia pravdepodobnosť, že daný pixel je koža, je aj vyššia pravdepodobnosť vytvorenia nového príznaku. Nie je braná do úvahy vzdialenosť vytváraného príznaku od už existujúcich. To ale nevádi, lebo príliš blízke príznaky budú hneď pri spracovaní nasledujúceho snímku nahradené.

Pokiaľ je v okolí sledovanej oblasti detekovaná ruka, je malá náhodne vybraná časť príznakov nahradená novými, ktoré sú vytvorené v oblasti detekcie. Mohlo by sa zdať, že by bolo lepšie nahrádzať iba príznaky, ktoré ešte nie sú v oblasti detekcie. To by ale znemožnilo, aby bola na základe detekcie sledovaná oblasť zväčšená, pretože všetky príznaky by sa už v oblasti detekcie nachádzali. Tiež nie je vhodné nahrádzať veľkú časť alebo dokonca všetky príznaky, pretože to kvôli nepresnosti detekcie rúk vedie k náhlým zmenám pozície sledovanej oblasti.

Inicializácia použitého algoritmu sledovania prebieha v dvoch krokoch. Na základe detekcie ruky je určená obdĺžniková oblasť. Potom náhodne na základe pravdepodobnosti kože je v tejto oblasti vytvorený potrebný počet príznakov.

Sledovanie je ukončené v prípade, keď sa nepodarilo na základe výpočtu optického toku sledovať dostatočné množstvo príznakov. Ďalším dôvodom pre ukončenie sledovania je, keď sa sledovaná oblasť dotkne kraja obrazu alebo, keď je sledovaná oblasť príliš malá.

Nevýhodou popísaného algoritmu je jeho neschopnosť vo všeobecnosti odhaliť zlyhanie sledovania. Príkladom toho môže byť situácia, keď sledovací algoritmus stratí ruku a začne sledovať iný objekt. Tento problém je nutné riešiť na vyššej úrovni rozhrania.

Navrhnuté rozhranie je schopné sledovať naraz neobmedzený počet rúk. Avšak v aktuálnej verzii sú udalosti, podľa ktorých cieľová aplikácia riadi svoju činnosť, generované vždy len na základe jednej, takzvane aktívnej, ruky.

Na základe detekcie rúk pri prehľadávaní vstupného snímku sú pre nájdené ruky vytvorené osobitné sledovače (trackers). Aby nebola náhodou nejaká ruka sledovaná viackrát, je sledovač vytvorený len pre detekcie, ktoré nie sú v blízkosti nejakého iného sledovača. Každý sledovač sa počas svojej existencie nachádza postupne v niekoľkých stavoch:

Inicializácia Sledovanie nie je aktívne a sledovač iba overuje, či sa ruka nachádza na danej pozícii v nasledujúcich snímkoch. Ak je ruka detekovaná dostatočne často, prejde sa do stavu čakania na aktiváciu. Ak ruka prestane byť na danej pozícii nachádzaná, je sledovač zrušený.

Čakanie na aktiváciu Detekcia ruky je na danej pozícii potvrdená a sledovač čaká, či bude aktivovaný. Sledovač je aktivovaný, pokiaľ v systéme ešte nie je aktívny iný sle-



Obr. 3.9: Bod reprezentujúci pozíciu ruky je znázornený malým krúžkom. Jeho poloha je odvodená od stredu spodnej časti sledovanej oblasti. Táto oblasť je znázornená obdĺžnikom.

dovač. Sledovač je tiež aktivovaný, pokiaľ v okolí momentálne aktívneho sledovača nebola po nejakú dobu detekovaná žiadna póza. Tento stav môže byť spôsobený tým, že algoritmus sledovania zlyhal a namiesto ruky sleduje iný objekt. Momentálne aktívny sledovač následne prejde do stavu ukončenia sledovania. Ak je v stave čakania na aktiváciu viac sledovačov naraz, je aktivovaný ten, ktorý bol vytvorený podľa detekcie s najväčšou plochou. Všetky sledovače v tomto stave, ktoré nie sú v aktuálnom snímku aktivované, sú zrušené.

Aktívne sledovanie V tomto stave sa sleduje pohyb ruky a sú rozpoznávané pózy a dynamické gestá. V súčasnej verzii rozhrania je v tomto stave vždy maximálne jeden sledovač.

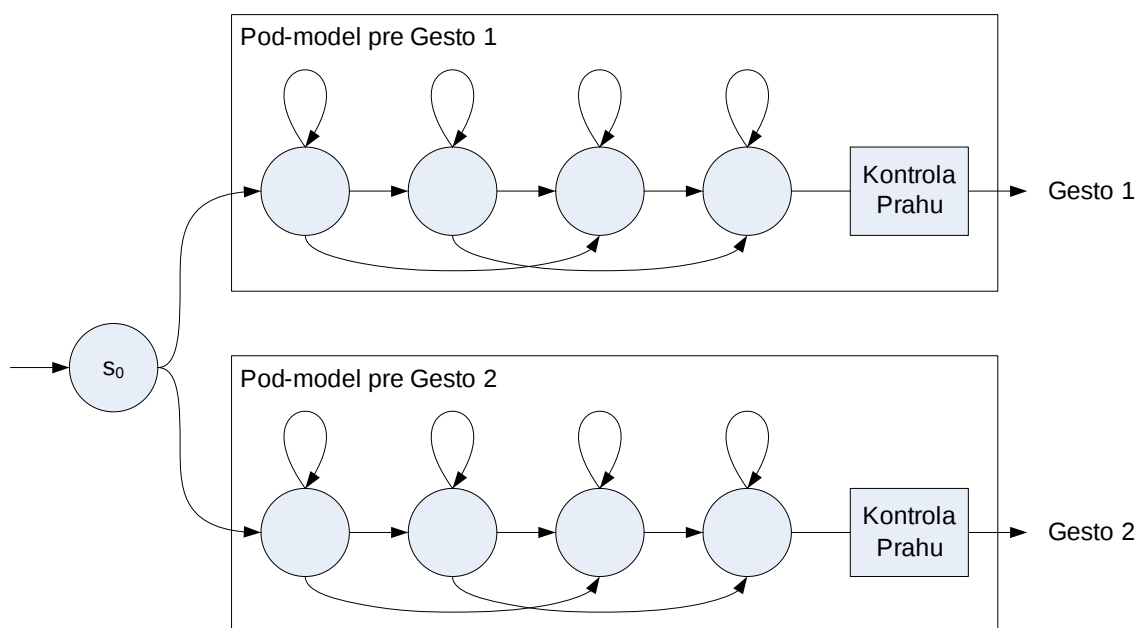
Ukončenie sledovania Do tohto stavu sa prejde v prípade detekcie zlyhania sledovania alebo, keď sa sledovaná oblasť dotkne kraja obrazu. Ďalším dôvodom je aktivácia iného sledovača. Sledovač je po prejdení do tohto stavu zrušený.

Pre interpretáciu pohybu ruky je výhodnejšie reprezentovať jej pozíciu ako bod namiesto obdĺžnika, ktorý je výstupom z algoritmu sledovania. K tomuto účelu bol zvolený bod v strede dolnej časti oblasti ruky tak, ako je to znázornené na obrázku 3.9. Táto pozícia bola vybraná, pretože je stále aj v prípade zmeny pózy. Aby sa kompenzovala nepresnosť sledovania, a aby bol pohyb výsledného bodu plynulejší, je na jeho pozíciu aplikovaný Kalmanov filter. Stav filtra je reprezentovaný polohou a rýchlosťou bodu. Predpokladá sa, že zrýchlenie je náhodné a riadi sa normálnym rozložením hustoty pravdepodobnosti.

3.6 Rozpoznanie dynamických gest

Na základe zisteného pohybu sledovanej ruky sú rozpoznávané dynamické gestá. Pre odlišenie gest od bežného pohybu je možné použiť niekoľko prístupov.

Pravdepodobne najjednoduchšie by bolo vyžadovať, aby počas celého trvania gesta bola ruka v určitej póze. Jednoduchosť spočíva v tom, že je známy začiatok aj koniec gesta a vyhodnotenie sa uskutoční pri prechode ruky do inej pózy. Tento prístup je však neprirodzený, pretože medzi dvomi po sebe nasledujúcimi gestami je nutná zmena pózy. Problémom tiež môže byť, že vzhľad ruky sa pri jej pohybe mení a póza by nemusela byť správne rozpoznaná počas celého vykonávania gesta. Nevýhodou je tiež potreba detektora ďalšej pózy ruky.



Obr. 3.10: Skrytý Markovov model rozpoznávajúci dve gestá. Model obsahuje špeciálny stav s_0 , na ktorý sú paralelne napojené pod-modely pre jednotlivé gestá. Gesto je rozpoznané, ak skóre tokenu v poslednom stave pre dané gesto dosiahne potrebnú hranicu.

Kvôli spomínaným problémom bol zvolený prístup, že gestá sú rozpoznávané neustále a bez ohľadu na pózu ruky. Gestá však musia byť zložitejšie ako pri predchádzajúcom prístupe, aby ich náhodné vykonanie pri bežnom pohybe bolo málo pravdepodobné. Ďalšie obmedzenie je, že žiadne gesto by nemalo byť časťou iného.

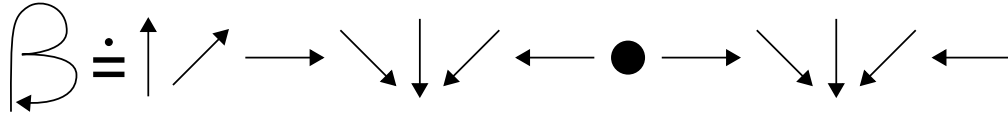
Na rozpoznávanie gest bol zvolený skrytý Markovov model (Hidden Markov Model – HMM). Ten sa osvedčil pri spracovaní reči na rozpoznávanie slov, čo je podobný problém ako rozpoznávanie gest.

Vstupom do rozpoznávača gest je v každom snímku zmena pozície ruky ($\Delta x, \Delta y$) a čas Δt , ktorý uplynul od spracovania predchádzajúceho vstupu. Na základe týchto vstupov je pre HMM vypočítaný vstupný vektor $\mathbf{o} = (v, \alpha)$, kde v je rýchlosť ruky v pixeloch za sekundu a α je uhol v radiánoch vyjadrujúci smer pohybu.

Všetky gestá sú rozpoznávané jediným modelom. Model obsahuje stav s_0 , na ktorý sú paralelne napojené pod-modely pre jednotlivé gestá. Výpočet pravdepodobností stavov je založený na Viterbiho token passing algoritme. Gesto je rozpoznané, ak skóre tokenu v poslednom stave pre dané gesto dosiahne potrebnú hranicu. Ukážka modelu pre dve gestá je na obrázku 3.10.

Token je definovaný svojim vekom age a logaritmickou pravdepodobnosťou p_{log} . Keďže sú modely gest oddelené, nie je nutné uchovávať postupnosť stavov, cez ktoré token prešiel. Pre každý token je stanovené jeho skóre, ktoré sa vypočíta ako priemerná pravdepodobnosť v čase podľa vzťahu $score = p_{log}/age$.

Na začiatku žiadny stav neobsahuje token. Pred spracovaním každého vstupného vektora $\mathbf{o}$ je do stavu s_0 vložený nový token ($p_{log} = 0, age = 0$). To je nutné z dôvodu, že nie je známy začiatok gesta. Takýmto spôsobom sa gestá začínajú rozpoznávať s každým vstupným vektorom. Každý token totiž zachytáva pravdepodobnosť generovania vstupných vektorov, ktoré prišli počas jeho existencie, stavmi, cez ktoré tento token prešiel.



Obr. 3.11: Príklad poskladania gesta zo základných zložiek. Šípky predstavujú pohyb v danom smere, krúžok predstavuje zotrvanie na mieste.

Spracovanie vstupného vektora $\mathbf{o}$ prebieha tak, že z každého stavu s_i , v ktorom je token, sa tento token presunie do každého napojeného stavu s_j . Tým dochádza k rozmnoženiu tokenu, pričom sa upravuje jeho pravdepodobnosť podľa vzťahu $p_{log}^{new} = p_{log} + \log(a_{ij}) + \log(p_j(\mathbf{o}))$, kde p_{log} je logaritmická pravdepodobnosť pôvodného tokenu, a_{ij} je prechodová pravdepodobnosť zo stavu s_i do stavu s_j a $p_j(\mathbf{o})$ je vysielacia pravdepodobnosť vstupu $\mathbf{o}$ stavom s_j . Pri presune je tiež zvýšený vek tokenu $age^{new} = age + 1$. Po presunutí všetkých tokenov zostane v každom stave iba token s najvyšším skóre a ostatné tokeny sa zrušia.

Ak skóre tokenu nachádzajúceho sa v poslednom stave niektorého gesta presiahne potrebnú hranicu, je toto gesto rozpoznané a všetky tokeny sa zrušia. Zrušenie tokenov je vykonané z dôvodu, aby sekvencia vstupov rozpoznaná ako gesto nebola znovu použitá na rozpoznanie ďalšieho gesta.

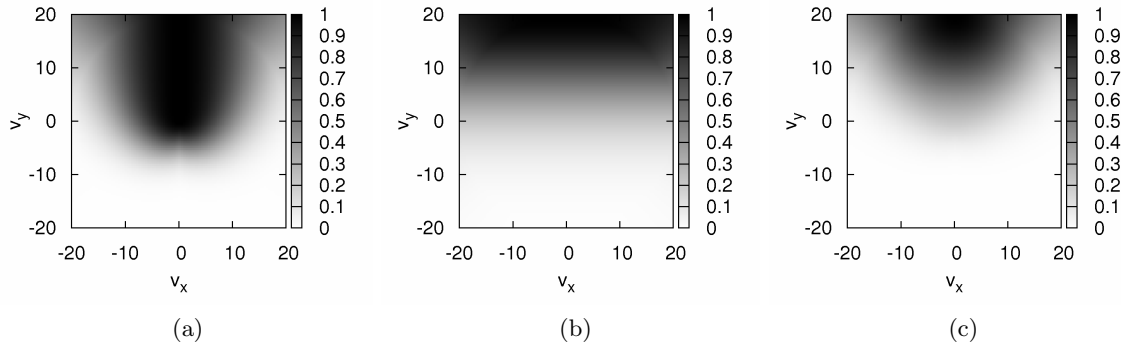
Nedostatkom doteraz popísaného algoritmu je, že nízke skóre tokenu môže byť jeho zotrvaním v poslednom stave gesta postupne zvýšené až nad potrebnú hranicu a gesto je tak rozpoznané bez ohľadu na to, že v predchádzajúcich stavoch bolo skóre tokenu nízke. Preto bola pri presune tokenu z nejakého stavu do iného pridaná kontrola skóre, ktoré token získal v aktuálnom stave. Pokiaľ je toto skóre nízke, nie je token do ďalšieho stavu presunutý. Skóre sa ale môže zotrvaním tokenu v aktuálnom stave postupne zvýšiť, a potom už môže byť token presunutý do iného stavu.

Rozhranie umožňuje pridávať rozpoznávanie nových gest bez potreby ich tréningu. Gestá sú poskladané zo sekvencie dvoch typov základných zložiek: priamočiary pohyb v zadanom smere a zotrvanie na mieste. Príklad poskladania gesta zo základných zložiek je znázornený na obrázku 3.11. Oba typy zložiek je možné rozpoznávať pomocou jedného stavu HMM. Pre tento stav je potrebné odhadnúť vysielaciu pravdepodobnosť ako funkciu dvoch premenných $p(v, \alpha)$. Rýchlosť v je zhora obmedzená hodnotou v_{max} tak, že všetky hodnoty $v > v_{max}$ sú považované za v_{max} . Vďaka tomuto obmedzeniu bude model fungovať pre ľubovoľne veľké rýchlosti. A tiež vďaka tomuto obmedzeniu môže byť vysielacia pravdepodobnosť reprezentovaná histogramom. Ukázalo sa, že na reprezentáciu funkcie $p(v, \alpha)$ stačí aj relatívne malé rozlíšenie histogramu 8x8, teda rýchlosť aj smer sú kvantované do ôsmich hladín.

Pri definícii vysielacích pravdepodobností sa vychádza z kombinácie niekoľkých Gaussových funkcií. Parametre vysielacích pravdepodobností boli stanovené len na základe apriórnych znalostí a experimentov a neboli použité žiadne tréningové dáta. Vysielacia pravdepodobnosť stavu HMM reprezentujúceho zotrvanie na mieste je funkcia:

$$p_{static}(v, \alpha) = \frac{1}{k} \cdot \exp \left[-\frac{1}{2} \left(\frac{v}{\sigma_v} \right)^2 \right] \quad (3.3)$$

Vysielacia pravdepodobnosť p_{static} teda klesá so zväčšujúcou sa rýchlosťou v a vôbec nezávisí na smere pohybu α . Konštanta σ_v má význam podobný smerodajnej odchýlke v normálnom rozložení a ovplyvňuje strmúť klesania. Konštanta k slúži na normalizáciu a zabezpečuje, že integrál funkcie $p_{static}(v, \alpha)$ cez všetky hodnoty vstupných parametrov je rovný 1.



Obr. 3.12: Príklad zložiek vysielacej pravdepodobnosti $p_\phi(v, \alpha)$ pre stav reprezentujúci priamočiary pohyb v smere $\phi = \pi/2$, pre parametre: $v_{max} = 20$, $\sigma_v^2 = 133$, $\sigma_\phi^2 = \pi/8$. (a) Zložka $f_{\phi 1}(v, \alpha)$ predstavujúca závislosť na smere pohybu. (b) Zložka $f_{\phi 2}(v, \alpha)$ reprezentujúca závislosť na rýchlosti pod v smere pod uhlom ϕ . (c) $f_{\phi 1}(v, \alpha) \cdot f_{\phi 2}(v, \alpha)$.

Vysielacia pravdepodobnosť $p_\phi(v, \alpha)$ stavu reprezentujúceho priamočiary pohyb pod uhlom ϕ je definovaná vzťahmi:

$$p_\phi(v, \alpha) = \frac{1}{k} \cdot f_{\phi 1}(v, \alpha) \cdot f_{\phi 2}(v, \alpha) \quad (3.4)$$

$$f_{\phi 1}(v, \alpha) = \exp \left[-\frac{1}{2} \left(\frac{\arccos(\cos(\alpha - \phi))}{\sigma_\phi} \cdot \frac{v}{v_{max}} \right)^2 \right] \quad (3.5)$$

$$f_{\phi 2}(v, \alpha) = \exp \left[-\frac{1}{2} \left(\frac{v \cdot \cos(\alpha - \phi) - v_{max}}{\sigma_v} \right)^2 \right] \quad (3.6)$$

Funkcia $f_{\phi 1}(v, \alpha)$ predstavuje závislosť na smere pohybu. Vzťah $\arccos(\cos(\alpha - \phi))$ vyjadruje absolútnu hodnotu rozdielu uhlov α a ϕ . V prípade, že rýchlosť v je nulová, nie je možné určiť uhol α , a preto je rozdiel uhlov α a ϕ považovaný za nulový. Platí, že pre malé rýchlosti v sú hodnoty funkcie $f_{\phi 1}(v, \alpha)$ podobné pre všetky uhly α a so zvyšujúcou sa rýchlosťou narastá význam rozdielu uhlov α a ϕ na hodnotu tejto funkcie. Iný spôsob interpretácie funkcie $f_{\phi 1}(v, \alpha)$ je, že so stúpajúcou rýchlosťou v , klesá jej hodnota tým strmšie, čím je väčší rozdiel uhlov α a ϕ . Takže najrýchlejšie klesá pre $\alpha = \phi + \pi$ a najpomalšie, prípadne vôbec neklesá, pre uhly α blízke hodnote ϕ .

Funkcia $f_{\phi 2}(v, \alpha)$ predstavuje závislosť na veľkosti rýchlosti v v smere danom uhlom ϕ . Vzťah $v \cdot \cos(\alpha - \phi)$ označuje veľkosť projekcie vektoru rýchlosti na priamku pod uhlom ϕ .

Príklady zložiek $f_{\phi 1}(v, \alpha)$ a $f_{\phi 2}(v, \alpha)$ vysielacej pravdepodobnosti $p_\phi(v, \alpha)$ sa nachádzajú na obrázku 3.12. Vysielacia pravdepodobnosť $p_\phi(v, \alpha)$ je maximálna v prípade $v = v_{max}$ a $\alpha = \phi$ a minimálna pre $v = v_{max}$ a $\alpha = \phi + \pi$. Konštanty σ_v a σ_ϕ majú význam podobný smerodajnej odchýlke v normálnom rozložení a ovplyvňujú strmosť klesania vysielacej pravdepodobnosti na základe rýchlosti a rozdielu uhlov. Konštanta k slúži na normalizáciu a zabezpečuje, že integrál funkcie $p_\phi(v, \alpha)$ cez všetky hodnoty vstupných premenných je rovný 1.

Po stanovení vysielacích pravdepodobností je ďalej potrebné určiť prechodové pravdepodobnosti medzi stavmi. Stav s sú v modeli usporiadané presne podľa poradia zložiek gesta. Stav s má prechod sám na seba a na ďalší stav $s+1$ v rade spracovávať ďalšiu základnú zložku. Pokiaľ stav predstavuje zotrvanie na mieste, potom existuje prechod medzi predošlým a nasledujúcim stavom. To zabezpečí, že zotrvanie ruky na mieste je nepovinné.

Pravdepodobnosti prechodov zo stavu do seba samého sú mierne vyššie ako pravdepodobnosti prechodu medzi rôznymi stavmi. Vďaka tomu dáva dlhšie vykonávanie gesta vyššie skóre ako jeho rýchle vykonanie.

3.7 Generovanie udalostí

Navrhnuté rozhranie komunikuje s ovládanou aplikáciou generovaním niekoľkých typov udalostí, pričom každá udalosť nesie informáciu o zmene v niektorej časti rozhrania. Rozhranie žiadnym spôsobom nerieši výslednú interpretáciu udalostí. Tá závisí len na konkrétnej ovládanej aplikácii. Sú generované tieto typy udalostí:

Začiatok sledovania ruky Táto udalosť informuje aplikáciu, že na určitej pozícii v obraze bola v niekoľkých po sebe nasledujúcich snímkoch detekovaná nová ruka, a že ju rozhranie začalo sledovať. Táto udalosť je generovaná pri prechode sledovača danej ruky do stavu aktívneho sledovania.

Zmena pozície ruky Ruka zmenila svoju pozíciu. Udalosť informuje aplikáciu o aktuálnej pozícii, a tiež o veľkosti zmeny pozície.

Koniec sledovania Ruka prestala byť sledovaná. To môže nastať z dôvodu, že vyšla mimo záber kamery alebo začala byť sledovaná iná ruka. Táto udalosť je generovaná pri prechode sledovača danej ruky do stavu ukončenia sledovania.

Zmena pózy Ruka zmenila svoju pózu. Udalosť nesie informáciu o novej póze.

Rozpoznanie gesta. Na základe sledovania pohybu ruky bolo rozpoznané gesto.

Kapitola 4

Implementácia rozhrania a demonštračnej aplikácie

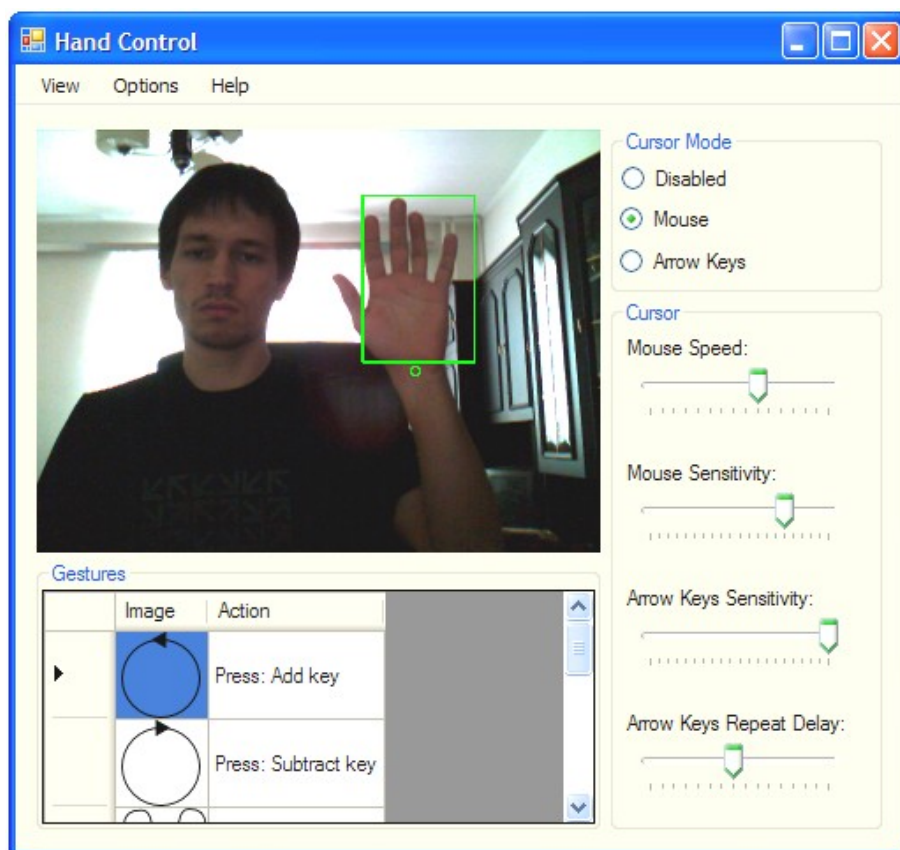
Navrhnuté rozhranie bolo implementované v jazyku C++ s použitím knižnice OpenCV. Bol kladený dôraz na konfigurovateľnosť a rozšíriteľnosť rozhrania, pritom ale so zachovaním jednoduchosti jeho použitia v aplikáciách. Jednotlivé časti rozhrania boli implementované s ohľadom na čo najväčšiu znovupoužiteľnosť. Medzi významné časti implementované ako samostatné moduly použiteľné mimo vytvoreného rozhrania patria napríklad: adaptovateľný model farby kože, sledovanie pohybu pomocou upraveného algoritmu Flocks of Features a rozpoznávanie dynamických gest. Implementované rozhranie by malo byť teoreticky prenosné na rôzne platformy avšak toto zatiaľ nebolo testované.

Implementovanému rozhraniu je možné nastaviť ľubovoľný počet detektorov rozpoznávajúcich rôzne pózy ruky a je možné určiť, ktorá póza bude používaná na inicializáciu sledovania. Taktiež sa dá nastaviť rozpoznávanie neobmedzeného počtu dynamických gest. Vstupom rozhrania je farebný obraz z kamery. Výstupom rozhrania sú udalosti, podľa ktorých potom cieľová aplikácia riadi svoju činnosť. Tá na spracovanie udalostí musí implementovať špeciálnu triedu a ukazateľ na túto triedu predať rozhraniu.

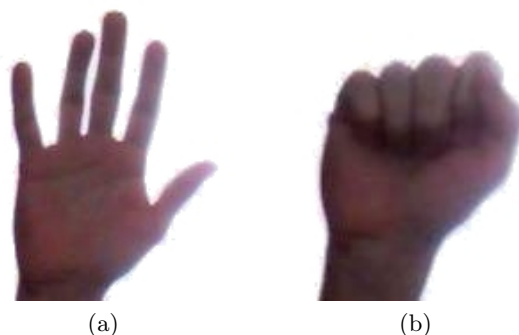
Implementované rozhranie bolo použité k vytvoreniu demonštračnej aplikácie. Táto na základe sledovania užívateľovej ruky umožňuje simulovať udalosti klávesnice a myši v operačnom systéme Windows a tým ovládať iné programy. Tento typ aplikácie bol zvolený z dôvodu, aby bolo možné pomocou rozhrania ovládať iné programy bez zásahu do ich kódu.

Po spustení aplikácie užívateľ vidí obraz z kamery. Tento obraz je prevrátený podľa zvislej osi, aby simuloval obyčajné zrkadlo. Je dôležité, aby kvôli adaptácii modelu farby kože bola v zábere užívateľova tvár. Práca s aplikáciou začne vložením otvorenej ruky do záberu kamery. Ruka začne byť sledovaná a aplikácia rozpoznáva pózy a detekuje dynamické gestá. Napríklad zatnutím ruky do päste je simulované stlačenie ľavého tlačidla myši. Viac o interpretácii pohybu ruky, rozpoznávaných pózach a gestách sa nachádza v nasledujúcich sekciách.

Demonštračná aplikácia bola podobne ako rozhranie implementovaná v jazyku C++. Na získanie obrazu z kamery bola použitá knižnica *videoInput*. Video vstup z knižnice OpenCV nebol použitý, pretože v aktuálnej verzii OpenCV 1.0 boli problémy s niektorými kamerami pod operačným systémom Windows Vista. Grafické užívateľské rozhranie demonštračnej aplikácie bolo implementované v .NET frameworku. Na obrázku 4.1 sa nachádza ukážka aplikácie v činnosti.



Obr. 4.1: Demonštračná aplikácia. V ľavej hornej časti okna je zrkadlovo prevrátený obraz z kamery. Sledovaná ruka je označená obdĺžnikom. Okolo obrazu z kamery sa nachádzajú rôzne nastavenia umožňujúce zmeniť interpretáciu pohybu ruky.



Obr. 4.2: Rozpoznávané pózy rúk: (a) otvorená ruka, (b) päšť. Otvorená ruka je inicializačná póza.

4.1 Interpretácia pohybu ruky a rozpoznania póz

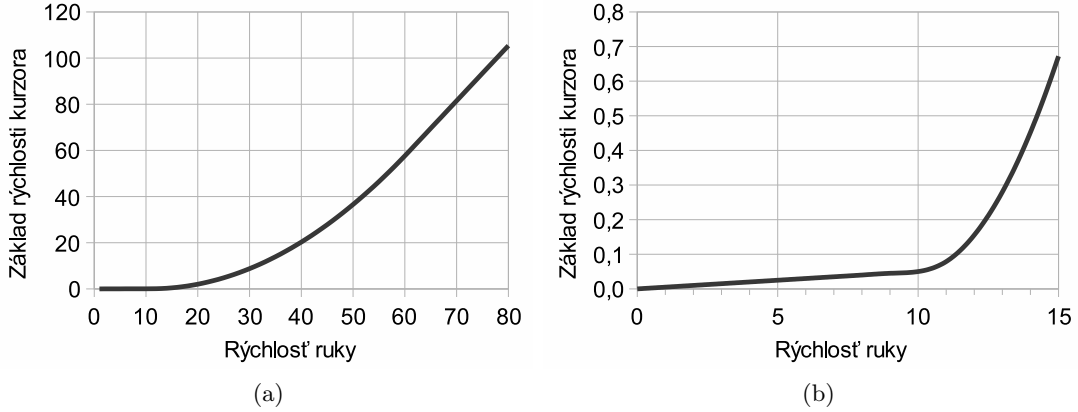
Demonštračná aplikácia rozpoznáva dve pózy: *voľne otvorená ruka* a *päšť*. Póza *otvorená ruka* je inicializačná, čo znamená, že touto pózou začína užívateľ svoju činnosť s aplikáciou. Rozpoznávané pózy sú znázornené na obrázku 4.2.

Vytvorená aplikácia umožňuje tri interpretácie pohybu ruky a póz. Prvou alternatívou je ignorovanie pohybu a póz, pričom ale stále funguje interpretácia dynamických gest. Takýto spôsob ovládania je vhodný napríklad pre multimediálne prehrávače, kde vystačí malé množstvo príkazov, ktoré je možné namapovať na jednotlivé gestá.

Druhý spôsob interpretácie pohybu ruky demonštračnou aplikáciou je simulácia stlačenia kláves šípok. Pri pohybe ruky nejakým smerom sa stlačí príslušná šípka. Raz za nastavený čas je stlačenie šípky opakované. Šípka sa uvoľní pri pohybe ruky späť do pôvodnej pozície. Päšť je interpretovaná ako stlačenie klávesy *enter*. Tento spôsob ovládania je vhodný napríklad v správcoch súborov alebo hrách.

Tretím spôsobom interpretácie pohybu je simulácia pohybu myši. Ruka zatatá v päšť znamená stlačenie ľavého tlačidla myši. Tento spôsob interpretácie má pravdepodobne najväčšie využitie a je vhodný napríklad na ovládanie elektronických máp alebo jednoduchých hier.

Prvým problémom pri interpretácii pohybu ruky ako pohyb myši je spôsob určenia pozície kurzora. Jednou možnosťou je určiť pozíciu kurzora priamo na základe pozície ruky v obraze z kamery. Druhou možnosťou je určiť pozíciu kurzora podľa relatívnej polohy ruky voči hlave užívateľa, čím sa zabezpečí nezávislosť polohy kurzora na polohe užívateľa v zábere. Obe varianty však majú vážny nedostatok, ktorým je malá presnosť. Majme napríklad bežnú kameru s rozlíšením 640x480 pixelov. Už pri relatívne nízkom rozlíšení monitora 1024x768 je maximálna teoretická presnosť kurzora 1,6 pixela. To však len za predpokladu, že sa ruka bude pohybovať v celom zábere kamery. Ďalej je nutné zahrnúť nepresnosť určenia pozície ruky a jej sledovania. Uvedené metódy určovania pozície kurzora sú teda nevhodné pri nízkom rozlíšení bežnej kamery. Preto bola zvolená metóda, kedy nie je určovaná poloha kurzora, ale zmena jeho polohy. Pohyb kurzora je odvodený od zmeny pozície ruky. Rýchlosť kurzora sa vypočíta podľa akceleračnej krivky danej vzťahom $v_{cursor} = r \cdot f(v_{hand})$, kde v_{cursor} je veľkosť výslednej rýchlosti kurzora, r je koeficient ovplyvňujúci výslednú rýchlosť a $f(v_{hand})$ je nelineárna akceleračná funkcia. Táto funkcia prechádza bodmi $[0, 0]$, $[m_x, m_y]$ a $[l_x, l_y]$. Akceleračná funkcia je pre rýchlosť pohybu ruky z intervalu $v_{hand} \in \langle 0, m_x \rangle$ lineárna. Na intervale $\langle m_x, l \rangle$ funkcia rastie kvadraticky a na intervale $\langle l, \infty \rangle$ je opäť lineárna. Derivá-



Obr. 4.3: Akceleračná krivka pre parametre $[m_x, m_y] = [10; 0,05]$ a $[l_x, l_y] = [60; 60]$. Na vodorovnej osi je rýchlosť ruky v_{hand} . Na zvislej osi je $f(v_{hand})$. Obidve rýchlosti sú udávané v pixeloch za sekundu. (a) Na začiatku rastie akceleračná funkcia lineárne. V strede rastie kvadraticky a na konci opäť lineárne. (b) Zväčšená začiatočná časť krivky, kde lineárne stúpanie prechádza na kvadratické.

cia funkcie je spojitá. Na základe týchto obmedzení bol pre akceleračnú funkciu odvodený vzťah:

$$f(v_{hand}) = \begin{cases} \frac{m_y \cdot v_{hand}}{m_x} & v_{hand} \in \langle 0, m_x \rangle \\ av_{hand}^2 + bv_{hand} + c & v_{hand} \in \langle m_x, l_x \rangle \\ (2al_x + b) \cdot (v_{hand} - l_x) + l_y & v_{hand} \in \langle l_x, \infty \rangle \end{cases} \quad (4.1)$$

$$a = \frac{l_y m_x - l_x m_y}{m_x^3 - 2l_x m_x^2 + l_x^2 m_x} \quad (4.2)$$

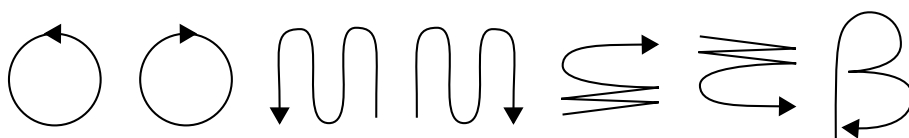
$$b = \frac{(m_y - 2am_x^2)}{m_x} \quad (4.3)$$

$$c = m_y - am_x^2 - bm_x \quad (4.4)$$

Odvozenie tohto vzťahu je možné nájsť v prílohe B. Takto definovaná funkcia umožňuje presne nastaviť pozíciu kurzora, keď sa ruka hýbe pomaly. Taktiež malé pohyby spôsobené nepresnosťou sledovania nemajú veľký vplyv na pozíciu kurzora. Pri rýchlom pohybe ruky je zase naopak možné premiestniť kurzor na veľkú vzdialenosť. Ukážka akceleračnej krivky je na obrázku 4.3.

Ukázalo sa ako vhodné aplikovať akceleračnú funkciu osobitne na horizontálnu a vertikálnu zložku rýchlosti. Vďaka tomu môže užívateľ pohyb kurzora lepšie obmedziť na horizontálny, prípadne vertikálny smer.

Ďalší faktor, ktorý by mohol ovplyvňovať rýchlosť kurzora je veľkosť ruky v obraze. Význam by mohol byť najmä v kompenzácii perspektívneho zmenšenia závislého na vzdialenosti od kamery. Iným využitím by bola možnosť aktívne meniť rýchlosť kurzora v závislosti na vzdialenosti od kamery. Napríklad, keď by užívateľ priblížil ruku ku kamere mohla by sa znížiť rýchlosť kurzora a tým by bolo možné presnejšie nastaviť jeho pozíciu. Vzdialením ruky by sa naopak rýchlosť zvýšila a bolo by možné kurzor presunúť na veľkú vzdialenosť. S nastavovaním rýchlosti kurzora podľa veľkosti ruky bolo experimentované, ale zatiaľ nie je zahrnuté vo vytvorenej aplikácii. Na praktické použitie tejto techniky bude pravdepodobne potrebné sprítniť určenie veľkosti ruky.



Obr. 4.4: Rozpoznávané dynamické gestá.

Ďalším užitočným spôsobom interpretácie pohybu, ktorý ale zatiaľ nie je implementovaný v demonštračnej aplikácii, by mohla byť simulácia rotácie koliečka myši. Táto interpretácia by mohla byť súčasťou simulácie pohybu myši tak, že keď by bola ruka v určitej póze, pohyb ruky by namiesto pohybu kurzora znamenal rotáciu koliečka. Bolo by tiež vhodné vyhradiť špeciálnu pózu, v ktorej keby sa nachádzala ruka, bol by jej pohyb dočasne ignorovaný. Toto by simulovalo nadzdvihnutie myši.

4.2 Interpretácia dynamických gest

Dynamické gestá sa dajú principiálne použiť na zadávanie príkazov, ktoré nezávisia na pozícii kurzora. Môžu to byť napríklad príkazy multimediálnemu prehrávaču alebo vyvolanie rôznych menu. Nevýhodou zadávania príkazov pomocou gest je ich dlhý čas vykonávania oproti zmene pózy ruky. Na druhú stranu je ale ich rozpoznanie výrazne menej výpočtovo náročné ako rozpoznanie pózy. Taktiež pomocou metódy uvedenej v sekcii 3.6 je možné pridávať rozpoznanie nových gest bez potreby tréningu, pričom na rozpoznanie novej pózy je vždy nutné relatívne zdĺhavé tréningovanie.

Demonštračná aplikácia rozpoznáva sedem gest znázornených na obrázku 4.4. Tieto gestá boli zvolené tak, aby ich pravdepodobnosť náhodného vykonania pri bežnom pohybe bola malá. Tiež bola snaha, aby gestá neboli rozpoznávačom ľahko zameniteľné. Nutnou podmienkou je preto, že žiadne gesto nie je obsiahnuté v inom.

Na základe detekcie gesta sú vykonávané dva typy akcií:

Simulácia stlačenia nastavenej klávesovej kombinácie Klávesy sú ihneď po stlačení automaticky uvoľnené. Využitie tohto spôsobu interpretácie gest je pre ovládanie programov pomocou klávesových skratiek. Napríklad namapovaním kláves *plus* a *mínus* na niektoré gestá je možné približovať a vzdalovať elektronické mapy, prípadne iné dokumenty.

Zmena módu interpretácie pohybu ruky Toto je užitočné, keď chceme dočasne vypnúť alebo zapnúť interpretáciu pohybu ruky.

4.3 Generovanie udalostí v operačnom systéme Windows

Na syntézu vstupných udalostí v operačnom systéme Windows slúži funkcia *SendInput()* zo štandardnej systémovej knižnice *user32*. Táto funkcia umožňuje simulovať vstupy z myši, klávesnice, prípadne z iného hardvéru. Takýmito vstupmi sú napríklad: stlačenie a uvoľnenie klávesy, stlačenie a uvoľnenie tlačidla myši, pohyb myši a podobne.

Na kliknutie tlačidla je teda nutné generovať dve udalosti: stlačenie a uvoľnenie. Pre simuláciu klávesovej kombinácie je potrebné postupne generovať stlačenie jednotlivých kláves a následne ich postupné uvoľňovanie. Napríklad pre kombináciu *Alt+F4* sú to štyri udalosti: stlačenie klávesy *Alt*, stlačenie *F4*, uvoľnenie *F4* a nakoniec uvoľnenie *Alt*.

Vďaka funkcii *SendInput()* je demonštračná aplikácia schopná na základe rozpoznania gesta generovať aj klávesy, ktoré sa nenachádzajú na bežných klávesniciach. Jedná sa napríklad o rôzne multimediálne klávesy ako: *play*, *next track*, *volume up*, *volume down*, *browser back*, *browser forward* a podobne.

Ďalej je potrebné upozorniť, že v prípade simulácie dlhšieho spojitého stlačenia klávesy, nie je automaticky generované jej opakovanie. Opakovanie klávesy je možné simulovať periodickým generovaním udalostí uvoľnenia a opätovného stlačenia danej klávesy.

I keď funkcia *SendInput()* umožňuje syntetizovať aj pohyb myši, je pozícia kurzora nastavovaná kvôli jednoduchšiemu použitiu dvojicou funkcií *GetCursorPos()* a *SetCursorPos()*. Tieto funkcie sú súčasťou systémovej knižnice *user32*. Funkcia *GetCursorPos()* slúži na získanie aktuálnej polohy kurzora na obrazovke a funkcia *SetCursorPos()* na jej nastavenie. Tieto funkcie však pracujú iba s celočíselnou polohou kurzora v pixeloch. To je ale nedostačujúce hlavne pre pomalé pohyby, kedy je zmena polohy kurzora na základe dvoch po sebe idúcich snímok menšia ako jeden pixel. V takom prípade by sa kurzor namiesto pomalého pohybu nehýbal vôbec. Preto vytvorená aplikácia interne akumuluje zmenu polohy kurzora vo formáte s desatinnou čiarkou. Keď akumulovaná zmena dosiahne po zaokrúhlení veľkosť aspoň jedného pixela, je nastavená nová pozícia kurzora na obrazovke a aplikovaná celočíselná zmena polohy sa odčíta od akumulátora.

Kapitola 5

Testy

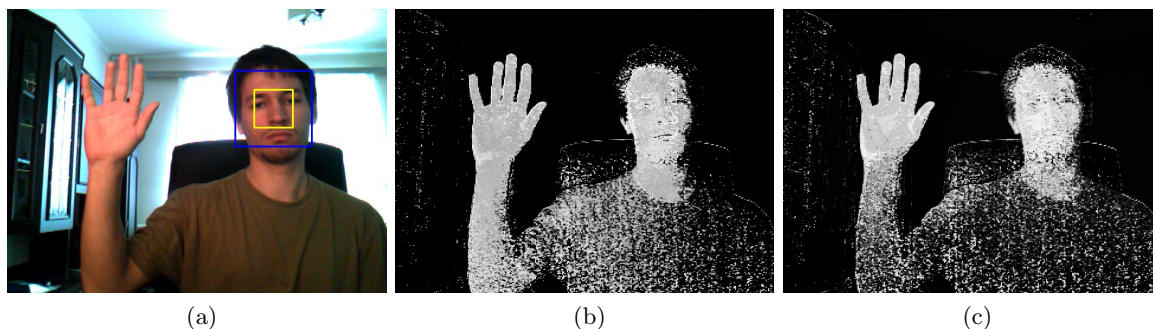
V tejto kapitole sa nachádzajú testy jednotlivých častí rozhrania a rozhrania ako celku. Taktiež je vyhodnotená efektivita ovládania aplikácií pomocou vytvoreného rozhrania.

5.1 Modely farby kože

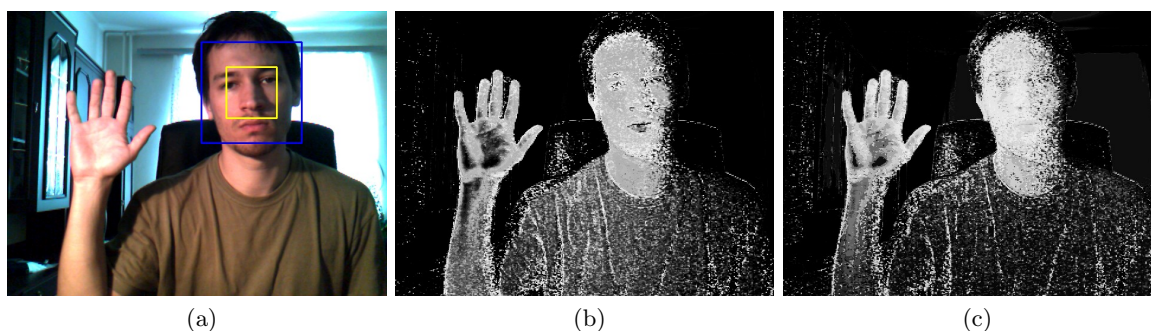
V tejto sekcii sú skúmané rôzne možnosti modelovania farby kože. Vo všetkých testoch prebieha adaptácia len na základe detekcie tváre tak, ako je to popísané v sekcii 3.1. Výstupom testov sú obrázky znázorňujúce pravdepodobnosť kože. Čím je pixel svetlejší, tým je väčšia pravdepodobnosť, že sa jedná o kožu.

V prvom teste boli porovnávané dve možnosti reprezentácie rozloženia hustoty pravdepodobnosti pre triedu farby kože: normálne rozloženie a normalizovaný histogram. Pozadie je v oboch prípadoch reprezentované normalizovaným histogramom. Výsledky rozpoznania kože sú znázornené na obrázkoch 5.1 a 5.2.

Pri dobrom osvetlení dávajú obe reprezentácie podobné výsledky, i keď histogram je schopný lepšie odlíšiť podobné farby. V prípade presvetlenej kože je dosiahnuté lepšie rozpoznanie pri použití histogramu. Dôvodom takýchto výsledkov je, že histogram menej generalizuje, a teda je schopný presnejšie popísať rozloženie hustoty pravdepodobnosti tréningových dát.



Obr. 5.1: Porovnanie reprezentácií rozloženia hustoty pravdepodobnosti pre triedu farby kože pri dobrom osvetlení. (a) Originálny obraz. (b) Použitie normálne rozloženie. (c) Použitý normalizovaný histogram.



Obr. 5.2: Porovnanie reprezentácií rozloženia hustoty pravdepodobnosti pre triedu farby kože v obraze s presvetlenými oblasťami kože. (a) Originálny obraz. (b) Použité normálne rozloženie. (c) Použitý normalizovaný histogram.

Druhý test modelov farby kože porovnával rôzne spôsoby adaptácie. V tomto prípade boli obe triedy reprezentované normalizovaným histogramom. Výsledky testu sú na obrázku 5.3.

Výsledky testu ukázali, že adaptácia modelu výrazne prispieva k dobrej kvalite rozpoznania kože. Rozdiel medzi MAP adaptáciou a kompletným pretrénovaním modelu je však malý. Toto je pravdepodobne spôsobené dostatočným množstvom kvalitných adaptačných dát získaných na základe detekcie tváre.

5.2 Úspešnosť detektorov rúk

V súčasnosti sú v demonštračnej aplikácii použité detektory dvoch póz: päšť a voľne otvorená ruka. Obidva detektory boli natrénované na základe dát automaticky získaných metódou popísanou v sekcii 3.4. Bola použitá implementácia Viola&Jones detektora z knižnice OpenCV. Rozlíšenie trénovacích príkladov bolo 24x24 pixelov. Trénovacie dáta boli získané za niekoľko málo hodín, pričom bola snaha simulovať rôzne svetelné podmienky.

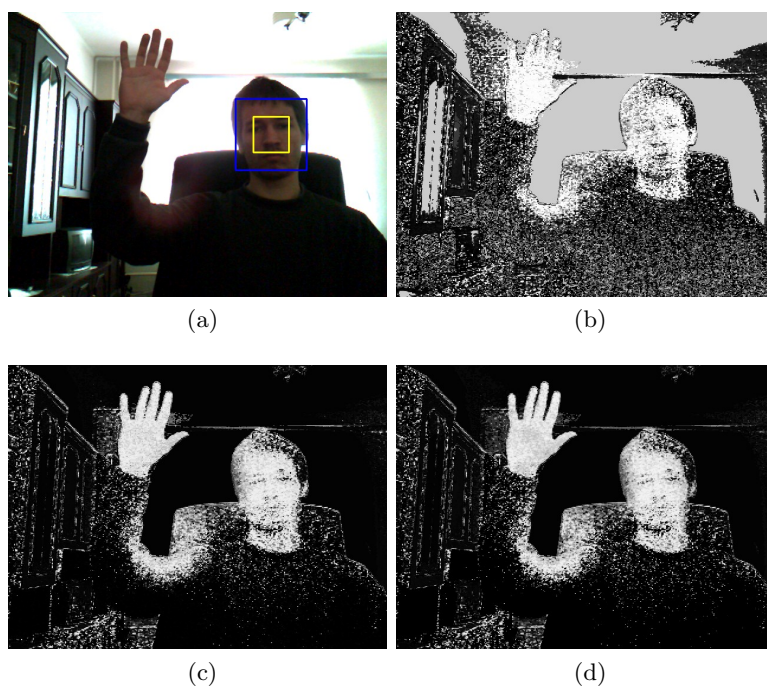
Detektory boli testované na snímkoch s rozlíšením 640x480 pixelov, ktoré obsahovali zložité pozadie a vysoký šum kamery. V každom snímku bol vždy práve jeden výskyt hľadanej pózy ruky. Výsledky testov sú znázornené ROC krivkami, kde false alarm rate je daný ako pomer počtu falošných detekcií k počtu snímkov.

Prvý testovaný detektor rozpoznával ruku zaťatú v päšť. Bol natrénovaný na 5625 príkladoch. Úspešnosť detektora bola testovaná na 1249 snímkoch. Výsledky testov sú znázornené na obrázku 5.4.

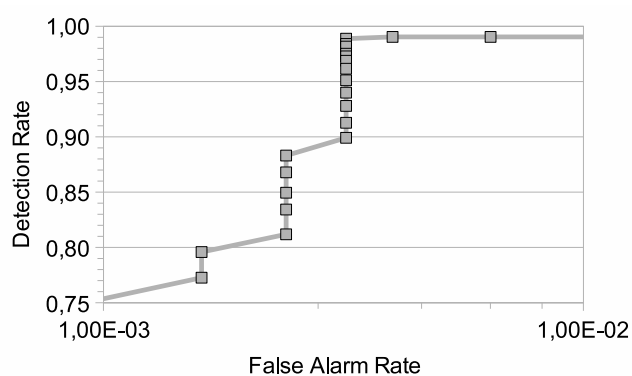
Druhý testovaný detektor bol natrénovaný na 6464 príkladoch voľne otvorenej ruky. Jeho úspešnosť bola testovaná na 1149 snímkoch. Výsledky sú znázornené na obrázku 5.5.

Testy ukázali, že detektor päste je veľmi kvalitný. Pri nájdení 98,9% rúk generuje falošnú detekciu iba raz za 250 snímkov.

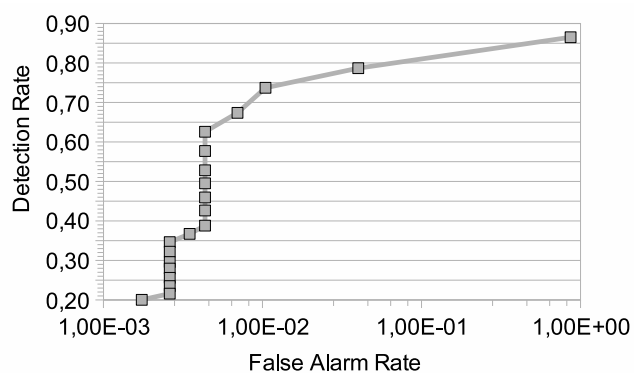
Detektor otvorenej ruky dosiahol výrazne horšie výsledky oproti detektoru päste. To môže byť spôsobené niekoľkými príčinami. Otvorená ruka má oveľa vyššiu variabilitu tvarov. Otvorená ruka nemá výraznú vnútornú štruktúru a je rozpoznateľná najmä podľa obvodu. Obrázky otvorenej ruky obsahujú výrazne viac pozadia ako obrázky päste. V prípade päste je pozadie iba na okrajoch, ale v prípade otvorenej ruky pozadie presvitá aj pomedzi prsty. Avšak aj napriek horším výsledkom testov je detektor otvorenej ruky pre potreby rozhra-



Obr. 5.3: Porovnanie možností adaptácie. (a) Originálny obraz. (b) Bez adaptácie. (c) MAP adaptácia. (d) Kompletne pretrénovanie.



Obr. 5.4: Výsledky detektora päste.



Obr. 5.5: Výsledky detektora voľne otvorenej ruky.

	Minimum	Maximum	Priemer
Neaktívne rozhranie	0	281	18,07
Aktívne rozhranie	46	188	79,82

Tabuľka 5.1: Čas potrebný na spracovanie jedného snímku implementovaným rozhraním. Hodnoty sú uvedené v milisekundách.

nia použiteľný. Sledovanie a spôsob rozpoznávania pózy, čiastočne kompenzujú jeho nízku úspešnosť.

Výsledky detektorov by bolo možné zlepšiť niekoľkými spôsobmi. Prvou možnosťou je zvýšiť rozlíšenie tréovacích príkladov [8]. To však spôsobí výrazný nárast času potrebného na tréovanie. Druhou možnosťou je získanie väčšieho množstva tréovacích príkladov, ktoré by zachytávali väčšiu variabilitu osvetlenia, pozadia, či použitých kamier. Taktiež by bolo dobré získať obrázky rúk od väčšieho počtu osôb.

5.3 Výpočtová náročnosť rozhrania

Pri vyhodnocovaní náročnosti rozhrania na výkon je nutné brať do úvahy dve situácie:

- Neaktívne rozhranie – užívateľ s rozhraním nepracuje a iba sa čaká na vloženie ruky do záberu kamery.
- Aktívne rozhranie – sleduje sa užívateľova ruka a ten môže pracovať.

Obidve situácie boli testované pomocou demonštračnej aplikácie na sekvencii 600 snímkov. Rozlíšenie použitej kamery bolo 640x480 pixelov. Testovací notebook obsahoval procesor Intel Core Duo T2080 s frekvenciou 1.73 GHz a 2 GB RAM s frekvenciou 666 MHz. Vyhľadávanie rúk a tváre sa striedali a vždy raz za 8 snímkov bolo aplikované práve jedno z nich. Výsledky testov sa nachádzajú v tabuľke 5.1.

Podľa očakávaní neaktívne rozhranie potrebuje výrazne kratší čas na spracovanie jedného snímku ako aktívne. Nulový minimálny čas platí pre všetky snímky, kde nebola vyhľadávaná tvár ani ruka.

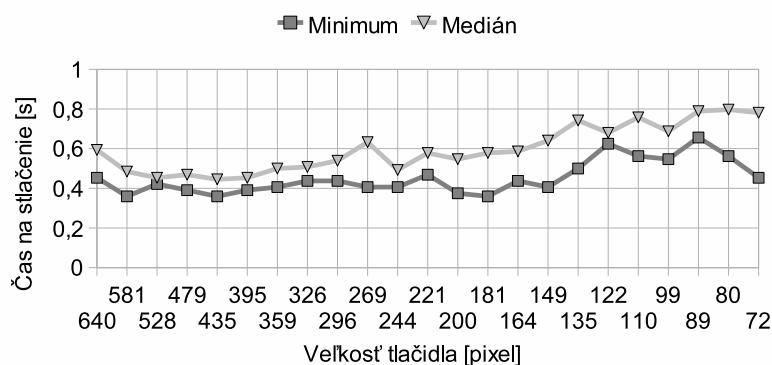
Priemerný čas na jeden snímok v prípade aktívneho rozhrania umožňuje spracovať približne 12,5 snímkov za sekundu. To je, ako ukázali testy v sekcii 5.4, dostatočné na interaktívne ovládanie aplikácií pomocou tohto rozhrania.

Z profilovania demonštračnej aplikácie sa ukázalo, že najviac času spotrebuje detekcia objektov. Urýchlenie by teda bolo možné dosiahnuť napríklad natrénovaním efektívnejších detektorov. Taktiež by sa dal zmenšiť počet pozícií, ktoré musia detektory prehľadávať.

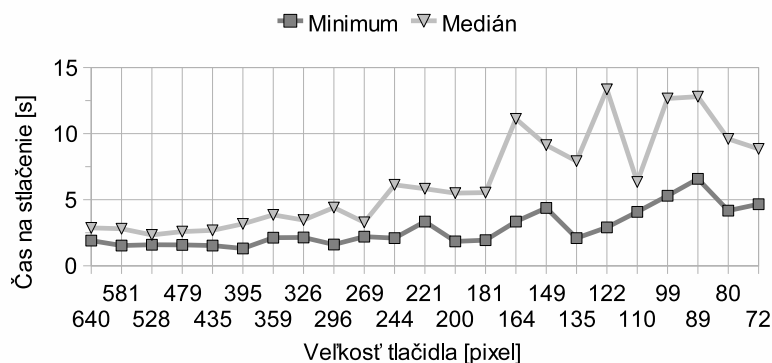
5.4 Efektivita práce s rozhraním

Vďaka dobrovoľníkom bola testovaná efektivita ovládania programov pomocou vytvoreného rozhrania. Efektivita ovládania bola porovnávaná s ovládaním myšou. Na testovanie bola použitá vytvorená demonštračná aplikácia umožňujúca ovládať programy v operačnom systéme Windows. Účastníci testu pracovali s rozhraním po prvý raz.

Testovanie prebiehalo na notebooku s procesorom Intel Core Duo s frekvenciou 1,73 GHz a s kamerou zabudovanou v hornej časti displeja, ktorá mala rozlíšenie 640x480 pixelov. Rozlíšenie obrazovky bolo 1280x800 pixelov.



(a)



(b)

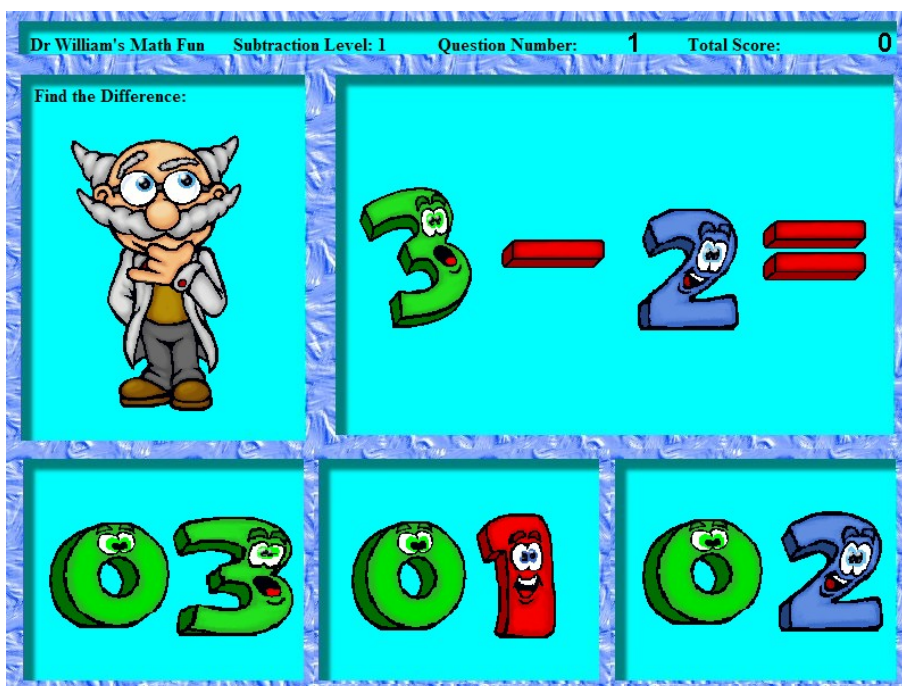
Obr. 5.6: Výsledky testu klikania na rôzne veľké tlačidlá. (a) Výsledky testov ovládania s myšou. (b) Výsledky testov ovládania pomocou vytvoreného rozhrania.

Boli zvolené tri testovacie úlohy: klikanie na rôzne veľké tlačidlá, jednoduchá aplikácia s veľkými tlačidlami a ovládanie elektronickej mapy. Dobrovoľníci vykonávali tieto úlohy pomocou myši a pomocou vytvoreného rozhrania. Testy vykonalo celkovo 12 dobrovoľníkov.

Pre otestovanie rýchlosti klikania na rôzne veľké tlačidlá bol vytvorený jednoduchý program. V tomto bolo na začiatku na náhodnej pozícii zobrazené tlačidlo s rozmermi 640x640 pixelov. Po kliknutí naň sa jeho veľkosť zmenšila a tlačidlo sa náhodne presunulo na inú pozíciu. Test končil pri veľkosti tlačidla 72x72 pixelov. Pre každú veľkosť tlačidla sa zaznamenával čas, ktorý užívateľ potreboval, aby naň klikol.

Výsledky testov sa nachádzajú v grafoch na obrázku 5.6. Pre každú veľkosť tlačidla bol sledovaný najlepší dosiahnutý čas a medián dosiahnutých časov. Medián bol namiesto priemeru zvolený z dôvodu odolnosti voči extrémnym hodnotám. Testy ukázali že, rýchlosť klikania na tlačidlá je výrazne vyššia v prípade použitia myši oproti použitiu vytvoreného rozhrania. Už samotné kliknutie pomocou vytvoreného rozhrania je pomalšie. To je spôsobené tým, že musí najskôr dôjsť k zmene pózy na päst a následne k zmene na inú pózu pre uvoľnenie tlačidla. A ako je uvedené v sekcii 3.3, pre zmenu pózy je nutné, aby bola nová póza detekovaná určitý čas.

Na druhú testovaciu úlohu bol použitý program znázornený na obrázku 5.7. Účastníci testu mali vypočítať 10 veľmi jednoduchých príkladov a výsledok zvoliť kliknutím na jednu z troch ponúkaných možností. Bol zaznamenávaný celkový čas, ktorý na to jednotliví účastníci potrebovali.



Obr. 5.7: Testovací program, v ktorom bolo nutné vypočítať 10 jednoduchých príkladov. Odpoveď sa zvolila kliknutím na príslušný výsledok.

Spôsob ovládania	Minimum	Maximum	Priemer	Medián
myš	32	42	36	35
vytvorené rozhranie	50	129	76	67

Tabuľka 5.2: Výsledky testu, kde bolo nutné vypočítať 10 jednoduchých príkladov a zvoliť správnu odpoveď kliknutím na jednu z troch možností. Časy sú uvedené v sekundách.

Výsledky pre túto úlohu sa nachádzajú v tabuľke 5.2. Opäť podobne ako v predchádzajúcej úlohe je ovládanie myšou efektívnejšie, ale rozdiel nie je taký výrazný. Príčinou menšieho rozdielu je, že ovládanie zaberá iba časť z celkového potrebného času. Zvyšný čas je spotrebovaný na samotné vypočítanie príkladu, a tiež na animácie programu po zadaní výsledku.

Poslednou úlohou bolo ovládanie elektronickej mapy na adrese <http://maps.google.com>. Približovanie a oddaľovanie bolo v prípade ovládania vytvoreným rozhraním riešené namapovaním kláves *plus* a *mínus* na gestá *krúžok vľavo* a *krúžok vpravo*. Bol sledovaný čas potrebný na nájdenie miesta na mape v Českej Republike alebo na Slovensku, ktoré účastník testu dobre poznal. Test začínal vždy z implicitnej pozície mapy nad USA a končil, keď bolo dané miesto úplne priblížené. Priblíženie bolo z počiatočnej pozície nutné vykonať minimálne 15-krát.

Výsledky testov ovládania elektronickej mapy sú v tabuľke 5.3. Znovu je ovládanie myšou značne rýchlejšie ako použitie vytvoreného rozhrania. V tomto prípade je to spôsobené hlavne pomalým približovaním mapy pomocou gest. Umožnením simulovania dvojkliku alebo koliečka myši by sa pravdepodobne efektívnosť ovládania máp vytvoreným rozhraním výrazne zvýšila.

Na základe vykonaných testov sa ukázalo, že vytvorené rozhranie je použiteľné na ovládanie aplikácií. Jeho efektívnosť však vyšla nižšia ako ovládanie myšou. Dosiahnutá nižšia efektívnosť je však čiastočne spôsobená neskúsenosťou účastníkov testu. Títo sa s rozhraním

Spôsob ovládania	Minimum	Maximum	Priemer	Medián
myš	18	58	30	26
vytvorené rozhranie	97	310	204	203

Tabuľka 5.3: Výsledky testu ovládania elektronickej mapy. Časy sú uvedené v sekundách a vyjadrujú celkový čas potrebný na nájdenie zvoleného miesta.

pri testovaní stretli prvý raz a podľa ich vlastných slov už počas vykonávania jednotlivých testovacích úloh pocítovali zlepšenie v používaní rozhrania. Ďalším dôvodom nižšej efektivity je fakt, že existujúce programy sú primárne určené na ovládanie myšou. Natívna podpora programov pre ovládanie rukou by mohla efektivitu použitia vytvoreného rozhrania výrazne zvýšiť. Určite je tiež priestor aj na zlepšenie a spresnenie samotného rozhrania.

Kapitola 6

Záver

V rámci tejto diplomovej práce bolo navrhnuté a implementované rozhranie umožňujúce ovládať aplikácie na základe sledovania pohybu užívateľovej ruky v obraze z kamery. Samotnému návrhu predchádzalo štúdium existujúcich metód detekcie a sledovania rúk. Navrhnuté rozhranie bolo implementované v jazyku C++ a bol kladený veľký dôraz na jednoduchosť jeho použitia v aplikáciách a možnosť jeho rozširovania v budúcnosti. Tiež bola snaha o čo najväčšiu znovupoužiteľnosť jeho jednotlivých častí.

Podľa vykonaných testov je rozhranie schopné spracovať na bežnom notebooku viac ako 12 snímkov s rozlíšením 640x480 pixelov za sekundu. To postačuje na interaktívne ovládanie aplikácií pomocou ruky. Výhodou vytvoreného rozhrania je, že nevyžaduje žiadny špeciálny hardvér a je možné s ním pracovať aj za použitia bežnej a málo kvalitnej kamery.

Implementované rozhranie bolo použité na vytvorenie demonštračnej aplikácie. Táto aplikácia je schopná na základe detekovaného pohybu, pózy a dynamických gest ruky generovať udalosti klávesnice a myši v operačnom systéme Windows. Takto je možné simulovať napríklad pohyb a klikanie myši alebo stláčanie klávesových kombinácií a tým ovládať iné programy bez nutnosti zásahu do ich kódu.

Vďaka niekoľkým dobrovoľníkom boli vykonané testy vytvoreného rozhrania, ktoré potvrdili, že toto rozhranie je použiteľné na ovládanie počítača. Medzi vykonanými testami bolo napríklad ovládanie online máp na adrese <http://maps.google.com>. Ovládanie programov pomocou tohto rozhrania však bolo podľa testov menej efektívne ako ovládanie myšou. To je ale čiastočne spôsobené tým, že existujúce programy nie sú na takýto spôsob ovládania navrhnuté. Natívna podpora programov pre ovládanie rukou by mohla efektívnosť použitia rozhrania zvýšiť. Ďalším dôvodom nižšej efektivity je, že dobrovoľníci testujúci vytvorené rozhranie sa s ním stretli prvý raz, pritom ale myš používajú často.

Vytvorené rozhranie by mohlo byť použiteľné v rôznych informačných termináloch, prípadne v situáciách, keď je potrebné zabezpečiť bezdotykové alebo diaľkové ovládanie. Ešte jedno využitie by mohlo byť napríklad v hrách.

Okrem vytvoreného rozhrania a demonštračnej aplikácie je ďalším prínosom tejto práce návrh inovatívnej metódy automatického získavania tréningových dát pre detektory rúk. Vďaka tejto metóde bolo možné nazbierať tisíce tréningových príkladov v priebehu niekoľkých hodín a natrénovať tak detektory rúk dostatočne kvalitné pre použitie vo vytvorenom rozhraní.

6.1 Námety na budúce pokračovanie práce

Pre pohodlnejšie použitie rozhrania bude potrebné zlepšiť detekciu rúk a spresniť ich sledovanie. V budúcnosti by mala byť tiež snaha o zníženie nárokov rozhrania na výpočtový výkon. To by umožnilo spracovať väčšie množstvo snímkov za sekundu, prípadne by bolo možné pracovať s vyšším rozlíšením kamery. Tým by sa pravdepodobne zvýšila presnosť ovládania. Ďalším dôvodom znižovania výpočtových nárokov je spotreba elektrickej energie. Tá je dôležitá najmä v prípade nasadenia rozhrania v prenosných zariadeniach.

Vhodným rozšírením rozhrania a vytvorenej aplikácie by bolo sledovanie obidvoch rúk užívateľa, a tiež rozpoznávanie väčšieho množstva póz a dynamických gest. Rozpoznanie väčšieho množstva póz umožní nové možnosti interpretácie pohybu ruky, akým je napríklad simulácia rotácie koliečka myši. Ďalšia špeciálna póza by mohla byť interpretovaná ako dvojklik.

Literatúra

- [1] Burget, L.: SRE – Statistické rozpoznávání vzorů II. 2006.
- [2] Černocký, J.: SRE – Skryté Markovovy modely HMM. 2008.
- [3] Forsyth, D. A.; Ponce, J.: *Computer Vision: A Modern Approach*. Prentice-Hall, první vydání, 2003, 693 s.
- [4] Francke, H.; Ruiz del Solar, J.; Verschae, R.: Real-Time Hand Gesture Detection and Recognition Using Boosted Classifiers and Active Learning. In *Pacific Rim Symposium on Image Video and Technology*, 2007, s. 533–547.
- [5] Freeman, W. T.; Anderson, D. B.; Beardsley, P. A.; aj.: Computer Vision for Interactive Computer Graphics. *IEEE Computer Graphics and Applications*, ročník 18, č. 3, 1998: s. 42–53, ISSN 0272-1716.
- [6] Isard, M.; Blake, A.: CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, ročník 29, 1998: s. 5–28.
- [7] Jo, K.-H.; Kuno, Y.; Shirai, Y.: Context-Based Recognition of Manipulative Hand Gestures for Human Computer Interaction. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision*, ročník II, London, UK: Springer-Verlag, 1997, ISBN 3-540-63931-4, s. 368–375.
- [8] Kölsch, M.: *Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments*. Dizertační práce, University of California, 2004.
- [9] Kohler, M.: Vision Based Hand Gesture Recognition Systems.
URL <http://ls7-www.cs.uni-dortmund.de:81/research/gesture/>
- [10] Malik, S.: Real-time Hand Tracking and Finger Tracking for Interaction. Technická zpráva, University of Toronto, 2003.
- [11] Manresa, C.; Varona, J.; Mas, R.; aj.: Real-Time Hand Tracking and Gesture Recognition for Human-Computer Interaction. 2000.
- [12] Pham, M.-T.; Cham, T.-J.: Fast training and selection of Haar features using statistics in boosting-based face detection. In *IEEE International Conference on Computer Vision*, 2007, s. 1–7.
- [13] Sudderth, E. B.; Mandel, M. I.; Freeman, W. T.; aj.: Visual Hand Tracking Using Nonparametric Belief Propagation. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, ročník 12, Washington, DC, USA: IEEE Computer Society, 2004, ISBN 0-7695-2158-4, str. 189.

- [14] Viola, P. A.; Jones, M. J.: Rapid Object Detection using a Boosted Cascade of Simple Features. In *Computer Vision and Pattern Recognition*, 2001, s. 511–518.
- [15] van Vlaardingen, M.: Hand Models and Systems for Hand Detection, Shape Recognition and Pose Estimation in Video. 2006.
- [16] Wikipedia: Hidden Markov model. April 2009.
URL http://en.wikipedia.org/wiki/Hidden_Markov_model
- [17] Wilson, A.: Hand gesture recognition system targets medical applications. 2008.
URL http://www.vision-systems.com/display_article/340083/19/none/none/Techt/Hand-gesture-recognition-system-targets-medical-applications
- [18] Zhu, Y.: *Hand Detection and Tracking in an Active Vision System*. Diplomová práce, York University, 2003.

Dodatok A

Zoznam netlačených príloh

1. CD nosič

Dodatok B

Odvodenie akceleračnej funkcie

Akceleračná funkcia $f(x)$ prechádza bodmi $[0, 0]$, $[m_x, m_y]$ a $[l_x, l_y]$. Akceleračná funkcia je pre rýchlosť pohybu ruky z intervalu $x \in \langle 0, m_x \rangle$ lineárna. Na intervale $\langle m_x, l \rangle$ táto funkcia rastie kvadraticky a na intervale $\langle l, \infty \rangle$ je opäť lineárna. Derivácia akceleračnej funkcie je spojitá.

Na intervale $x \in \langle 0, m_x \rangle$ sa tak jedná o lineárnu interpoláciu bodov $[0, 0]$ a $[m_x, m_y]$. Takže pre tento interval platí $f_1(x) = \frac{m_y \cdot x}{m_x}$.

Na intervale $x \in \langle m_x, l_x \rangle$ akceleračná funkcia rastie kvadraticky, čo je možné popísať všeobecne vzťahom $f_2(x) = ax^2 + bx + c$. Funkcia $f_2(x)$ prechádza bodmi $[m_x, m_y]$ a $[l_x, l_y]$, takže musí platiť:

$$f_2(m_x) = m_y = am_x^2 + bm_x + c \quad (\text{B.1})$$

$$f_2(l_x) = l_y = al_x^2 + bl_x + c \quad (\text{B.2})$$

Keďže je derivácia akceleračnej funkcie spojitá, musí platiť $f_2'(m_x) = f_1'(m_x)$, takže dostávame ďalšiu rovnicu:

$$\frac{m_y}{m_x} = 2am_x + b \quad (\text{B.3})$$

Riešením sústavy troch lineárnych rovníc B.1, B.2 a B.3 dostaneme hodnoty koeficientov a , b , c :

$$a = \frac{l_y m_x - l_x m_y}{m_x^3 - 2l_x m_x^2 + l_x^2 m_x} \quad (\text{B.4})$$

$$b = \frac{(m_y - 2am_x^2)}{m_x} \quad (\text{B.5})$$

$$c = m_y - am_x^2 - bm_x \quad (\text{B.6})$$

Na poslednom intervale $x \in \langle l_x, \infty \rangle$ je akceleračná funkcia znovu lineárna: $f_3(x) = kx + q$. Vzhľadom na spojitosť derivácie akceleračnej funkcie musí platiť $f_3'(l_x) = f_2'(l_x)$, teda $k = 2al_x + b$. Funkcia $f_3(x)$ prechádza bodom $[l_x, l_y]$, takže platí $f_3(l_x) = l_y = kl_x + q$. Z toho sa odvodí $q = l_y - kl_x$. Dosadením získaných hodnôt k a q dostaneme $f_3(x) = (2al_x + b)(x - l_x) + l_y$.

Spojením všech intervalů dostaneme výsledný vztah pro akcelerační funkci:

$$f(x) = \begin{cases} f_1(x) = \frac{m_y \cdot x}{m_x} & x \in \langle 0, m_x \rangle \\ f_2(x) = ax^2 + bx + c & x \in \langle m_x, l_x \rangle \\ f_3(x) = (2al_x + b)(x - l_x) + l_y & x \in \langle l_x, \infty \rangle \end{cases} \quad (\text{B.7})$$