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

HRA S OVLÁDÁNÍM POMOCÍ GEST RUKY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE JOZEF BAŠTEK
AUTHOR

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

HRA S OVLÁDÁNÍM POMOCÍ GEST RUKY

HAND GESTURE CONTROLLED GAME

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JOZEF BAŠTEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL HRADIŠ
SUPERVISOR

BRNO 2009

Abstrakt

Tato práce se zabývá využitím počítačového vidění, jako prostředku pro interakci člověka s PC. Její součástí je návrh a implementace rozhraní pro ovládání hry pomocí gest a pohybů ruky, které snímá kamera. V práci jsou popsány a vyhodnoceny algoritmy segmentace obrazu (odečítání pozadí, prahování, Bayessův teorém, Gaussův model) i klasifikace gest ruky (knowledge-based klasifikace, Hu invariantní momenty a K-nearest neighours). Součástí implementace je mechanická konstrukce držáku snímací kamery a demonstrační 3D FPS hra.

Abstract

This paper is intended for computer vision usage as a way of human-computer interacion. It covers design and implementation of interface for game controlling by hand gestures and movements, that are captured by camera. It describes and evaluates image segmentation algorithms (background subtraction, thresholding, Bayes' theorem, gaussian model) as well as hand gesture classification algorithms (knowledge-based classification, Hu invariant moments and K-nearest neighours). Implementation includes machanical contruction of camera holeder and 3D FPS game as a interface usage demonstration.

Klíčová slova

počítačové videní, gesto ruky, rozpoznávání, morfologie, odečítání pozadí, prahováni, gauss, bayess, klasifikace, pozorování, kontura, Hu momenty, k-nearest neighours

Keywords

computer vision, hand gesture, recognition, morphology, background subtraction, thresholding, gauss, bayess, classification, knowledge based, contour, Hu moments, k-nearest neighbours

Citace

Jozef Baštek: Hra s ovládáním pomocí gest ruky, bakalářská práce, Brno, FIT VUT v Brně, 2009

Hra s ovládáním pomocí gest ruky

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiša. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

. . . . . .
Jozef Baštek
18. května 2009

Poděkování

Rád by som poďakoval Ing. Michalovi Hradišovi za jeho trpezlivosť a odborné rady k celému projektu.

© Jozef Baštek, 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 Spracovanie obrazu a detekcia pokožky **6**
2.1 Farebné modely 6
2.2 Morfológia obrazu 8
2.3 Bayessova klasifikacia 9

3 Modelovanie tried **11**
3.1 Background subtraction 11
3.2 Background average 11
3.3 Running average 12
3.4 Prahovanie 12
3.5 Modelovanie gaussovkou 12

4 Klasifikácia gesta **14**
4.1 Knowledge based klasifikácia 14
4.2 Kontúry a Hu momenty 15
4.3 K-nearest neighbours (KNN) 17

5 Navrhnuté riešenie **18**
5.1 Konštrukcia a použitý HW 18
5.2 Detekcia ruky a klasifikácia gesta 20
5.3 Demonštračná aplikácia 21

6 Popis implementácie a demonštračná aplikácia (hra) **23**
6.1 Štruktúra navrhnutého interface-u 23
6.2 Štruktúra hry 24

7 Vyhodnotenie prístupov a testovanie **25**
7.1 Vhodnosť jednotlivých metód spracovania obrazu a klasifikácie 25
7.2 Výber gest vhodných pre detekciu a určenie chybvosti klasifikátora 28
7.3 Odolnosť voči rušivým vplyvom a nepriaznivým podmienkam 30
7.4 Užívateľský test rozhrania 31

8 Záver **32**

A Obsah priloženého DVD **35**

B Návod na hru **36**

Seznam obrázků

1.1 Klávesnica . . . 3
1.2 Myš . . . 3
1.3 Trackball, touchpad, tablet . . . 3
1.4 Brain computer interface . . . 4
1.5 Hry pre Sony EyeToy . . . 4

2.1 RGB model . . . 6
2.2 HSV model . . . 6
2.3 HS model . . . 6
2.4 Transformácia RGB modelu na HSV model . . . 7
2.5 Transformácia RGB modelu na TSL model . . . 7
2.6 Štruktúrny element . . . 8
2.7 Efekt morfologických operácií . . . 8

3.1 Gaussova funkcia . . . 13
3.2 2D Gaussova funkcia . . . 13

4.1 Klasifikácia na základe histogramov intenzít . . . 14
4.2 Rozpoznávanie gesta ruky na základe geometrie blobu . . . 15
4.3 Algoritmus K-nearest neighbours . . . 17

5.1 Zorný uhol kamery - 50° a 80° . . . 19
5.2 Konštrukcia pre kameru . . . 19
5.3 Navrhnutý reťazec spracovania . . . 20
5.4 Páska na zápästí a jej účinok . . . 21
5.5 Obrázky z navrhnutej hry . . . 22

6.1 Stavový automat klasifikátora . . . 23

7.1 Detekcia ruky odčítavaním pozadia . . . 25
7.2 Detekcia ruky modelovaním farby kože . . . 26
7.3 Gaussov model farby kože . . . 26
7.4 Gesto I a histogramy intenzít . . . 27
7.5 [Gesto L a histogramy intenzít . . . 27
7.6 Detekcia ruky a klasifikácia gesta na základe geometrie . . . 28
7.7 Confusion matrix pre 10 gest . . . 29
7.8 Confusion matrix pre 5 gest . . . 30
7.9 Viacnásobná kalibrácia . . . 30
7.10 Úprava kontrastu a jasu obrazu pred spracovaním . . . 31

Kapitola 1

Úvod

Na počiatku vývoja prvých PC boli k dispozícií iba obmedzené možnosti použitia. Počítače boli jednoduché, rovnako ako aplikácie, ktoré na nich bežali. Keďže úlohou PC je pomáhať ľuďom, nutne musí existovať nejaký spôsob interakcie medzi užívateľom a počítačom. Napriek závratnému vývoju technológií sa metódy interakcie zmenili len veľmi málo. V prvých bežne dostupných PC bola postačujúca klávesnica na zadávanie príkazov. Tá zostala dodnes ako samozrejmá a neodmysliteľná súčasť PC. Napriek zmenám dizajnu, sa jej podstata žiadno nezmenila (obr 1.1).

Významnú úlohu vo vývoji interakcie PC s človekom zohral prechod z terminálov na grafické prostredia. Tu už klávesnica neposkytuje dostatočné možnosti ovládania a prišli nové vstupné zariadenia - myši. Podobne ako klávesnice prešli zmenami, ale iba dizajnovými. Funkčnosť sa opäť žiadno nezmenila (obr 1.2)

Po nástupe myší pribudlo ešte niekoľko ďalších vstupných zariadení, ktoré majú veľmi podobnú funkciu. Ide napríklad o trackball, tablet, touchpad a iné (obr 1.3). Žiadne z nich ale neprináša významnú inováciu do princípu práce s PC. Existujú však nové, alternatívne spôsoby ovládania, ktoré sa pomaly začínajú objavovať v rámci výskumu, i v bežných domácnostiach.

Obrázek 1.1: Vľavo: Starší typ klávesnice. Vpravo: Novší typ klávesnice. [2]

Obrázek 1.2: Vľavo: Starší typ myši. Vpravo: Novší typ myši. [3]

Obrázek 1.3: Ďalšie typy vstupných zariadení. Zľava: trackball, touchpad, tablet. [4], [7], [1]

Jedným z najnovších spôsobov interakcie človeka s PC je použitie myšlienok. V [10] je popísaný projekt ovládania robota pomocou myšlienok - špeciálny HW (tzv. Brain Computer Interface - BCI) skenuje aktivitu mozgových vĺn a správanie krvného riečiska (obr 1.4). Testy ukazujú až 90% úspešnosť s možnosťou kopmlexného ovládania robota (ruky i nohy). Podľa [10] už existujú prvé komerčne dostupné systémy vyhodnocujúce náladu človeka.

Obrázek 1.4: Brain Computer Interface na ovládanie robota. Vľavo: celý systém, Vpravo: elektródy na hlave. [10]

Ovládanie PC pomocou hlasu je dnes prístupné širokej verejnosti a má za sebou dlhší vývoj. Na trhu je už dostupné veľké množstvo elektroniky vybavenej mirkofónom a SW reagujúcim na hlasové povely - od hlasového vytáčania v mobilnom telefóne až po komplikované rozhrania schopné (takmer) plnohodnotného ovládania PC. Projekt TalkingDesktop [8] je zameraný na ovládanie PC na užívateľskej úrovni. Poskytuje možnosť diktovať text pre rôzne aplikácie, prehliadať internetové stránky, klikať na ikony a položky menu vyslovením ich názvu a podobne. Dokonca reaguje na bežné „ľudské“ otázky a vety typu „What is the price of General Motors?“ , „I want to write a letter.“ atď. Je to jeden z mnohých nástrojov, ktoré sú dnes bežne dostupné.

Ďalšou alternatívou interakcie človek - PC je ovládanie pohybmi tela. Základom je vždy kamera, ktorá sníma scénu - pohyb človeka (prípadne časti jeho tela) - a systém, ktorý vyhodnocuje rôzne udalosti na základe spracovania obrazu z kamery. Touto problematikou sa zaoberá disciplína počítačové videnie. Aj v tejto oblasti už existujú dostupné riešenia - ide hlavne o herný priemysel. Spoločnosť Sony vyvinula systém EyeToy [9], ktorý umožňuje hrať hru pohybmi tela (obr 1.5).

Obrázek 1.5: Hry pre Sony EyeToy. [9]

Cieľom tejto práce je vytvoriť práve hru ovládanú gestami ruky. Keďže ovládanie PC pohybom tela je zaujímavé a potrebný HW je dostupný (oproti napr. BCI), je možné túto technólogiu ľahšie presadiť. Práca pojednáva o základných princípoch spracovania obrazu, o rôznych metódach hľadania ruky v obraze a o metódach určenia (klasifikácie) gesta ruky. Opisuje navrhnutý 2-stupňový systém. V prvej časti je použitý Bayessov klasifikátor s jedným modelom (modelom farby kože), ktorý vyhodnocuje masku ruky. V druhej časti nastupuje klasifikácia na základe Hu invariantných momentov a štatistické zaradenie gesta do určitej skupiny (algoritmus k-nearest neighbours). V závere je zostavený systém testovaný na jednoduchej 3D hre a vyhodnotená jeho spoľahlivosť.

Po úvode sú v druhej kapitole popísané metódy spracovania obrazu a spôsoby vyhľadania ruky v obraze. Tretia kapitola pojednáva o spôsoboch modelovania tried pre Bayessov klasifikátor. Po vyhľadaní ruky je potrebné zistiť, aké gesto hráč ukazuje - túto problematiku rozoberá štvrtá kapitola. V piatej časti je navrhnuté finálne riešenie po HW i SW stránke. Nasleduje stručný popis implementácie a použitých nástrojov (OpenCV a Irrlicht Engine). Siedma kapitola sa zaoberá testovaním systému ako celku. Rozoberá jeho správanie a spoľahlivosť v rôznych podmienkach, rovnako ako reakcie ľudí na nový spôsob ovládania PC. Posledná časť - Záver - hodnotí dosiahnuté výsledky a budúci vývoj projektu. Ďalej nasleduje zoznam použitej literatúry.

Kapitola 2

Spracovanie obrazu a detekcia pokožky

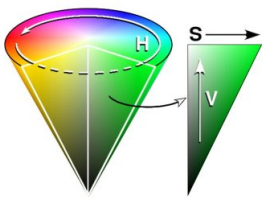
V systémoch počítačového videnia je primárne vstupné zariadenie kamera. Celý obraz získaný kamerou však neposkytuje žiadne informácie vhodné priamo pre využitie v riadených systémoch (PC, robot, automobil...). Preto je nutné obraz najprv spracovať a nájsť v ňom časti, ktoré potrebnú informáciu poskytujú. Hlavnou úlohou tejto fázy spracovania je nájdenie (detekcia) objektu záujmu, v tomto projekte teda ruky (resp. pokožky).

2.1 Farebné modely

Obraz z kamery pozostáva z elementárnych jednotiek - z bodov. Každý bod v sebe nesie jedinú informáciu - farbu (ak neuvažujeme 3D kamery a pod). Napriek tomu, že obraz principálne vzniká tým istým spôsobom, zachytené informácie môžu byť reprezentované rôzne. Jednotlivé spôsoby reprezentácie sa označujú ako farebné modely.

Nie všetky modely sú rovnako vhodné pre detekciu farby ruky, o čom pojednáva [19]. Mnohé kamery poskytujú data o obraze vo formáte RGB (red, green, blue - obr. 2.1). Podľa [19] je RGB model na detekciu ruky nevhodný, pretože má vysokú závislosť medzi jednotlivými kanálmi a mieša údaje o farbe a svietivosti (jase) do všetkých zložiek RGB. Z toho vyplýva, že pri malej zmene intenzity osvetlenia (oblak, či posun slnka na oblohe) sa môžu významne meniť zložky RGB, čo je nežiadúca vlastnosť. [19] popisuje aj ďalšie modely - normalizované RGB, YCrYCb, YUV, HSV a TSL, z ktorých práve HSV a TSL rieši spomínaný problém vysokej závislosti kanálov.

Obrázek 2.1: RGB farebý model [6]



Obrázek 2.2: HSV farebý model [5]

Obrázek 2.3: HS model - HSV so zanedbaným V kanálom

HSV model (Hue - farba, Saturation - saturácia, Value - hodnota), tiež známy ako HSI (Intensity - intenzita), alebo HSL (Lightness - svetlosť), obr. 2.2, ponúka prirodzenú predstavu farebného modelu - Hue v zmysle farby, Saturation v zmysle sýtosti a Value v zmysle svetlosti. Tento model explicitne oddeľuje farbu (HS) a svetlosť (V) - teda jas, čo je významné pre účely detekcie kože podľa farby. Zmenou intenzity osvetlenia sa (ideálne) mení iba zložka V, ktorú je možné úplne zandebať, čím sa dosiahne invariantnosť voči zmenám intenzity osvetlenia. Obr. 2.3 zobrazuje model HS, ktorý vznikol z modelu HSV zanedbaním V kanálu.

Model TSL (Tint - Odtieň, Saturation - saturácia, Ligtness - jas) má podobné vlastnosti, ale prevod modelu RGB do TSL je výpočetne náročnejší ako do HSV (obr. 2.4 a obr. 2.5).

$$
\begin{aligned}
H &= arccos\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}} \\
S &= 1-3\frac{min(R,G,B)}{R+G+B} \\
V &= \frac{1}{3}(R+G+B)
\end{aligned}
$$

Obrázek 2.4: Transformácia RGB modelu na HSV model [19]

$$
\begin{aligned}
S &= \sqrt{\frac{9}{5}(r'^2+g'^2)} \\
T &= \begin{cases} \frac{arctan(\frac{r'}{g'})}{2\pi}+\frac{1}{4} & \text{pre } g'>0 \\ \frac{arctan(\frac{r'}{g'})}{2\pi}+\frac{3}{4} & \text{pre } g'<0 \\ 0 & \text{pre } g'=0 \end{cases} \\
L &= 0.299R+0.587G+0.114B \\
&\text{kde} \\
r' &= r-\frac{1}{3} \\
g' &= g-\frac{1}{3} \\
r &= \frac{R}{R+G+B} \\
g &= \frac{G}{R+G+B}
\end{aligned}
$$

Obrázek 2.5: Transformácia RGB modelu na TSL model [19]

2.2 Morfológia obrazu

Matematická morfológia je veda, ktorá sa zaoberá analýzou priestorových štruktúr [15]. Je zameraná na analýzu tvaru a tým sa stáva vhodným nastrojom v oblasti spracovania obrazu. Môžeme ju aplikovať na binárne i šedotónové obrazy. Ďalej sa budú uvažovať iba binárne obrazy. Pomocou morfologických operátorov je možné napríklad odstrániť malé bloby (objekty), filtrovať šum, zvýrazňovať hrany, segmentovať obraz, či analyzovať textúry [15].

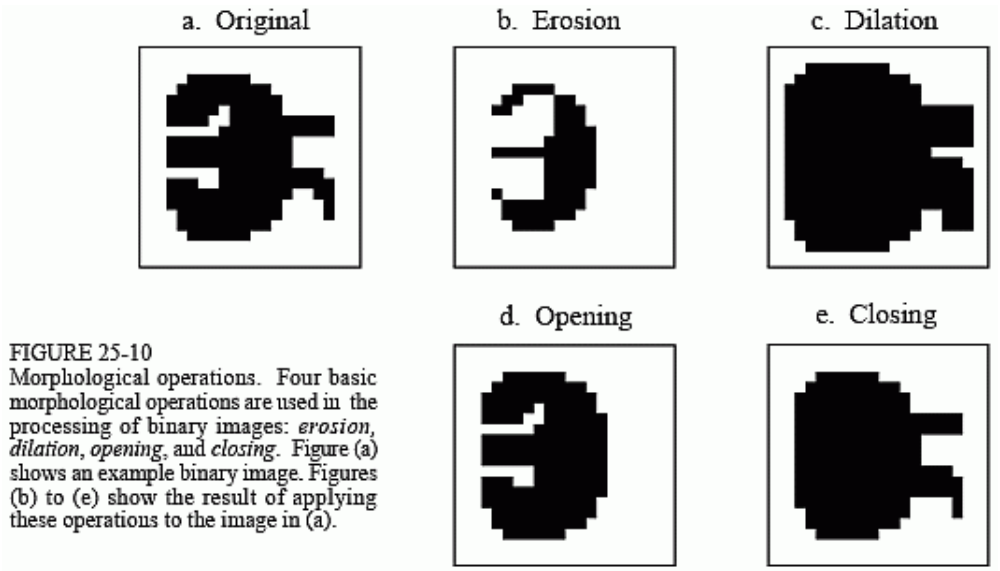
Morfologické operátory extrahujú relevantné štruktúry pomocou testovania obrazu ďalším obrazom - štruktúrnym elementom (jadrom), ktorého tvar je dopredu známy (obr. 2.6). Jadro býva oveľa menšie ako vstupný obraz a jeho tvar sa mení podľa požadovaného efektu. Okrem samotného tvaru je potrebné poznať i jeho stred (počiatok). Základné operácie - erózia a dilatácia - prebiehajú postupným umiestňovaním jadra na jednotlivé body (stred jadra = skúmaný bod) obrazu a následnou analýzou prieniku, zjednotenia a pod. Podrobný výklad morfológie je možné nájsť v [15]. Obr. 2.7 ukazuje efekt rôznych operátorov.

.	x	.
x	**X**	x
.	x	.

x	x	x
x	**X**	x
x	x	x

.	.	.
x	**X**	x
.	.	.

Obrázek 2.6: Rôzne tvary štruktúrneho elementu. Stred elementu je zvýraznený.



Obrázek 2.7: Efekt morfologických operácií na zdrojový obrázok (vľavo hore). [16]

Erózia

Erózia slúži na zmenšenie vonkajších okrajov plochy. Tým sa súčasne dosiahne zväčšenie vnútorných dier. Pomocou nej sa dajú odstrániť z obrazu malé objekty (ktoré neobsahujú jadro), prípadne oddeliť objekty, ktoré sa dotýkajú. Erózia je opačná operácia k dilatácií, ktorá objekty naopak zväčšuje. Erózia obrazu X jadrom B ($\varepsilon_B(X)$) sa dá vyjadriť ako:

$$\varepsilon_B(X) = \{x|B_x \subseteq X\} \tag{2.1}$$

kde x je skúmaný bod, B_x jadro B posunuté o vektor $x, \forall x \in X$ (tzn. jadro je umiestňované postupne do všetkých bodov obrazu X) a X je vstupný obraz. Výsledkom erózie teda budú všetky body x, ktoré splnia podmienku: po umiestnení jadra B do bodu x je B podmnožinou X.

Dilatácia

Dilatácia slúži na zväčšenie vonkajších okrajov plochy. Tým sa súčasne dosiahne zmenšenie vnútorných dier. Pomocou nej sa dajú zapĺňať malé diery v objektoch, prípadne spájať objekty, ktoré sú blízko pri sebe (do vzdialenosti veľkosti jadra). Dilatácia je opačnou operáciou k erózií. Dilatácia obrazu X jadrom B ($\delta_B(X)$) sa dá vyjadriť ako:

$$\delta_B((X) = \{x|B_x \cap X \neq 0\} \tag{2.2}$$

kde x je skúmaný bod, B_x jadro B posunuté o vektor $x, \forall x \in X$ (tzn. je jadro umiestňované postupne do všetkých bodov obrazu X) a X je vstupný obraz. Výsledkom dilatácie budú všetky body x, ktoré splnia podmienku: po umiestnení jadra B do bodu x má B neprázdny prienik s obrazom X.

Otvorenie

Erózia zmenšuje objekty a odstraňuje štruktúry, v ktorých nie je obsiahnuté jadro. Na rekonštrukciu obrazu do pôvodného stavu je možné následne použiť dilatáciu, ktorá obnoví väčšinu strateného povrchu, okrem úplne odstránených objektov. Táto sekvencia definuje operátor otvorenia, ktorý filtruje objekty v obraze (podľa tvaru jadra).

Zatvorenie

Zatvorenie je opačným operátorom k otvoreniu, teda dilatácia a následná erózia obrazu. Jedná sa o operátor, ktorým je možné vyplniť malé diery a následne zrekonštruovať čo najväčšiu časť obrazu do pôvodného stavu.

2.3 Bayessova klasifikacia

Hlavnou časťou detekcie pokožky v obraze podľa farby je rozhodnutie, či daný bod patrí do kategórie „ruka“, alebo nie. Túto úlohu môže plniť jednoduchý Bayessov klasifikátor, ktorý hodnotí pravdepodobnosť príslušnosti neznámej vzorky dát (tzv. feature vector - ďalej iba faba) k určitej triede. Táto pravdepodobnosť sa niekedy nazýva posterior. Nejedná sa o „obyčajnú“ pravdepodobnosť, ale o podmienenú pravdepodobnosť danou triedou. S obmedzením na detekciu ruky je možné ďalej uvažovať iba o dvoch triedach - pozadie a popredie (ruka). Po vyhodnotení pravdepodobností pre každý bod obrazu (a každú triedu) je možné získať žiadané objekty prahovaním - napríklad „body s farbou pokožky sú tie, ktoré majú pravdepodobnosť príslušnosti do tejto triedy viac ako 50%“.

Výsledok klasifikácie pre každú triedu môžme zapísať $P(\omega_j, X)$, kde ω_j označuje triedu a X testovanú vzorku dát. Do rovnice bayessovho klasifikátora sa dosádza funkcia rozloženia pravdepodobnosti (probability density function - PDF), ktorá popisuje (teda modeluje) rozloženie dát určitej triedy. Táto funkcia vzťahovaná k triede (class conditional PDF) sa

niekedy označuje likelihood (vierohodnosť). Ďalším predpokladom môžu byť dopredu známe pravdepodobnosti výskytu vzoriek v daných triedach (priory), ktoré sú v byessovom vzorci zohľadnené. Ak nie sú dopredu známe, dajú sa vypočítať z trénovacích dát klasifikátoru (viď [18]). Základná forma Bayessovho teorému je:

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)} \tag{2.3}$$

kde $P(\omega_j|x)$ je pravdepodobnosť, že x patrí do triedy ω_j, ak skúmame vzorku (farbu) x, $p(x|\omega_j)$ je hodnota podmienenej funkcie hustoty rozdelenia pravdepobnosti (likelihood) triedy ω_j, $P(\omega_j)$ je apriorná pravdepodobnosť triedy ω_j a $p(x)$ je evidence [21], normalizačná konštanta, ktorá nemá vplyv na vyhodnocovanie, pokiaľ je cieľom iba samotná klasifikácia. Túto konštantu vyjadruje rovnica:

$$p(x) = \sum_{j=1}^{c} p(x|\omega_j)P(\omega_j) \tag{2.4}$$

Apriorné pravdepodobnosti je možné taktiež zanedbať, pretože pri rozhodovaní nehrajú rolu. Ovplyvňujú síce hodnotu výsledku, ale rovnako pre všetky triedy, tzn. pri zmene apriorných pravdepodobností sa dá dosiahnuť rovnaký výsledok zmenou prahu, ktorým sa oddeľujú jednotlivé triedy. Podrobnejší popis a rozšírenia tohto spôsobu klasifikácie je možné nájsť v [18] a [21].

Kapitola 3

Modelovanie tried

Vstupom Bayessovho klasifikátora je funkcia rozloženia pravdepodobnosti triedy, pre ktorú vyhodnocujeme testované data (farbu).

Túto funkciu môžme nazvať modelom. Pretože model je napodobnenina systému iným systémom a tieto funkcie „napodobňujú“ rozloženeie trénovacích dát pre danú triedu, môžme ich tiež nazvať modelmi. V tejto kapitole sú popísané niektoré funkcie - modely, ktoré je možné použiť ako vstupný parameter Bayessovho klasifikátora.

3.1 Background subtraction

Z názvu vyplýva, že sa jedná o techniku, ktorá sa používa na modelovanie pozadia scény. Ako model pozadia slúži iba jeden obrázok (frame) prázdnej scény. Ten sa následne odčíta od aktuálneho obrázku. Na výsledok sa môže aplikovať prahovanie (thresholding). Princíp metódy je nasledovný (podľa [14]):

$$|frame_i - background| > Th \tag{3.1}$$

Kde $frame_i$ je aktualny snímok, $background$ je model pozadia. Ak je tento rozdiel väčší ako prah Th, daný bod nepatrí do pozadia. Výsledok tejto funkcie sa dosádza do rovnice Bayessovho klasifikátoru ako priame zaradenie do triedy pozadia (0.0 - bod nie je pozadie, 1.0 -bod je pozadie). Vynechaním prahovania a normalizáciou je možné vyjadrovať reálne hodnoty pravdepodobnosti v intervale $(0.0, 1.0)$ podľa veľkosti rozdielu. Táto metóda trpí nedostatkami - i malá zmena osvetlenia, alebo minimálny pohyb kamery môže znehodnotiť model pozadia.

3.2 Background average

Jedným zo spôsobov, ako odstrániť hlavný nedostatok predchádzajúcej metódy - nulová adaptácia modelu pozadia na zmeny (osvetlenia) - je použitie algoritmu background average [14]. Model pozadia pozostáva z priemeru posledných n snímkov. Tento fakt vytvára ďalšie dve obmedzenia:

- Veľká spotreba pamate (n * veľkosť snímku - pri $n = 15$, RGB obraze 640x480 cca. 13 MB)

- Pre správne vyhodnotenie je potrebná vysoká rýchlosť pohybu detekovaného objektu (popredia). Keďže sa model pozadia postupne priemeruje s novými snímkami, výsledný model bude po n snímkoch obsahovať 100% informácií z novej snímky a teda i popredie sa stane pozadím. Z toho taktiež vyplýva, že okrem pomaly pohybujúceho sa objektu je nemožné detekovať objekt, ktorý sa nepohybuje vôbec.

3.3 Running average

Jedná sa o ďalšie vylepšenie predchádzajúcej metódy - rieši priveľkú pamäťovú náročnosť [14]. Samotná klasifikácia opäť prebieha analogicky s algoritmom Background subtraction, pričom aktualizácia modelu pozadia prebieha podľa rovnice:

$$B_{i+1} = \alpha F_i + (1 - \alpha) B_i \tag{3.2}$$

kde B_{i+1} je nový model pozadia, α je tzv. koeficient učenia, F_i je aktuálny snímok a B_i je predchádzajúci model pozadia. α má analogickú funkciu ako n v algoritme Background average. Problémy spojené s potrebou dostatočnej rýchlosti detekovaného objektu táto metóda neodstraňuje. α máva hodnotu obyčajne 0.05.

3.4 Prahovanie

Prahovanie jednotlivých kanálov obrazu podľa [14] opäť poskytuje jednoznačné zaradenie testovaného bodu do skúmanej triedy. Táto metóda sa dá použiť na modelovanie akejkoľvek triedy podľa farby. Funkcia obsahuje skupinu pravidiel - ak testovaný bod vyhovuje všetkým pravidlám, výsledok bude 1.0 (100% patrí do tejto skupiny), ak bod pravidlám nevyhovuje, výsledok bude 0.0 (nepatrí do tejto skupiny). Jeden možný set pravidiel v priestore RGB, ktorý popisuje farbu pokožky, môže byť napríklad [14]:

$$\begin{gathered} R > 95 \quad \wedge \quad G > 40 \quad \wedge \quad B > 20 \quad \wedge \\ max(R, G, B) - min(R, G, B) > 15 \quad \wedge \\ |R - G| > 15 \quad \wedge \quad R > G \quad \wedge \quad R > B \end{gathered}$$

3.5 Modelovanie gaussovkou

Komplexnejší prístup k modelovaniu popredia podľa farby poskytuje gaussovská (normálna) funkcia rozloženia pravdepodobnosti [19], [18]:

$$p(c|skin) = \frac{1}{2\pi\sqrt{|\Sigma_s|}} e^{-\frac{1}{2}(c-\mu_s)^T \Sigma_s^{-1}(c-\mu_s)} \tag{3.3}$$

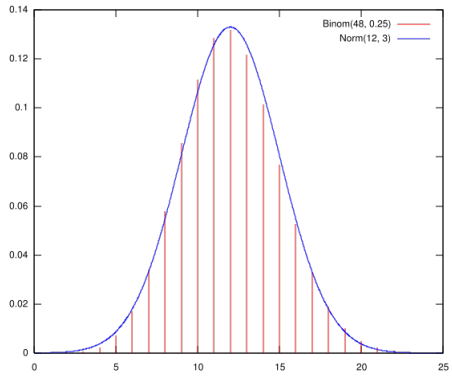
Kde c je testovaná farba (vektor farby), $skin$ je názov triedy (koža), μ_s a Σ_s sú parametre gaussovskej distribúcie - μ_s je stredná hodnota a Σ_s je tzv. kovariančná matica. Tieto parametre je možné získať z trénovacích dát nasledovne:

$$\mu_s = \frac{1}{n}\sum_{j=1}^{n} c_j \tag{3.4}$$

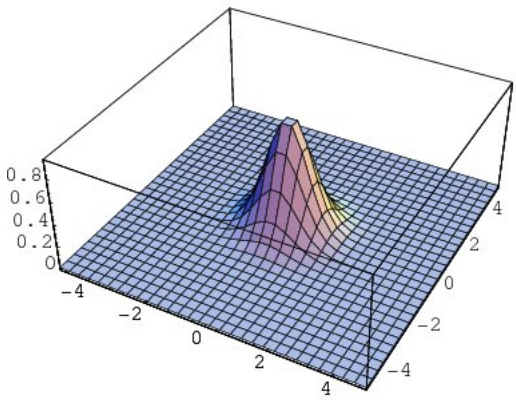
$$\Sigma_x = \frac{1}{1-n}\sum_{j=1}^{n}(c_j - \mu_s)(c_j - \mu_s)^T \tag{3.5}$$

Kde n je celkový počet vzoriek a c je vzorka z trénovacích dát. Základom trénovacích dát môže byť normalizovaný histogram. Správne nastavenie parametrov rozloženia vytvorí funkciu, ktorá pokryje tento histogram. 1D trénovacie data (šedotónový obraz) môžeme opísať gaussovkou podľa obr. 3.1. Pre 2D modely (napr. v priestore HS) sa použie 2D gaussovka [20], obr. 3.2.

Po vypočítaní parametrov funckie gaussovského rozloženia môžeme do modelu dosadzovať testované data (farbu c). Výsledok tejto funkcie môže byť dosadený do Bayessovho vzorca.



Obrázek 3.1: 1-rozmerná gaussova funkcia [8]



Obrázek 3.2: 2-rozmerná gaussova funkcia [20]

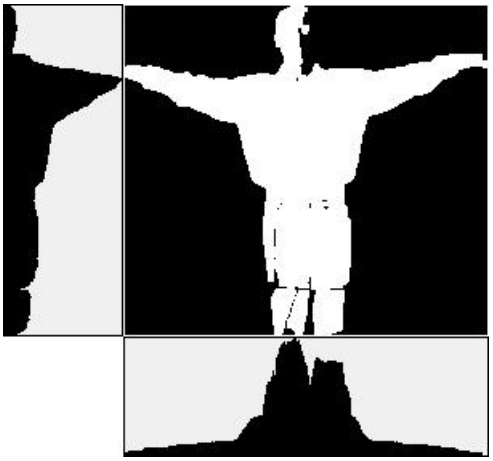
Kapitola 4

Klasifikácia gesta

Samotné rozdelenie bodov obrazu do jednotlivých tried neposkytuje dostatočné informácie na ovládanie zložitejšieho systému (hry, robota...). Preto je nutné vybrať triedy(u), ktoré reprezentujú objekt záujmu („popredie“, „ruka“, „jablko“) a ďalej tento objekt klasifikovať podľa ďalších vlastnosti - veľkosť, tvar a pod. Z toho vyplýva nutnosť vytvorenia nových tried, do ktorých sa bude objekt zaraďovať („gesto 1“, „gesto 2“, „veľké jablko“, „zelené jablko“).

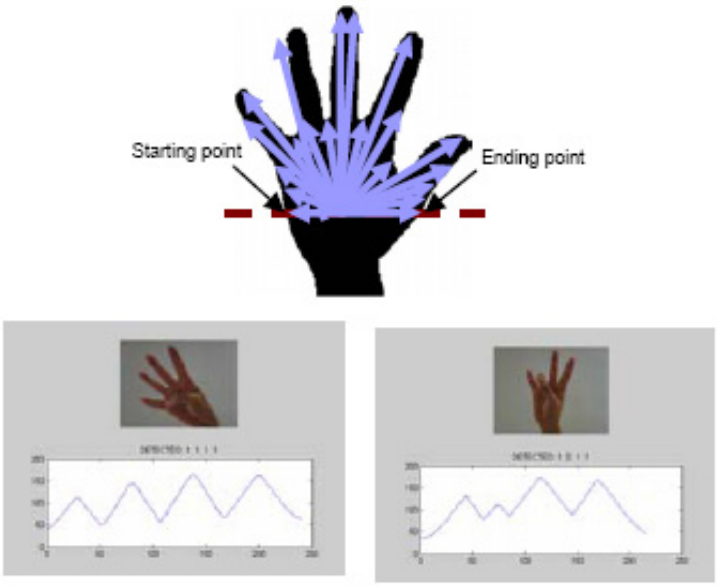
4.1 Knowledge based klasifikácia

V práci [13] sa pojednáva o klasifikácií gesta na základe tvaru horizontálnych a vertikálnych histogramov intenzít. V tomto prípade ide o „gestá celého tela“ - ľavá / pravá ruka pri tele, rovno do strany alebo šikmo hore. Vlastné rozhodovanie je založené na skupine výrazných geometrických čŕt histogramov, ktoré sa dajú ľahko rozpoznať pozorovaním (preto knowledge based). Na obr. 4.1 je možné rozdeliť horizontálny histogram na tretiny a pomocou 2 prahov zistiť, že v ľavej a pravej časti je nízka úroveň intenzity, v strdnej časti vysoká. Podobne je možné rozdeliť i vertikálny histogram. Tieto fakty jednoznačne identifikujú zobrazené gesto medzi ostatnými. Histogramy pred samotnou analýzou je možné predspracovať - napríklad prahovaním na jednoznačné určenie napr. 3 úrovní intenzít.



Obrázek 4.1: Gesto s roztiahnutými rukami. Vpravo hore maska človeka, vľavo vertikálny histogram intenzity, dole horizontálny histogram intenzity. [13]

Alternatívny prístup, ktorý nepracuje s tvarom histogramov, ale priamo s geometriou detekovaného blobu (objektu záujmu) a s jeho význačnými črtami popisuje práca [17]. Klasifikácia je obmedzená na malú skupinu gest ruky (čísla). Sledujú sa maximá a minimá vzdialenosti kontúry blobu (viď ďalej) od jeho stredu - obr. 4.2.



Obrázek 4.2: Rozpoznávanie gesta ruky na základe geometrie blobu. Hore - určenie vzdialeností okrajových bodov od stredu dlane. Dole - graf reprezentujúci vzdialenosti jednotlivých bodov kontúry. [17]

4.2 Kontúry a Hu momenty

Po získaní binárneho o brazu objektu záujmu, je možné rastrové údaje transformovať do vektorovej podoby pre ďalšiu (matematickú) analýzu. Metódy transformácie sú popísané v [11].

Kontúra je zoznam bodov, ktorý popisuje geometricky obvod blobu, prípadne jeho dier. Pomocou nej je možné ďalej parametrizovať daný blob. Základnú metriku, ktorú je možné získať, je napríklad obvod blobu (resp. dĺžka kontúry). Komplexnejšími výpočtami je možné blob aproximovať polygonom, alebo mu vpísať (opísať) štvorec, obdĺžnik, kruh, či elipsu, čo sa časo využíva napr. na zvýraznenie oblasti, ktorá bola v obraze detekovaná ako objekt záujmu. Z geometrie opísanej elipsy je možné určiť pomer strán blobu a otočenie voči začiatku súradnicovej sústavy.

Významným využitím kontúr je možnosť ich porovnávania so šablonami (nejedná sa o template matching) ako metóda klasifikácie do rôznych tried. Kontúry je možné porovnávať podľa tzv. momentov. Moment je charakteristika kontúry, ktorá sa vypočíta integráciou (resp. sčítaním) cez všetky body kontúry podľa rovnice

$$m_{p,q} = \sum_{j=1}^{n} I(x,y)x^p y^q \tag{4.1}$$

kde n je počet bodov kontúry, $I(x, y)$ vyjadruje hodnotu bodu kontúry (1.0), p je x-stupeň momentu, q je y-stupeň momentu, pričom prvky v súčte sú umocňované príslušnými stupňami.

Z rovnice 4.1 vyplýva, že $m_{0,0}$ je vlastne dĺžka kontúry (za predpokladu, že uvažujeme kontúru hrubú 1 bod [11]). V praxi sa používajú momenty do 3. stupňa.

Takto vypočítaný moment stále nie je vhodný na porovnávanie kontúr, pretože jeho výsledok je závislý na polohe kontúry (i na ďalších vlastnostiach).

Z tohto dôvodu sa zavádzajú centrálne a normalizované momenty. Centrálne momenty sú momenty, v ktorých vystupujú prvky upravené strednou hodnotou kontúry (rovnica 4.2). Normalizované momenty vychádzajú z centralných, majú však naviac normalizačný koeficient, ktorý odstraňuje závislosť výslednej hodnoty od celkovej veľkosti kontúry (scale) (rovnica 4.3).

$$\mu_{p,q} = \sum_{j=1}^{n} I(x, y)(x - x_{avg})^p (y - y_{avg})^q \tag{4.2}$$

kde $x_{avg} = \frac{m_{1,0}}{m_{0,0}}$ a $y_{avg} = \frac{m_{0,1}}{m_{0,0}}$, teda $[y_{avg}, y_{avg}]$ je geometrický stred kontúry.

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{0,0}^{(p+q)/2+1}} \tag{4.3}$$

kde $\mu_{p,q}$ je centralálny moment a $m_{0,0}^{(p+q)/2+1}$ je normalizačná konštanta.

Významným typom momentov sú tzv. Hu invariantné momenty, ktoré sú lineárnou kombináciou centrálnych momentov. Je matematicky dokázané, že tieto hodnoty sú invariantné voči rotácií, posunutiu a veľkosti , čo je kľúčová požiadavka pri klasifikácií objektov podľa tvaru. Výpočet 7 Hu momentov ukazuje rovnica 4.4.

$$\begin{aligned}
h_1 &= \eta_{2,0} + \eta_{0,2} \\
h_2 &= (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2 \\
h_3 &= (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2 \\
h_4 &= (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2 \\
h_5 &= (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})((\eta_{3,0} - \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2) + \\
&\quad (3\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2) \\
h_6 &= (\eta_{2,0} - \eta_{0,2})((\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2) + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3}) \\
h_7 &= (3\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2) \\
&\quad -(\eta_{3,0} - 3\eta_{1,2})(\eta_{2,1} + \eta_{0,3})(3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2)
\end{aligned} \tag{4.4}$$

Na základe týchto metrík už je možné určiť spoľahlivo podobnosť dvoch kontúr. Opäť existuje niekoľko spôsobov, ako vypočítať celkový „stupeň podobnosti“. V [11] sú popísané 3 spôsoby:

$$I_1(A,B) = \sum_{i=1}^{7} |\frac{1}{m_i^A} - \frac{1}{m_i^B}| \tag{4.5}$$

$$I_2(A,B) = \sum_{i=1}^{7} |m_i^A - m_i^B| \tag{4.6}$$

$$I_2(A,B) = \sum_{i=1}^{7} |\frac{m_i^A - m_i^B}{m_i^A}| \tag{4.7}$$

kde

$$\begin{aligned} m_i^A &= sign(h_i^A) log|h_i^A| \\ m_i^B &= sign(h_i^B) log|h_i^B| \end{aligned}$$

4.3 K-nearest neighbours (KNN)

KNN algoritmus je metóda klasifikácie z kategórie algoritmov s asistovaným učením (supervised learning) [12]. Jeho úlohou je určiť, do ktorej triedy objektov z trénovacích (vzorových) dát patrí testovaný objekt s najväčšou pravdepodobnosťou. Algoritmus (obr. 4.3, [12]) je možné adaptovať takmer na akúkoľvek úlohu, keďže výpočet metrík môže byť ľubovoľný. Pri veľkom počte trénovacích dát, prípadne pri zložitejších metrikách môže byť značne výpočetne náročný.

- Urči parameter k - počet najbližších susedov
- Vypočítaj vzdialenosť (rozdiel) medzi testovaným objektom a všetkými objektami v trénovacej sade dát
- Zoraď rozdiely vzostupne
- Testovaný objekt zaraď do skupiny, ktorá sa vyskytuje v prvých k najmenších rozdieloch najviac krát

Obrázek 4.3: Algoritmus KNN [12]

Kapitola 5

Navrhnuté riešenie

Na základe zadania bolo vytýčených niekoľko hlavných cieľov. Z hľadiska univerzálnosti vytvoriť klasifikátor, ktorý rozpozná čo najväčší počet gest ruky (aspoň 10) a bude čo najodolnejší voči nepriaznivým vplyvom. Systém by mal spoľahlivo pracovať s prirodzeným i umelým osvetlením. Keďže sa jedná o detekciu gest ruky, postačujúca bude statická kamera namierená na stôl. Gestá sa budú prevádzať rukou položenou na stole. Je môžné uvažovať so sjednodušením v podobe jednofarebného pozadia.

Návrh aplikácie demonštrujúcej interfejs by nemal príliš „zaostávať“ za aplikáciami dnešnej doby. Ďalším dôvodom vytýčenia tohto cieľa je užívateľská prívetivosť a zaujímavosť. Keďže v súčastnosti nie je možnosť interfejs otestovať v reálnej (vedeckej) aplikácií, vytvorená hra by mala otestovať presnosť a intuitívnosť ovládania.

5.1 Konštrukcia a použitý HW

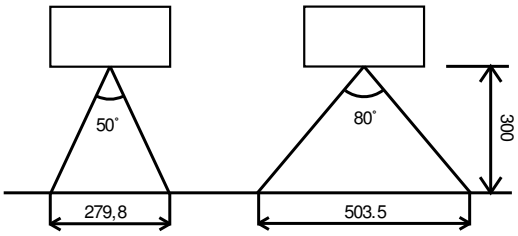
Keďže sa dá predpokladať, že v blízkej budúcnosti bude ovládanie PC pomocou gest viac rozšírené a hlavne do domácností, hlavný dôraz bol kladený na použiteľnosť, kompaktnosť a ekonomickosť riešenia. Na trhu existuje veľmi veľa rôznych kamier, od lacných webových, cez drahé modely s IR osvetlením, dokonca i 3D web-kamery pre domáce použitie, až po profesionálne prístroje v hodnote (desiatok) tisíc korún. Kedže riešenie má byť čo najlacnejšie, výber kamier sa obmedzil iba na webové kamery. Ďalej sa nebrali do úvahy najlacnejšie kamery, ktoré svojimi parametrami nespĺňali vhodnú úroveň kvality snímaného obrazu (neostrosť, digitálny šum, slabý kontrast, slabá možnosť nastavenia kamery...). Zo zbytku kamier na našom trhu jednoznačne zvíťazila kamera ASUS AF-200 pre niekoľko podstatných vlastností, ktoré žiadne iné kamery neponúkali.

V prvom rade ide o zorný uhol. V dobe riešenia tejto práce bola AF-200 na trhu jediná širokouhlá webová kamera. Zorný uhol bežných kamier je 50°, čo je oveľa menej oproti 80° ponúkaných touto kamerou. To priamo nahráva požiadavke kompaktnosti riešenia, pretože širší zorný uhol významne znižuje výšku potrebnej konštrukcie na zachytenie rovnakej plochy stola - viď obr. 5.1.

Ďalej je táto kamera vybavená 2 Mpix CMOS senzorom, ktorý prakticky vôbec netrpí digitálnym šumom a technológiou bright vision, vďaka ktorej kamera sníma v takmer uplnej tme kvalitné snímky. To všetko je podporené automatickým zaostrovaním (3cm až nekonečno) a skutočne ostrým obrazom za každých podmienok.

Konštrukcia - stojan pre kameru - má zásadný vplyv na kompaktnosť riešenia. Hlavným cieľom bola možnosť rýchlej manipulácie bez použitia akýchkoľvek nástrojov (skrutkovač

a pod). Ideálne by konštrukcia mala byť dokonca prispôsobiteľná rôznym požiadavkam. Všetky tieto ciele sa podarilo splniť vďaka jednoduchému nápadu - použiť unverzálne nastaviteľné držiaky na police. Použitím dvojíc držiakov pre každú časť konštrukcie je možné meniť šírku i výšku takmer neobmedzene, pri nedostatočných rozmeroch je možné použiť tretiu lištu. Podobne nastaviteľné sú drevené podpery pre zvýšenie stability konštrukcie. Potrebu nástrojov sa podarilo odstrániť krídlovými maticami s veľkými podložkami. Obr. 5.2 ukazuje detaily konštrukcie.

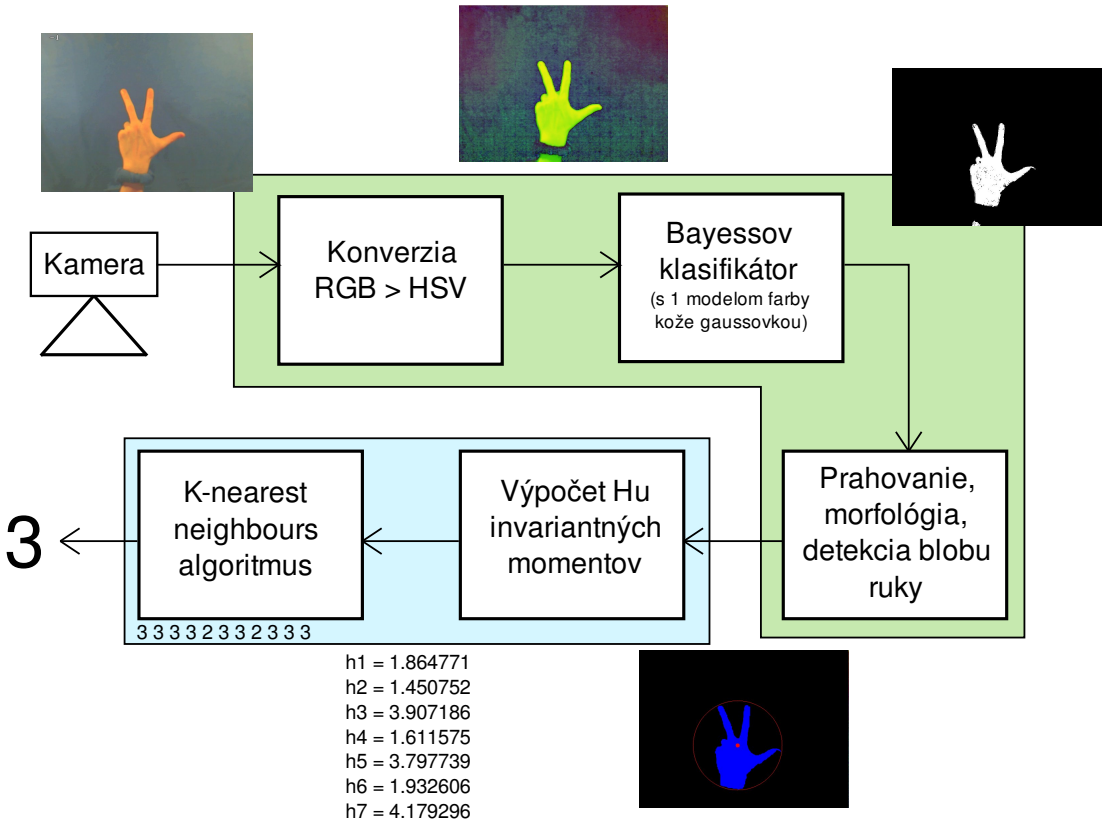


Obrázek 5.1: Zorný uhol kamery - 50° a 80° a veľkosť zachytenej plochy.

Obrázek 5.2: Konštrukcia pre kameru. Vľavo hore krídlová matica s podložkou pre ľahkú manipuláciu iba s prstami, vľavo dole „L-diel“ držiaci kameru, vpravo hore celý stojan, vpravo dole rozložená konštrukcia.

5.2 Detekcia ruky a klasifikácia gesta

Po preskúmaní rôznych metód bol navrhnutý reťazec spracovania obrazu tak, aby poskytoval čo najlepšie výsledky. Spracovanie obrazu je rozdelené do 2 častí. Vývojový diagram ukazuje obr. 5.3.



Obrázek 5.3: Reťazec spracovania obrazu. V zelenej časti prebieha detekcia ruky, v modrej časti prebieha klasifikácia gesta. Obrázky znázorňujú výsledky každej časti spracovania. Čísla pod K-nearest neigbours klasifikátorom predstavujú n najpodobnejších gest z trénovacích dát.

V prvej časti prebieha rozpoznanie ruky. Po získaní obrazu z kamery vo formáte RGB nasleduje konverzia do farebného priestoru HS(V). Ďalej Bayessov klasifikátor určí, ktoré body patria do pozadia a ktoré patria popredíu - ruke. Kedže bola zvolená metóda modelovania popredia gaussovým rozložením, pred použitím systému je nutná kalibrácia. Touto kalibráciou sa získajú trénovacie data, na základe ktorých sa vypočítajú parametre gaussovej funkcie rozloženia pravdepodobnosti. Nakoľko apriorné pravdepodobnosti nemajú vplyv na výsledok klasifikácie, v bayessovom klasifikátore nevystupujú. Model pozadia je taktiež zanedbaný - resp. nahradený konšntanou funkciou s nízkou hodnotou. Ignorovaním modelu pozadia sa dosiahla nezávislosť vyhodnocovania na farbe a type pozadia.

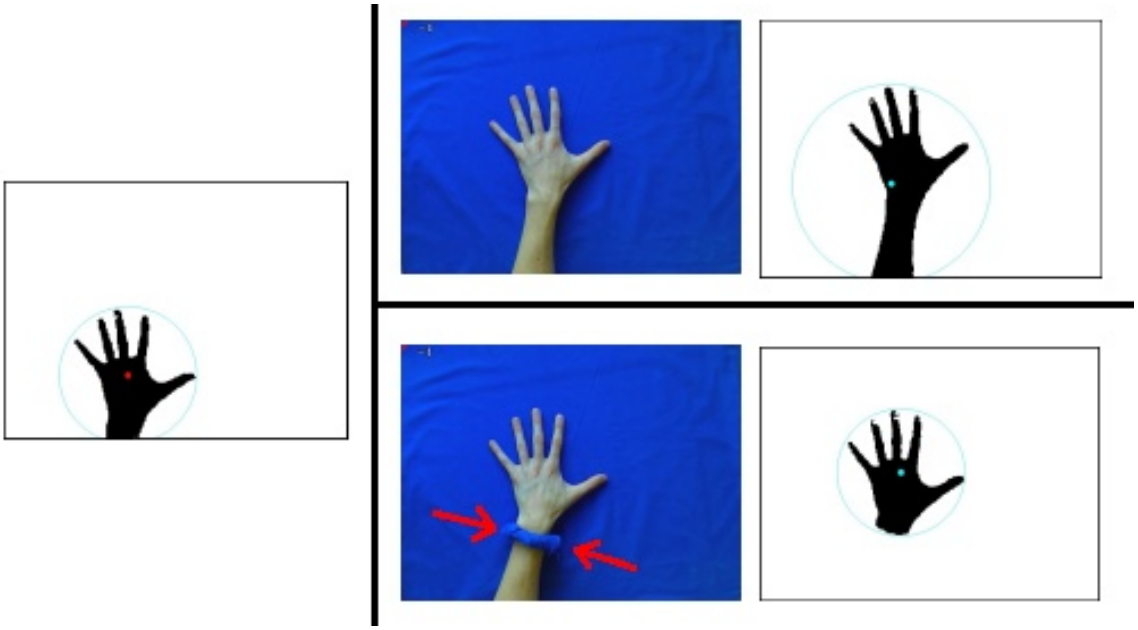
Výsledok prvotného rozdelenia bodov obrazu na „body pozadia“ a „body ruky“ sa upravuje morfologickými operáciami - eróziou sa odstrani šum, následnou dilatáciou sa kompenzuje strata povrchu blobov, ktoré neboli úplne odstránene eróziou. Z obrazu sa vyberie najväčší blob, resp. najčlenitejší - s najvašším počtom bodov v kontúre (za predpokladu splnenia podmienky, že ruka je najčlenitejší objekt s farbou pokožky, ide o významné

zvýšenie výkonu oproti metóde počítania povrchu blobu za účelom nájdenia „najväčšieho“ blobu). Ten sa ďalej rozmaže, čo spôsobí vyhladenie „zubatých“ okrajov blobov. Následné prahovanie s prahom 1 opäť vytvorí binárnu masku ruky.

Binárny obraz skúmanej masky postupuje do druhej časti, kde prebieha klasifikácia gesta a výpočet ďalších vlastností. Výpočtom Hu invariantných momentov sa získajú parametre blobu, na základe ktorých prebieha K-nearest neigbours klasifikácia. Počas trénovania klasifikátora vznikla databáza Hu-momentov pre jednotlivé gestá, nad ktorou pracuje KNN klasifikátor. Jeho výstupom je konkrétny typ gesta. Okrem typu gesta sa ďalej blob parametrizuje jeho stredom (geometrickým) a veľkosťou.

Z definície Hu momentov vyplýva, že bloby rôznych tvarov majú rôzne hodnoty momentov. Detekovaný blob ruky však významne mení svoj tvar pohybom ruky v smere osi y. Tzn. napriek tomu, že ruka ukazuje to isté gesto, momenty celého blobu sú uplne odlišné a preto v tomto prípade detekcia na základe Hu momentov zlyhá. Tento problém je vyriešený páskou na zápästí (najlepšie páskou farby pozadia), ktorá zabezpečí relatívne rovnaký tvar blobu (viď obr. 5.4). Toto riešenie ďalej vyžaduje úpravu detekcie blobu ruky tak, aby bol označený blob, ktorý je najvyššie v obraze.

Pre zvýšenie spoľahlivosti klasifikácie je ukázané gesto potvrdené až po tom, ako sa vyhodnotí 5x po sebe rovnako. To odstraňuje náhodné falošné zaradenia do nesprávnej skupiny.



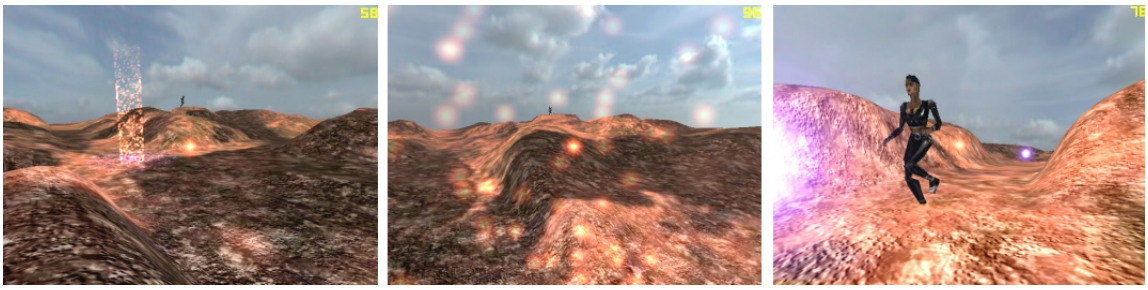
Obrázek 5.4: Účinok pásky na zápästí pri detekcí gesta. Vľavo požadované gesto. Hore ruka pohnutá v smere osi y a detekovaný blob, ktorý je veľmi odlišný od požadovaného. Dole ruka s páskou na zápästí a detekovaný blob, ktorý je veľmi podobný požadovanému.

5.3 Demonštračná aplikácia

Demoaplikácia - teda hra - by mala byť niečim zaujímavá. Cieľ bol jasný - jednoznačne by malo ísť o 3D hru. Na tejto hre by malo byť možné plne otestovať intuitívnosť ovládania a takisto rýchlosť odozvy systému na reakcie užívateľa. Z mnohých nápadov bol nakoniec

uskutočnený klasický námet - outdoorová FPS (first person shooter) hra. Interface by mal poskytovať plnohodnotné informácie, na základe ktorých sa dá vytvoriť ovládanie, ktoré zodpovedná bežným hrám, teda pohyb do všetkých strán, otáčanie a zmena smeru pohľadu, strieľanie, prípadne ďalšie rozšírenia týchto štandardných úkonov.

Navrhnutá hra je situovaná do otvorenej krajiny. Hlavnou úlohou je zastrelenie všetkých pohybujúcich sa robotov. Pre demonštráciu použitia väčšieho počtu gest ruky bol navrhnutý „výťah“. Jeho výšku je možné ovládať kombináciu gest. Slúži na „rozhliadnutie sa“ po krajine, kde sú prípadne ďalšie roboty.



Obrázek 5.5: Obrázky z hry. Zľava: pohľad z kopca na výťah, vo výťahu, robot s teleportami

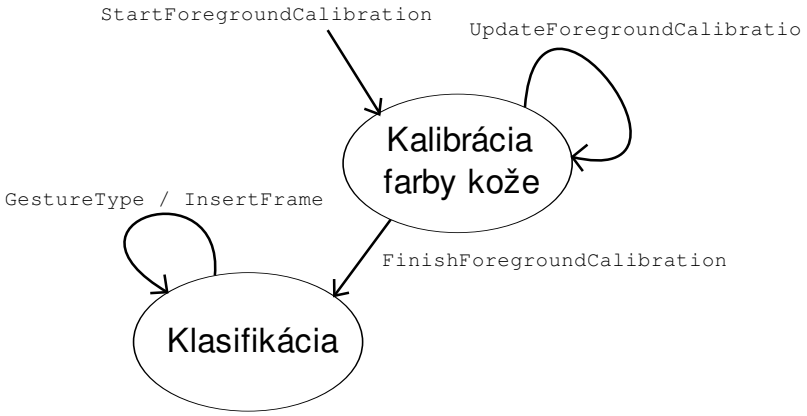
Kapitola 6

Popis implementácie a demonštračná aplikácia (hra)

Táto kapitola stručne popisuje implementáciu klasifikátora, hry a použité nástroje.

6.1 Štruktúra navrhnutého interface-u

Interface je navrhnutý technológiou OOP. Využíva funkcie knižnice Open Computer Vision Library (OpenCV), ktorá je určená na prácu s obrazom, obsahuje mnoho bežne používaných algoritmov a podporuje vývoj aplikácií z oblasti počítačového videnia. Po vytvorení objektu detektora sa dá popísať jeho „život“ jednoduchým stavovým automatom (obr. 6.1).



Obrázek 6.1: Stavový automat klasifikátora

Pred použitím systému je nutná kalibrácia klasifikátora. Tá sa začína volaním metódy `StartForegroundCalibration(IplImage *mask)`, kde `mask` reprezentuje masku oblasti, z ktorej sa budú snímať trénovacie data. Po začatí kalibrácie je nutné vložiť do systému niekoľko obrázkov trénovacích dát pomocou metódy `InsertFrame(IplImage *img)`. Ak kontrola kalibrácie ukáže nedostatky - teda diery v detekovanom blobe (viď kapitola Odolnost voči rušivým vplyvom), je možné zahájiť aktualizáciu modelu popredia volaním metódy `UpdateForegroundCalibration(IplImage *mask)`, kde `mask` reprezentuje novú masku oblasti, z ktorej sa budú snímať trénovacie data. Ak je kalibrácia dostatočná, je nutné ukončiť túto časť metódou `FinishForegroundCalibration(void)`.

Po skončení kalibrácie je potrebné vypočítať model popredia (`ComputeGauss()`), nastaviť prah, ktorý určuje minimálnu pravdepodobnosť príslušnosti bodu k triede „ruka“ metódou `SetGaussThreshold(double thr)` a načítať šablony, podľa ktorých sa štatisticky vyhodnocujú typy ukázaných gest `LoadTemplates(void)`.

Počas behu klasifikátora sa vkladajú snímky na vyhodnocovanie opäť metódou `InsertFrame(IplImage *img)`. Samotné spracovanie obrazu a klasifikácia gesta prebieha až na požiadanie metódou `int GestureType(int k, CvPoint *p)`, kde `k` je parameter algoritmu KNN, `p` je výstupný parameter, ktorý bude obsahovať geometrický stred detekovanej ruky a návratová hodnota funkcie reprezentuje číslo detekovaného gesta.

Pre maximálne využitie výkonu dnešných viacjadrových procesorov beží detekcia ruky v samostanom vlákne. V druhom vlákne beží demonštračná aplikácia - hra.

6.2 Štruktúra hry

Demonštračná aplikácia bola tiež realizovaná technológiou OOP, nakoľko vychádza z 3D herného enginu Irrlicht Engine. Kedže hra slúži iba na demonštračné účely, je vytvorená na základe vzorových projektov, ktoré sa nainštalujú s týmto enginom. Irrlicht engine poskytuje silnú podporu pre všetky bežné stratégie programovania hier. Pomocou niekoľkých riadkov kódu je možné vytvoriť terén krajiny a FPS kameru, na ktorú pôsobí gravitácia. Ďalej boli zo vzorových príkladov použité modely robotov s animáciami chôdze. Irrlicht engiine poskytuje podporu detekcie kolízí, preto bolo do hry zakomponované strielanie na tieto roboty. Cieľom hry je zastreliť všetky roboty. Prípadný dotyk robota s hráčom znamená koniec hry.

Renderovanie grafiky beží v samostatnom vlákne. Ovládanie prebieha asynchrónne pomocou prepínania stavových premenných. Na základe týchto premenných sa posielajú eventy Irrlicht enginu.

Kapitola 7

Vyhodnotenie prístupov a testovanie

V tejto kapitole sú stručne popísané výsledky metód spracovania obrazu a klasifikácie gesta spomenutých v doterajšom texte.

7.1 Vhodnosť jednotlivých metód spracovania obrazu a klasifikácie

Detekcia pokožky

V prvej časti vyhodnocovania sa nachádza detekcia ruky. Základný algoritmus odčítavania pozadia (background subtraction) poskytuje najvyšší výkon, no v tejto aplikácií nespĺňa požiadavku odolnosti voči zmenám intenzity osvetlenia. Okrem osvetlenia hrajú veľkú rolu tiene samotnej ruky, pretože v praxi (v domácnosti) nie je možné dosiahnuť konštantného osvetlenia z každej strany. Algoritmy background average a running average nie sú vhodné principiálne - po určitej (veľmi krátkej) dobe sa popredie stáva pozadím.

V ďalšej časti spracovania - určenie blobu, ktorý reprezentuje ruku a klasifikácia gesta - majú tieto metódy ďalšiu nevýhodu. Tá je zapríčinená tým, že metódy odčítania pozadia rozpoznávajú iba 2 triedy - „pozadie“ a „všetko ostatné“ a nerozdeľujú podľa farby (teda selektívne vyhľadávajú pozadie, nie objekt záujmu v tomto prípade). Ak sa na scéne objavia 2 rôzne objekty, ktoré sa prekrývajú, následná analýza tvaru blobu je znehodnotená (odbr. 7.1).

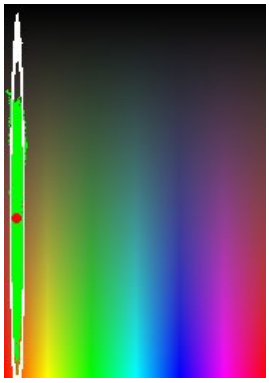
Obrázek 7.1: Detekcia ruky odčítavaním pozadia. Pozadie tvorí jednu triedu, všetky ostatné objekty sú zaradené do druhej. Zľava: model pozadia, scéna s niekoľkými objektami, detekovný blob - napriek odlišnej farbe objektov sa nedá získať informácia o tvare ruky. Problém nastáva pri prekrytí s akýmkoľvek objektom.

Metódy modelovania triedy podľa farby (kože) sa ukázali ako vhodnejší postup, keďže selektívne hľadajú objekt záujmu podľa jeho vlastností. Zvláštny prípad existencie viacerých objektov v obraze sa obmedzuje na problematickú existenciu 2 objektov farby kože, ktoré sa prekrývajú (obr. 7.2).

Obrázek 7.2: Detekcia ruky s modelovaním triedy podľa farby (kože) gaussovkou. Body s farbou kože tvoria jednu skupinu, ostatné body druhú skupinu. Zľava: scéna s niekoľkými objektami, dekcia pokožky, výsledný blob - objekty odlišnej farby nemajú vplyv na detekciu, problém nastáva pri prekrytí s ďalším objetom farby kože (druhá ruka).

Jednoduché prahovanie kanálov obrazu (či RGB,alebo HS) vykazuje oveľa lepšie výsledky ako predchádzajúce algoritmy a stále poskytuje vysokú rýchlosť. S pravidlami podľa [19] poskytuje pomerne dobré výsledky, no stále sa objavuje veľa falošných a nesprávnych vyhodnotení.

Poslednou testovanou metódou je modelovanie farby kože gaussovským rozložením pravdepodobnosti. Napriek tomu, že je potrebná kalibrácia (obr. 7.3) - a teda užívateľov zásah - vykazuje jednoznačne najlepšie výsledky a odolnosť voči náhlym zmenám intenzity osvetlenia. Rozdiely v osvetlení a prípadné tiene na ruke, môžu byť kompenzované viacnásobnou kalibráciou (viď kap. Odolnosť voči rušivým vplyvom a nepriaznivým podmienkam), čo významne zvyšuje kvalitu detekcie ruky.



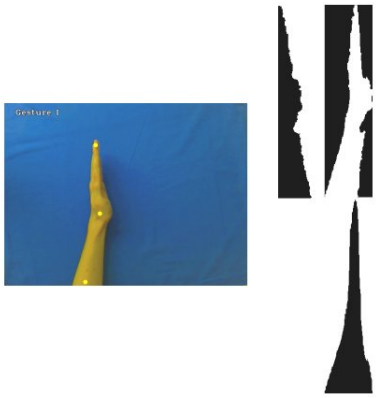
Obrázek 7.3: Výsledok kalibrácie: Gaussova funkcia v grafe hue x sat. Zelená farba označuje výskyt aspoň jedného bodu v trénovacich datach s príslušnou farbou, červený bod označuje strednú hodnotu gaussovej funkcie (μ), biela elipsa označuje interval pravdepodobnosti 10...15%, že príslušná farba patrí do skupiny „farba kože“.

Klasifikácia gesta

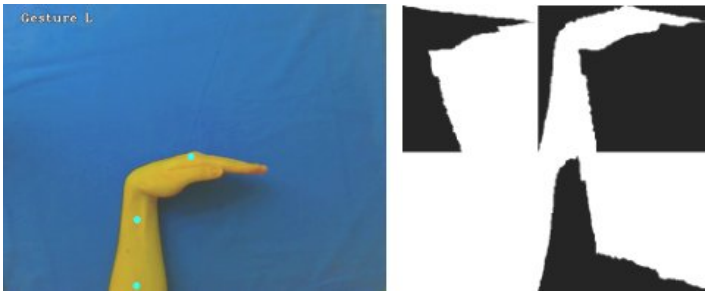
V ďalšej časti spracovania sa nachádza klasifikácia gesta. Knowledge-based metódy sú jednoduché, no ich jednoduchosť je vykúpená obmedzeným množstvom (relatívne spoľahlivo)

rozpoznaných gest. Metóda analýzy histogramov je jednoduchá, ale obmedzená na malý počet výrazných gest. Najlepšie vyhodnocuje „pravouhlé gestá“. Natočenie ruky prakticky znemožňuje správne vyhodnotenie gesta. Taktiež neumožňuje spoľahlivé vyhodnotenie gest s rôznymi prstami, keďže v horizontálnom histograme intenzity sa objaví súčet napr. 2 prstov. Obr. 7.4 a 7.5 ukazujú 2 gestá:

- I gesto: Vertikálny histogram má klesajúcu tendenciu smerom zdola nahor (prsty by mali byť užšie ako šírka zápästia). Horizontálny histogram zapĺňa celú svoju šírku vysokými intenzitami, pričom smerom k stredu sú intenzity najvyššie (obr. 7.4).
- L gesto: na vetikálnom histograme je výrazná oblasť vyšších hodnôt intenzity v hornej časti, spodná časť histogramu má nízke hodnoty. Horizontálny histogram má podobný tvar, pričom z polohy vyšších intenzít (vľavo alebo vpravo) sa dá presnejšie určiť, či je gesto L doprava, alebo doľava (celá časť ruky na jednej strane, úzke prsty na druhej) (obr. 7.5).



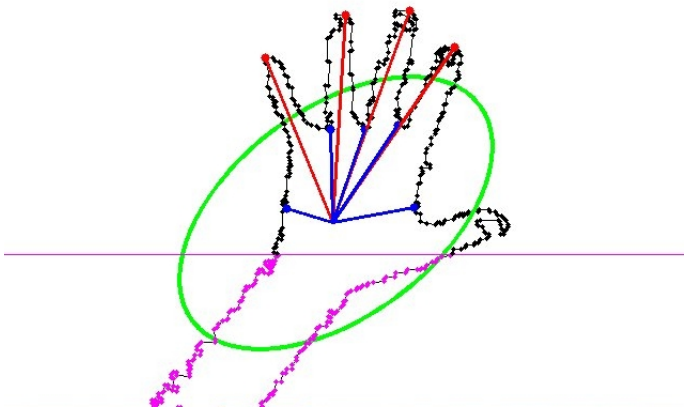
Obrázek 7.4: Gesto I. Vľavo ukázané gesto. Vpravo trojica obrázkov - pravo hore detekovaný blob, vľavo vertikálny histogram intenzít, dole horizontálny histogram intenzít.



Obrázek 7.5: Gesto L. Vľavo ukázané gesto. Vpravo trojica obrázkov - vpravo hore detekovaný blob, vľavo vertikálny histogram intenzít, dole horizontálny histogram intenzít

Analýza geometrie blobu poskytovala väčšie množstvo rozpoznateľných gest, ale získané údaje boli veľmi nestabilné. Stred detekovanej dlane sa neustale pohyboval a pri zlyhaní detekcie sa objavil i mimo blob ruky. Obr. 7.6 ukazuje detekciu prstov prahovaním vzdialenosti bodov kontúry od stredu dlane. Čierna časť kontúry označuje dlaň detekovanú podľa pravidla: ruka smerom k dlani zmenšuje svoju šíriku, na začiatku dlane svju šírku náhle

zväčší (o 30 bodov). Červené čiary zobrazujú maximá, modré čiary minimá vzdialenosti bodov kontúr od stredu blobu.



Obrázek 7.6: Detekcia ruky a klasifikácia gesta na základe geometrie blobu

V aplikácií podľa požiadavkov je dôležitá stabilita vyhodnotených údajov, pričom popísané metódy túto podmienku nespĺňajú v dostatočnej miere. Tento spôsob klasifikácie je priamy a neumožňuje štatistické vyhodnocovanie pomocou K-nearest neighbours.

Metóda porovnávania kontúr pomocou Hu momentov sa ukázala ako skvelý kompromis medzi náročnosťou a výsledkom. Porovnávanie momentov s jedným vzorovým gestom fungovalo dobre iba s rukou, na ktorej boli vypočítané vzorové momenty. Tento fakt priamo podporuje potrebu použitia algoritmu K-Nearest Neighbours na štatistické vyhodnotenie typu gesta. Výpočet Hu momentov je časovo veľmi náročný, preto musel byť upravený spôsob ukladania trénovacích dát (ukladanie predvypočítaných momentov).

K-nearest neighbours klasifikácia v spojení s Hu momentami rapídne zvyšuje spoľahlivosť klasifikácie a odolnosť voči zmenám veľkosti, rotácie i posunutia. Pri malom počte trénovacích gest spoľahlivosť stúpa so svyšujúcim sa počtom trénovacích dát, pri vyššom počte trénovacích dát (rádovo desiatky pre každé gesto) sa spoľahlivosť ustáli. Strata výkonu je v navhrnutom riešeni takmer zanedbateľná (vďaka predvypočítaným momentom všetkých šablon).

7.2 Výber gest vhodných pre detekciu a určenie chybvosti klasifikátora

V požiadvkach na aplikáciu bolo 10 rozpoznávaných gest (viď obr. 7.7). Toto sa neskôr ukázalo ako nesplniteľné a nakoniec i náročné pre užívateľa.

Napriek dostatočnému množstvu trénovacích dát klasifikátor prejavoval silnú nestabilitu medzi podobnými gestami. Obrázok 7.7 predsatvuje confusion matrix za najlepších podmienok pre klasifikátor (tzn. umelé osvetelenie, blob bez defektov). Pri tomto teste nebola uvažovaná kontrola pravosti s potvrdením gesta po zaradení 5x do tej istej skupiny. Test ukazuje priame výsledky zaradenia algoritmu KNN. Každé gesto bolo hodnotené 300x v rôznej pozicií, v rôznych uhloch a v rôznych uhloch roztiahnutia prstov. V prvom riadku

tabulky sú uvedené testované gestá, v prvom stĺpci sú možnosti vyhodnotenia. Tabuľka obsahuje percentuálne zastúpenie správnych a nesprávnych rozhodnutí klasifikátora. Celková úspešnosť klasifikátora pre 10 gest bola 63.16%, čo stále nie je dostatočná presnosť.

Pre zvýšenie spoľahlivosti bolo z najnestabilnejších dvojíc odstránené jedno gesto. Tým sa počet gest znížil na 5, ale s vysokou úspešnosťou rozhodovania (95.3%) - viď obr. 7.8. Opäť boli gestá testované v najlepších podmienkach, bez kontroly pravosti s 5-násobným vyhodnotením, každé gesto 300x a s malými obmenami v uhloch medzi prstami.

Po zvyknutí hráča na reakcie klasifikátora a po naučení sa „lepšieho držania gesta“ sa nesprávne rozhodnutia klasifikátora obmedzia a jeho spoľahlivosť sa priblíži ešte viac k 100%.

Gestá, s ktorými pracovali knowledge-based klasifikátory, neboli pre túto aplikáciu vhodné ani stabilitou, ani počtom.

Testované gesto

Vyhodnotené gesto

	3.6%									1.0%
	2.0%	**81.0%**								13.0%
			62.3%	33.6%	17.0%			5.3%		
	0.3%		4.0%	**37.0%**	23.0%			3.3%	4.0%	
	2.6%		22.3%	26.3%	**58.3%**			28.3%	0.3%	
	7.6%		7.3%			**86.3%**			0.3%	29.0%
	2.6%					7.0%	**99.6%**			12.0%
	64.3%		1.6%	3.0%	1.6%			**63.0%**		
	2.0%		2.0%			6.6%	0.3%		**95.3%**	
	14.6%	19.0%								**45.0%**

Obrázek 7.7: Confusion matrix pre 10 gest. Celková úspešnosť vyhodnotenia 63.16%.

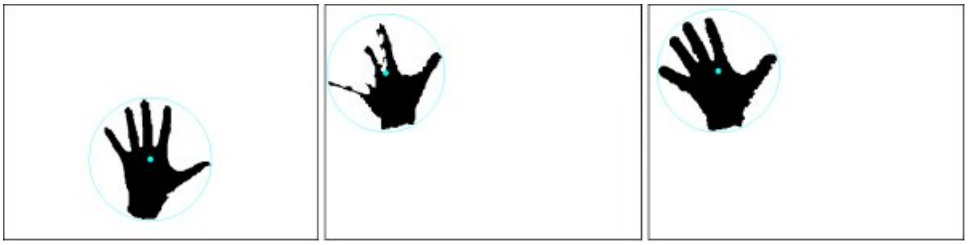
Vyhodnotené gesto \ Testované gesto					
	90.6%	2.6%			
	2.6%	**97.0%**			
	5.3%	0.3%	**100.0%**	0.3%	10.6%
	0.3%			**99.6%**	
	0.3%				**89.3%**

Obrázek 7.8: Confusion matrix pre redukovaný počet gest. Celková úspešnosť vyhodnotenia 95.3%.

7.3 Odolnosť voči rušivým vplyvom a nepriaznivým podmienkam

Zostava bola testovaná v najrôznejších svetelných podmienkach. Prebehli 2 skupiny testov - v prirodzenom osvetlení a v umelom osvetlení.

Keďže boli použité 2 lampy, v rohoch „hracej“ plochy, osvetlenie je veľmi silné a viditeľne zafarbené do žlta. V mnohých prípadoch sa farba blíži až k bielej. Po prvtonej kalibrácií farby ruky umiestnenej v strede hracej plochy, v horných oblastiach dochádza k nesprávnemu vyhodnoteniu bodov. Tak vznikajú rôzne „defekty“ a nerovnosti v blobe, prípadne zmizne celý prst. To ovplyvňuje výsledok klasifikácie gesta. Tento nedostatok bol eliminovaný viacnásobnou kalibráciou v rôznych miestach hracej plochy (eliminácia nerovnomerného osvetelenia) i uhloch (eliminácia tieňov). Tento problém znázorňuje obrázok 7.9, Podobný postup je možné aplikovať i za prirodzeného osvetlenia na potlačenie tieňov na ruke.



Obrázek 7.9: Účinok viacnásobnej kalibrácie. Zľava: ruka po 1. kalibrácií v strede plochy, po 1. kalibrácií v rohu - blob je zdeformovaný, po 2. kalibracií v rohu - blob je v poriadku.

Pri zatemnení miestnosti a s umelým osvetlením su prakticky nemožné náhodné zmeny osvetlenia spôsobené oblakmi či postupom slnka po oblohe, preto vyhodnocovanie v týchto

podmienkach bolo spolahlivé neobmedzene dlho.

Počas denného svetla je plocha konštantne osvetlená a nevznikajú problémy v horných rohoch hracej plochy. Namiesto toho vznikajú problémy s tieňmi samotnej ruky. Tiene na samotnej ruke sa dajú opäť kompenzovať viacnásobnou kalibracioou, ale tiene na hracej ploche majú veľmi nepriaznivý vplyv na detekciu ruky. Pri pozadí konštantnej farby (najlepšie výsledky pre modrú) sú tieto vplyvy významne potlačené. Pri rôznorodom pozadí tiene znemožňujú klasifikáciu.

Ďalším problémom sa ukázal dlhší beh aplikácie, kedy slnko významne zmenilo uhol svetla, či „zašlo za roh“ a hracia plocha sa z priameho slnečného žiarenia dostala do tieňa. Podobný problém vznikol pri prechode veľkého mraku. Pri stálych poveternostných podmienkach je správanie stabilné. Najhoršie výsledky vykazuje systém v ranných a večerných hodinách - pri západe a východe slnka.

Systém detekcie ruky ďalej vykazuje lepšie výsledky, ak sa zvýši saturácia obrazu a vhodne upraví kontrast a jas (obr 7.10).

Obrázek 7.10: Úprava kontrastu a jasu obrazu pre zlepšenie spoľahlivosti klasifikátora. Vľavo obraz s defaultnými nastaveniami, vpravo upravený obraz, ktorý je lepšie analyzovateľný navrhnutým systémom.

7.4 Užívateľský test rozhrania

Ovládanie navrhnutej hry funguje nasledovne: V strede obrazu je tzv. „mŕtva oblasť“. Ak sa stred ruky nachádza v tejto oblasti, nič sa nedeje. Opustením tejto oblasti sa so vzdialenosťou od nej zvyšuje sila reakcie - tzn. do strán rýchlosť otáčania, dopredu rýchlosť pohybu (pre viac informácií viď priložené DVD prípadne Návod na hru).

Testovania hry sa zúčastnilo niekoľko osôb. Z objektívneho hladiska je ovládanie gestami ruky v tomto projekte neporovnateľné s ovládaním pomocou klávesnice. Najväčším problémom bola rýchlosť reakcií samotného človeka - na zmenu smeru otáčania pri klávesnici stačí spraviť malý pohyb (stlačenie klávesy - pre prst cca. 0.5cm), pri ovládaní gestami je nutné prejsť rukou na druhú stranu hracieho poľa, čo predstavuje cca. 30-40 cm.

Ovládanie gestami ruky bolo jednoznačne označené za zaujímavé, i keď pre daný typ hry značne namáhavé. Zostrelenie všetkých robotov pomocou klávesnice sa pohybuje v rozmedzí 1-2 min, pomocou gest ruky 6 - 12 min. Tieto časy je možné zlepšiť trénovaním osôb, alebo iným princípom ovládania hry.

Kapitola 8

Záver

Cieľom tejto práce bol návrh a implementácia hry ovládanej gestami ruky. Vlastný návrh riešenia predchádzalo štúdium, testovanie a vyhodnocovanie vhodnosti rôznych metód spracovania obrazu a rozpoznávania gesta.

Pri detekcí ruky v obraze sa algoritmy odčítavania pozadia prejavili ako nevhodné, na rozdiel od algoritmov modelovania popredia - farby kože (Bayessov teorém, modelovanie gaussovým rozložením). Klasifikácia gesta ruky najspoľahlivejšie fungovala pomocou porovnávania Hu invariantných momentov a následným štatistckým zaradením do skupiny (algoritmus K-nearest neighbours). Knowledge-based algoritmy založené na analýze histogramov intenzít alebo geometrie detekovaného objektu záujmu sú vhodné len na veľmi obmedzené množstvo výrazných gest a teda nevhodné pre túto aplikíciu.

Pôvodné požiadavky na univerzálny klasifikátor s veľkým počtom rozpoznávaných gest (10) sa ukázali ako nesplniteľné. Preto bol tento počet zredukovaný na polovicu, čím sa úspešnosť priamej klasifikácie (bez ďalších algoritmov eliminujúcich náhodné a krátkodobé falošné vyhodnotenia) priblížila 96%.

Navrhnutá a implementovaná demonštračná aplikácia - 3D FPS (first person shooter) hra pokrýva využitie všetkých gest a jednej dvoj-kombinácie. Napriek nekomplexnému riešeniu hry, implementovaná časť dostatočne demonštruje použitie všetkých gest a možnosti ovládania PC.

Celý systém pracuje za väčšiny bežných svetelných podmienok spoľahlivo. Najlepšie výsledky vykazuje pri jednofarebnom (modrom) pozadí, i keď nie je nutné. Nevýhodou je nutnosť zásahu užívateľa - nutnosť vytvorenia modelu farby kože - teda kalibrácia klasifikátora pri každom spustení.

V budúcnosti sú vytýčené dva hlavné ciele: odstrániť potrebu kalibrácie, ktorá zaťažuje užívateľa, a zvoliť vhodnejší algoritmus rozpoznávania ukazovaného gesta, aby bolo možné detekovať napríklad gestá znakovej reči. Doplnením sledovania trajektórie ruky (resp. rúk) sa z tohto projektu môže stať významná súčasť štúdií o analýze znakovej reči pre postihnutých.

Literatura

[1] Tablet [online]. http://sk.wikipedia.org/wiki/Tablet, 2008-11-17 [cit. 2009-04-21].

[2] Počítačová klávesnica [online]. http://sk.wikipedia.org/wiki/Počítačová_klávesnica, 2009-01-17 [cit. 2009-04-21].

[3] Mouse (computing) [online]. http://en.wikipedia.org/wiki/Mouse_(computing), 2009-04-13 [cit. 2009-04-21].

[4] Trackball [online]. http://en.wikipedia.org/wiki/Trackball, 2009-04-15 [cit. 2009-04-21].

[5] Color model [online]. http://en.wikipedia.org/wiki/Color_model, 2009-04-16 [cit. 2009-04-22].

[6] RGB color model [online]. http://en.wikipedia.org/wiki/RGB_color_model, 2009-04-16 [cit. 2009-04-22].

[7] Touchpad [online]. http://en.wikipedia.org/wiki/Touchpad, 2009-04-21 [cit. 2009-04-21].

[8] Normal distribution [online]. http://en.wikipedia.org/wiki/Normal_distribution, 2009-04-24 [cit. 2009-04-26].

[9] Sony EyeToy [online]. http://www.eyetoy.com/index.asp, [cit. 2009-04-22].

[10] Blain, L.: Honda's Brain-Machine Interface: controlling robots by thoughts alone [online]. http://www.gizmag.com/honda-asimo-brain-machine-interface-mind-control/11379/, 2009-03-22 [cit. 2009-04-21].

[11] Bradski, G.; Kaehler, A.: *Learning OpenCV*. O'REILLY, 2008, iSBN 978-0-596-51613-0.

[12] Kardi, T.: K-Nearest Neighbors Tutorial [online]. http://people.revoledu.com/kardi/tutorial/KNN/, 2009-04-06 [cit. 2009-04-27].

[13] Kirillov, A.: Hands Gesture Recognition [online]. http://www.codeproject.com/KB/audio-video/hands_gesture_recognition.aspx, 2008-08-22 [cit. 2009-04-26].

[14] Piccardi, M.: Background subtraction techniques: a review [online]. http://www-staff.it.uts.edu.au/ massimo/BackgroundSubtractionReview -Piccardi.pdf, 2004-04-15 [cit. 2009-04-26], iSBN 0-9660176-3-3.

[15] Soille, P.: *Morphological Image Analysis - Principles and Applications*. Springer, druhé vydání, 2003, iSBN 3-540-42988-3.

[16] Steven W. Smith, P.: The Scientist and Engineer's Guide to Digital Signal Processing [online]. http://www.dspguide.com/, 1997 [cit. 2009-04-24], iSBN 0-9660176-3-3.

[17] Teng, K.; Ng, J.; Lim, S.: Hand Gesture Recognition [online]. http://www.kenteng.com/httpdocs/projects/hand_gesture.html, 2007-01-26 [cit. 2009-04-26].

[18] Theodoridis, S.; Koutroumbas, K.: *Pattern Recognition*. Academic Press, druhé vydání, 1999, iSBN 0-12-686140-4.

[19] Vezhnevets, V.; Sazonov, V.; Andreeva, A.: A Survey on Pixel-Based Skin Color Detection Techniques [online]. http://graphics.cs.msu.ru/en/publications/text/gc2003vsa.pdf, 2003-09-17 [cit. 2009-04-23].

[20] Weisstein, E.: Gaussian Function [online]. http://mathworld.wolfram.com/GaussianFunction.html, 2009-01-22 [cit. 2009-04-26].

[21] Černocký, J.: Štatistické rozpoznávání vzorů (2. prednáška kurzu Systémy zpracování řeči) [online]. http://www.fit.vutbr.cz/study/courses/SRE/public/.cs, 2005 [cit. 2009-04-24].

Dodatek A

Obsah priloženého DVD

- `doc/pdf` - tento dokument vo formáte pdf
- `doc/source` - zdrojový kód (LaTeX) tohto dokumentu
- `poster` - prezentačný plagát k hre
- `project/bin` - preložený spustiteľný projekt (+ knižnice OpenCV a Irrlicht Engine)
- `project/source` - zdrojový kód projektu (C++, MS Visual Studio 2009 projekt)
- `video` - ukážkové videá

Dodatek B

Návod na hru

V strede obrazu je tzv. „mŕtva zóna“. Ak je stred ruky (červený kruh) v tejto oblasti, nič sa nedeje. Ak stred ruky opustí túto oblasť, jeho vzdialenosť od nej určuje silu efektu - doľava rýchlosť otáčania doľava, doprava rýchlosť otáčania doprava, hore rýchlsť pohybu vpred.

Cieľom hry je zostrelenie všetkých (5) robotov. Uprostred hracej plochy je výťah pozostávajúci z 2 časticových systémov. Výška výťahu sa dá zvýšovať iba v oblasti modrého systému. Po vstupe do oranžového systému začne hráč stúpať do danej výšky. Otáčaním sa môže rozhliadnuť po okolí krajiny a zistiť, kde sa nachádzajú ďalšie roboty. Dotyk hráča s robotom znamená prehru.

Hráč má na 1 nabitie 10 nábojov. Po zminutí nábojov musí zbraň opäť nabiť.

Pohyb a otáčanie

Streľba

Nabiť zbraň

a následne

Zvýšiť výšku výťahu