

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

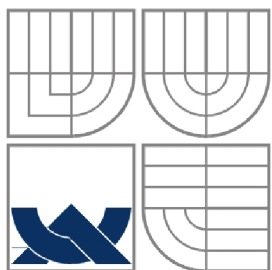
OVLÁDÁNÍ POČÍTAČE GESTEM RUKY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

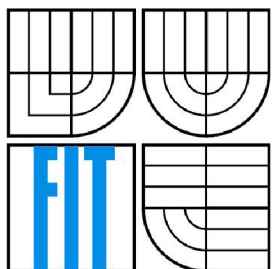
AUTOR PRÁCE
AUTHOR

MILOŠ MATĚJÍČKA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

OVLÁDÁNÍ POČÍTAČE GESTEM RUKY

HAND COMPUTER CONTROLLING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MILOŠ MATĚJÍČKA

VEDOUCÍ PRÁCE
SUPERVISOR

ING VÍTĚZSLAV BERAN

BRNO 2008

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Matějčka Miloš**

Obor: Informační technologie

Téma: **Ovládání počítače gestem ruky**

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte dostupnou literaturu týkající se tématu. Proveďte rozbor dosavadního stavu problematiky.
2. Na základě nastudovaných informací navrhnete postup řešení zadání. Diskutujte zvolené metody a vysvětlíte navržené řešení.
3. Vytvořte anotovanou testovací sadu dat pro závěrečné testování Vašeho řešení.
4. Navrhnete a implementujete knihovnu realizující Vámi navržený postup. Dbejte na programovou čistotu, modulárnost a přenositelnost knihovny. Knihovnu dokumentujte.
5. Vyhodnoťte Vaše řešení pomocí testovacích dat, srovnajte Vaše výsledky s výsledky jiných prací a proveďte jejich diskuzi.

Literatura:

- Žára, J., Beneš, B., Felker, P.: Moderní počítačová grafika, Computer Press, 1998.
- Hlaváč, V.: Zpracování signálů a obrazů, Praha, ČVUT, 2005.
- Rafael C. Gonzalez, Richard E. Woods: Digital image processing, Prentice-Hall, 2002.
- Jan, Jiří.: Číslíková filtrace, analýza a restaurace signálů, Brno, VUTIU, 2002.
- Dále dle pokynu vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1., 2. a 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

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

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

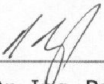
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo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

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

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
L.S.
602 00 Brno, Božetěchova 2


doc. Dr. Ing. Pavel Zemčík
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Miloš Matějčka**
Id studenta: 79318
Bytem: Chotěboř - Příjemky 35, 583 01 Chotěboř
Narozen: 20. 03. 1986, Havlíčkův Brod
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Ovládání počítače gestem ruky
Vedoucí/školitel VŠKP: Beran Vítězslav, Ing.
Ústav: Ústav počítačové grafiky a multimédií
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

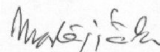
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel



.....

Autor

Abstrakt

Bakalářská práce se zabývá ovládáním počítače pomocí jednoduchých gest ruky. Je zde popsán princip rozpoznávání ruky pomocí barvy pokožky, zpracování a klasifikace gest rukou. Součástí práce je implementace aplikace pro rozpoznání gest ve videu. Pro práci s videem byla použita knihovna OpenCv.

Klíčová slova

gesta ruky, ovládání počítače, OpenCv, rozpoznávání pokožky, rozpoznávání trajektorie, zpracování obrazu, rozpoznávání gest

Abstract

Bachelor's thesis is engaged in computer control with simple one hand gestures. There is described principal of hand detection by skin color recognition, processing and classification of hand gestures. The thesis includes implemented application for hand gesture recognition from video. For work with video library OpenCv is used.

Keywords

hand gesture, computer control, OpenCv, skin color recognition, track recognition, image processing, gesture recognition

Citace

Miloš Matějčka: Ovládání počítače gestem ruky, bakalářská práce, Brno, FIT VUT v Brně, 2008

Ovládání počítače gestem ruky

Prohlášení

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

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

.....
Jméno Příjmení
Datum

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Vítězslavu Beranovi za jeho trpělivost a pomoc při tvorbě této práce.

© Miloš Matějčíka, 2008.

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

Obsah

1 Úvod.....	2
2 Teoretický základ.....	4
2.1 Detekce částí těla.....	4
2.2 Analýza oblastí.....	6
2.3 Rozpoznávání gest.....	7
3 Ovládání počítače gestem ruky.....	9
3.1 Struktura systému.....	9
3.2 Detekce rukou.....	11
3.3 Analýza oblastí ruky.....	15
3.4 Rozpoznávání gest ruky.....	16
4 Vyhodnocení systému.....	20
4.1 Způsoby hodnocení.....	20
4.2 Výsledky testů.....	20
5 Závěr.....	23
Literatura.....	24
Seznam příloh.....	25
Přílohy.....	26

1 Úvod

Při stále se zvyšujících výkonech dnešních počítačů se nabízejí nové možnosti pro jejich využití. Jedním z nich je zpracování videa v reálném čase, které má své využití nejen v osobních počítačích, ale také ve videokamerech, digitálních fotoaparátech, či mobilních telefonech. V současné době je velmi rozvíjejícím se oborem automatizované rozpoznávání a detekce. Lidé se snaží, aby co nejvíce jejich práce mohli vykonávat počítače i v této oblasti. Rozpoznávání obrazu se využívá například v dopravě, lékařství, pro vojenské účely, při zkoumání vesmíru, ale i v zábavním průmyslu. Možnost ovládat počítač jinak než klasickými prostředky je více než lákavá. Vyvíjejí se tak systémy pro ovládání počítače gesty, ale také například hlasem. V minulosti bylo využití rozpoznávání v praxi značně limitováno výkonem počítačů. Dnes již máme k dispozici dostatečně výkonné a zároveň cenově dostupné stroje, takže se automatické rozpoznávání stále více začíná využívat i v oblastech, kde by to bylo ještě před několika lety nemyslitelné. Nejen, že ve videu umíme dnes detekovat osoby, ale dokážeme rozpoznat i jejich jednotlivé části těla či výraz jejich obličeje. Přesná detekce je základem pro virtuální realitu.

Ve své bakalářské práci se věnuji rozpoznávání gest ruky, které může být využito pro jednoduché ovládání počítače, v počítačových hrách, nebo by šlo využít i ve vestavěných systémech. Pro ovládání počítače gesty ruky je nejdůležitější zvolit způsob, jakým získáme potřebné informace o osobě u které rozpoznáváme gesta. Tím asi nejpřirozenějším a zároveň nejdostupnějším způsobem je snímání osoby kamerou.

Cílem práce je vytvoření systému, který umožňuje detekci jednoduchých gest rukou ve videu. Systém bude umožňovat načítání a zpracování videa ve kterém bude detekovat gesta. Na tyto gesta bude systém reagovat. Dále bude systém umožňovat zpracování testovacích dat pro trénování a následnou detekci.

Gesta rukou je možné rozdělit do několika oblastí. Nejednodušší gesta jsou taková, kdy se sleduje pohyb jedné ruky. Jiný typ gest mohou sledovat obě ruce, jejich vzájemnou polohu, případně jejich pozici vzhledem k hlavě. Dalšími gesty jsou pohyby prstů. V tomto případě se již nesleduje pohyb ruky, ale její tvar. U složitějších gest je možné sledovat kombinaci sledování polohy ruky a jejího tvaru. Pro tuto práci jsem vybral gesta pohybu jedné ruky. Rozpoznávána je tedy trajektorie pohybu jedné ruky.

Rozpoznávání gest lze rozdělit do několika etap. V první etapě se ve videu detekují oblasti, které jsou pro analýzu gest podstatné, tedy oblasti, v nichž se nachází ruka sledované osoby. V další etapě je třeba detekované oblasti analyzovat. Do analýzy lze zahrnout rozpoznání výrazných bodů na ruce, její polohy, úhlu natočení, nebo třeba tvar. Dále dochází k hledání odpovídajících si

oblastí v čase a to tak, abychom získali souvislý soubor informací, který použijeme pro rozpoznání konkrétních gest. Pro každou oblast je tak generována množina dat, která uchovává výsledky analýzy v čase. V poslední fázi dochází k detekci samotných gest. Toto rozdělení umožňuje samostatnou práci na jednotlivých dílčích částech, které pak tvoří výsledný systém.

Text bakalářské práce je rozdělen na několik částí. Po úvodu následuje kapitola, která se věnuje obecným přístupům použitelným pro detekci rukou v obraze, analýze těchto oblastí a rozpoznávání samotných gest. Zejména je zde kladen důraz na možné problémy při rozpoznávání gest a jejich řešení.

Třetí kapitola se věnuje návrhu a vlastní realizaci systému pro detekci gest. Je zde popsána struktura vytvořené knihovny, která je jádrem celého systému, dále jsou zde podrobněji přiblíženy její základní komponenty.

Čtvrtá kapitola se věnuje testování systému. V této části jsou popsány způsoby jakými byl celý systém testován. Výsledky těchto testů jsou znázorněny v přehledných tabulkách a jsou doplněny i komentáři vysvětlující možné příčiny některých výsledků.

Pátá závěrečná kapitola shrnuje dosažené výsledky a nastiňuje možnosti dalšího rozvoje a vylepšení knihovny.

2 Teoretický základ

Návrh každé aplikace vychází z teoretického základu. V této kapitole se věnuji základním součástem systému pro detekci gest. Každá z těchto součástí nabízí mnoho způsobů pro řešení. Ty nejzákladnější jsou v této části popsány. Dále se zde věnuji možným problémům a způsobům jejich řešení.

2.1 Detekce částí těla

Analýza vstupního videa je velmi časově náročná. Proto je potřeba analýzu nějak zjednodušit. Nejjednodušším způsobem je omezení oblastí se kterými se pracuje. Pro zpracování gest lze tuto oblast omezit na část obrazu ve které se nachází ruka.

2.1.1 Detekce oblastí pokožky

Detekce ruky v obraze je první fází pro rozpoznávání gest. Nejpoužívanější metodou pro detekci rukou je rozpoznávání pomocí barvy pokožky [1]. Lidská kůže nabývá mnoha odstínů což přináší nemalé problémy. Dalším problémem je kvalita nasnímaného signálu a jeho zkreslení. Ke zkreslení může dojít buď nekvalitní kamerou, nebo barvou osvětlení scény. Pokud je scéna osvětlena jiným než bílým světlem, pokožka má jiné odstíny. Hlavní výhodou rozpoznávání rukou pomocí barvy pokožky je její jednoduchost a rychlost, která je pro detekci gest v reálném čase nesmírně důležitá.

Jednodušším způsobem detekce pokožky je určení barev, které považujeme za pokožku. Tyto barvy můžeme určit jako výčet jednotlivých barev, nebo pomocí podmínek na jednotlivé složky barvy. Tato metoda je jednoduchá, protože přímo dostaneme pixely které jsou pokožkou a které nikoliv. Výstupem je tedy dvoubarevná bitmapa, kterou můžeme dále analyzovat. Problém je, jak nalézt optimální výčet barev pokožky.

Přesnějším způsobem detekce pokožky v obraze je pravděpodobnostní ohodnocení jednotlivých pixelů. Každému pixelu je přiřazena určitá hodnota, která znázorňuje pravděpodobnost s jakou máme daný pixel pokládat za pixel pokožky. Pro ohodnocení pixelů lze použít dvě možnosti. Tou první je nalezení nějakého matematického vzorce, který pro každý pixel dokáže vygenerovat ohodnocení z jeho barevných složek. Druhou možností je vytvoření histogramu, ze kterého lze danou pravděpodobnost odečíst. Výstupem je v tomto případě bitmapa ve stupních šedi, kde jas každého pixelu vyjadřuje pravděpodobnost výskytu pokožky. Pro rozpoznání pokožky lze pak použít některý z algoritmů pro rozpoznání shluků jako například Mean shift algoritmus, který je blíže popsán v kapitole 2.1.3.

2.1.2 Barevné modely

Pro detekci rukou pomocí barvy pokožky můžeme využívat různé barevné modely [2]. Existuje celá řada barevných modelů, které lze pro detekci pokožky použít. V dalším textu se zaměřím na dva základní modely.

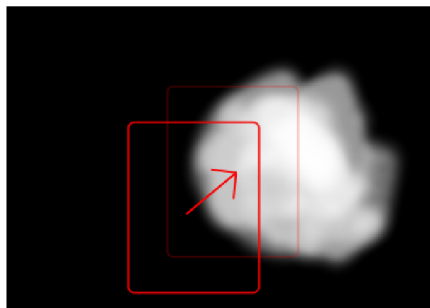
Nejpoužívanější barevný model RGB, který uchovává jasy jednotlivých barevných složek (červené, zelené a modré) se nabízí jako první. Jeho výhodou je, že je to model, který je běžný pro uchování barev v počítači. Používá se jak v kamerách pro snímání obrazu, tak i v monitorech pro zobrazení videa. Ovšem pro detekci není úplně výhodný. Největším problémem je intenzita osvětlení scény, která se projevuje ve všech třech barevných složkách.

Další možností je barevný model HSV (hue – odstín, saturation – sytost a value – jas). Změna jasu se projevuje jen v jedné barevné složce. Pro určení barev, které můžeme považovat za pokožku tak ze tří barevných kanálů stačí pracovat pouze se dvěma.

2.1.3 Mean shift algoritmus

Mean shift algoritmus je algoritmus pro iterativní hledání lokálních maxim. Tento algoritmus je možné použít pro detekci rukou z šedotónové bitmapy, kterou získáme pravděpodobnostním ohodnocením jednotlivých pixelů. Algoritmus vychází z toho, že směrem ke středu shluku se intenzita jasu zvyšuje [3].

Princip algoritmu Mean shift je ten, že vychází ze zadané oblasti v bitmapě a každým svým krokem tuto polohu zpřesňuje. Pro zadanou oblast spočítá polohu váženého průměru oblasti (těžiště). Polohu oblasti posune tak, že nalezené těžiště bude středem nové oblasti (obrázek 1). Tento postup se opakuje dokud vzdálenost středu oblasti od těžiště je větší než předem zvolená odchylka, případně dokud se nevyčerpá předem zvolený počet iterací. Pro výpočet polohy těžiště se využívá obrazových momentů, které jsou stručně popsány v části 2.2.1.



obrázek 1: Ilustrace hledání nové polohy oblasti - Mean Shift

Problém je nalezení počátečních oblastí pro hledání shluků. Jedním z přístupů je rozdělení vstupního obrazu na několik menších částí. Na každou část pak aplikujeme Mean shift algoritmus. Výsledné oblasti, které se překrývají můžeme považovat za jednu oblast, tedy za jeden nalezený shluk. Tento přístup má nevýhodu ve své rychlosti, protože se často musí prohledávat místa, kde žádný shluk není.

Jiným přístupem je přímé označení oblastí, kde shluk začneme hledat. To pro automatickou detekci gest není optimální přístup. Můžeme ale využít nalezené shluky z předchozího snímku vstupního videa. Je vysoce pravděpodobné, že se pozice shluků změní jen částečně a tak bude detekce mnohem rychlejší. Problém nastává v situaci, kdy se shluky ztrácejí z obrazu a následně znovu objevují. K tomuto jevu může dojít například překryvem ruky s jinou částí těla, pohybem ruky mimo snímanou oblast, nebo třeba prudkým pohybem, kdy může dojít k silnému rozmazání ruky což vede k deformaci barevné informace.

2.2 Analýza oblastí

Obrazová informace není pro rozpoznávání gest nejvhodnější. Po detekci oblastí je nutné jednotlivé oblasti analyzovat. Ze složité obrazové informace tak dostaneme sadu hodnot, které můžeme zaznamenávat a později dále zpracovávat.

2.2.1 Obrazové momenty

Obrazové momenty jsou prostředkem pro matematický popis obrazu. Vstupem pro výpočet obrazových momentů jsou intenzity jednotlivých bodů obrazu. Pomocí obrazových momentů je možné získat informace o těžišti, středu, rozměrech celkové intenzitě a orientaci objektu [4].

2.2.2 Hledání kontextu mezi jednotlivými snímky videa

Nalezení spojení mezi jednotlivými snímky videa je pro detekci gest nezbytné. Základním způsobem je podobnost mezi oblastmi. Vychází z toho, že mezi dvěma snímky videa nedochází k příliš velké změně polohy a postavení ruky. Je možné využít například překryvu oblastí rukou. Při rychlejším pohybu a nižším počtu snímků videa za vteřinu může snadno dojít k tomu, že se ve dvou po sobě jdoucích snímcích oblasti ruky nepřekrývají. Pak je třeba hledat odpovídající oblast i jinde.

Dalším problémem při hledání kontextu je možnost překrytí rukou na jednom snímku. Ze dvou oblastí se tak stává oblast jediná. Obdobně je tomu při následném oddělení rukou. Dalším problémem je situace, kdy ruka přestane být vidět. K tomu může dojít tím, že se ruka dostane mimo obraz, nebo do zákrytu s jiným předmětem ve scéně. Řešením těchto problémů vyžaduje

hlubší analýzu scény. To je zejména rozpoznání objektů scény tak, že dokážeme identifikovat který objekt je levá ruka, který pravá a které objekty vůbec ruce nejsou.

Další možností je částečné ignorování těchto faktorů. Zmizením objektu ze scény se sledování objektu ukončí a při objevení nového objektu se začne se sledováním od okamžiku objevení. Při tomto přístupu je problém s překrytím rukou. Oblast s jednou rukou zanikne a druhá má své pokračování v oblasti s oběma rukama. Může tak dojít k přerušení gesta.

2.3 Rozpoznávání gest

Po analýze oblastí jsou k dispozici vektory uchovávající informace o těchto oblastech. K těmto vektorům je třeba přiřadit odpovídající gesta. Pro jednoduchou detekci gest se v těchto vektorech nachází informace o trajektorii každé sledované oblasti. Každá trajektorie je trajektorií potenciálního gesta. Proto je nutné trajektorie analyzovat a zjistit, zda se jedná o nějaké gesto či nikoliv.

2.3.1 Problémy při rozpoznávání gest

Po úspěšném vytvoření vektoru s informacemi o jedné konkrétní oblasti je nutné se vypořádat s několika nepříznivými faktory. Prvním z nich je to, že gesta mohou být různě dlouhá, případně může být jejich rychlost proměnlivá. Pro správné rozpoznání gesta je nutné zvolit takový přístup, který si s tímto faktorem poradí.

Dalším problémem při sledování trajektorie gesta jsou odchylky mezi skutečnou a detekovanou trajektorií. Například pokud sledovaným bodem je střed ruky, tak může dojít k tomu, že bude detekována jen část ruky a tím se střed oblasti posune. K těmto posunům dochází neustále. Pokud je odchylka dostatečně velká, ovlivní tak trajektorii. Jednou z možností, jak se s tímto problémem vyrovnat je vyhlazení trajektorie. Jednotlivé polohy bodů trajektorie mohou být nahrazeny průměrem poloh okolních bodů. Jinou možností je nahrazení trajektorie aproximační křivkou, která zajistí vyhlazení trajektorie.

2.3.2 Tvorba referenčního modelu

Zřejmě nejrozšířenější způsob detekce gest je vytvoření modelu pro zvolený soubor gest. Trajektorie jsou pak předkládány jednotlivým modelům. Výsledkem je tak pravděpodobnost s jakou se načtená trajektorie shoduje s daným modelem. Pro tvorbu modelu existují dva základní přístupy. Tím prvním je přímé vytvoření modelu bez zpracovávání testovacích dat. To je výhodné zejména pro velmi jednoduchá gesta a jednoduché modely. Druhým přístupem je trénování modelu z předem vybraných testovacích dat. V takovém případě je možné tvořit jak jednoduché, tak i

složité a sofistikované modely. Úspěšnost trénování modelu závisí zejména na kvalitě vstupních dat a také na jejich počtu. Je nutné, aby trénovací data co nejvíce odpovídala přirozenému provedení gest.

2.3.3 Určování gest

Pro každou načtenou trajektorii je nutné zjistit pravděpodobnost s jakou odpovídá jednotlivým načteným modelům. Je tak vybrán nejpravděpodobnější model. Dále je nutné získanou pravděpodobnost porovnat s minimální požadovanou pravděpodobností. Pokud je získaná pravděpodobnost vyšší než minimální požadovaná, je možné gesto označit za rozpoznané. Po rozpoznání gesta je třeba zajistit, aby po přidání dalšího bodu trajektorie nebyla znovu trajektorie celá zpracovávána. Docházelo by tak k několikanásobnému rozpoznání jednoho gesta.

Pro přiřazení pravděpodobnosti shody trajektorie a gesta se nejčastěji používají konečné automaty. Příkladem poměrně úspěšného konečného automatu pro detekci gest jsou Skryté Markovovy modely (Hidden Markov models) [5], které se používají i pro mnohem náročnější rozpoznávání, jako je například analýza mluveného slova, nebo psaného textu. Markovovy modely pracují s množinou stavů a s pravděpodobnostmi jejich přechodů. Hlavní předností je to, že je pravděpodobnost přechodu do dalšího stavu závisí pouze na předchozích stavech. Největší nevýhodou Skrytých Markovových modelů je především jejich složitost a nutnost velkého množství referenčních dat pro natrénování těchto modelů.

Pro rozpoznávání jednoduchých gest je možné načtené trajektorie převést na sadu vektorů. Je možné zvolit mezi čtyřmi, osmi, nebo i více základními směry. Po této konverzi se již nepracuje s velkým množstvím bodů, ale jen s několikanásobně menším počtem vektorů. Výhodou je, že se eliminuje proměnlivá rychlost gest. Další výhodou je, že se zmenší vliv nechtěných odchylek. Tato konverze má ovšem i své nevýhody. Tou největší bezesporu je ztráta informace, která se projeví zejména při volbě čtyř směrů vektorů. Vektory tedy mají nižší vypovídající hodnotu, než původní trajektorie.

3 Ovládání počítače gestem ruky

Tato kapitola se věnuje vlastnímu návrhu a tvorbě aplikace pro ovládání počítače pomocí gest ruky. Návrh systému navazuje na předchozí teoretický základ.

Pro implementaci byla použita knihovna OpenCv, která byla vyvinuta společností Intel pro zpracování obrazu. [6]. Tato knihovna má v sobě implementováno mnoho užitečných algoritmů pro práci s videem. Není tak nutné zacházet při práci s videem do zbytečných detailů, ale je možné používat předem připravené a odladěné funkce knihovny.

Pro vlastní implementaci byl vybrán operační systém Windows a jako vývojové prostředí Visual Studio 2005. Pro toto prostředí jsou rovněž připraveny i projektové konfigurační soubory pro překlad a sestavení programů.

3.1 Struktura systému

Byla vytvořena knihovna pro detekci gest. Knihovna obsahuje moduly pro načítání a zpracování videa, detekci pokožky, detekci oblastí rukou a z nich následné vytvoření trajektorií pohybu, dále obsahuje modul pro rozpoznání gest. Knihovna také disponuje možnostmi pro generování modelů kůže i modelů jednotlivých gest. Pro předvedení funkčnosti knihovny byly vytvořeny celkem čtyři demonstrační aplikace.

Nejdůležitější demonstrační aplikace implementuje rozpoznávání gest ze vstupního video signálu. Jejím vstupem je tedy video, dále model kůže a množina předem natrénovaných modelů gest. Výstupem je pak výpis detekovaných gest. Aplikace má tři hlavní části. Tou první je zpracování vstupního obrazu a detekce oblastí, ve kterých se vyskytují ruce. Druhou částí je analýza, kde se hledá spojitost mezi jednotlivými oblastmi a snímky. Ve třetí části dochází k detekci samotných gest. K dalším funkcí tohoto programu patří možnost ukládání aktuálně načtených snímků videa a ukládání výřezů detekovaných trajektorií pro další použití. Ukázka aplikace je na obrázku 2 a její struktura na obrázku 3.

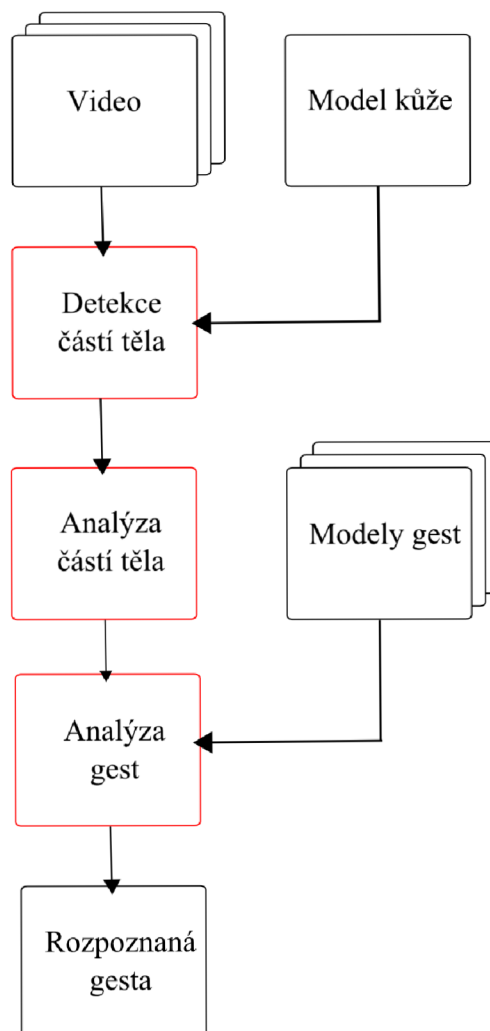
Další demonstrační aplikací je program pro tvorbu modelu pokožky. Jeho vstupem je bitmapa s vyřezanými částmi pokožky. Výstupem je histogram modelu kůže.

Třetí demo aplikací je program pro generování modelu gest. Jeho vstupem jsou trajektorie referenčních gest a rozlišení videa, ze kterých byly tyto trajektorie vytvořeny. Výstupem je pak model gesta, který může být použit pro rozpoznávání gest.

Poslední demonstrační aplikací je program pro klasifikaci zadané trajektorie. Výstupem programu je tak nejpravděpodobnější model gesta a míra shody zadaného gesta s modelem.



obrázek 2: Ukázka aplikace. Vlevo výstup detekce pokožky, vpravo video s vyznačenými oblastmi a trajektoriemi



obrázek 3: Struktura programu pro detekci gest

Detekce oblastí pracuje se snímky vstupního videa. Jejím výstupem jsou oblasti v pravděpodobnostní mapě vyznačující nejvýraznější části obrazu. Součástí této části je i vytvoření histogramu pro rozpoznání pokožky. Histogram se vytvoří pouze jednou a po celou dobu detekce se nemění. Histogram je možno generovat vždy při spuštění aplikace, nebo jednou a následně histogram uložit na pevný disk. Zvolené řešení je to, že je histogram generován mimo hlavní aplikaci a je uložen na disku. Z disku je načten při spuštění aplikace a dále se nemění. Princip detekce pokožky je popsán v kapitole 3.2.

Analýza pracuje s výstupem detekce oblastí. Výstupem analýzy je sada trajektorií obsahujících informace o jednotlivých oblastech v čase. Pro každou oblast je vygenerována jedna trajektorie. Při zániku oblasti je trajektorie zrušena, protože by pak docházelo k jejímu opakovanému zpracování. Aby nedocházelo k přetěžování počítače, je délka trajektorie omezena. To znamená že při překročení určité délky vektoru jsou nejstarší informace postupně odmazávány. Délka trajektorie je možná nastavit podle předpokládané nejdelší délky jednoho gesta. Například lze předpokládat, že jedno gesto nebude delší než 10 vteřin. Při rychlosti 15 snímků za vteřinu tedy stačí uchovávat posledních 150 záznamů. Na testovacích datech bylo zjištěno, že gesta nepřesahují délku sto snímků. Velikost trajektorií byla tedy omezena na sto položek. Analýza oblastí je popsána v kapitole 3.3.

Samotná detekce gest pracuje s trajektoriemi vygenerovanými analýzou. Pokud je gesto rozpoznáno, dojde k pokrácení trajektorie tak, aby nebylo gesto rozpoznáno znovu při dalším snímku. Rozpoznávání gest se věnuje kapitola 3.4.

3.2 Detekce rukou

Detekce rukou zahrnuje zpracování videa, detekce pokožky a lokalizace oblastí. Pro jednoduchost se zpracovávají všechny oblasti, které mají jistou pravděpodobnost, že jsou oblasti pokožky. Při detekci se tedy nerozlišuje zda je daná oblast ruka, hlava, nebo například kus nábytku s barvou podobnou pleti. Předpokládá se, že snímaná scéna je statická, takže i detekované části obrazu, které nejsou rukami budou statické. Protože mají detekovaná gesta nastavenou minimální velikost, tak budou tyto oblasti při detekci vynechány.

3.2.1 Záznam obrazu

Na kvalitě záznamu videa z velké části závisí celá úspěšnost detekce gest. Velmi důležitá je zejména věrnost nasnímaných barev. Zejména při umělém osvětlení může docházet k barevnému zkreslení. Detekce takového obrazu je pak velmi obtížná. Dalším problémem je hladina šumu.

Mezi další parametry nasnímaného videa patří jeho rozlišení. Pro jednoduchou detekci gest není potřeba nějak velké rozlišení. Pro testování bylo používáno video a rozlišením 320 na 240 pixelů. Při použití vyššího rozlišení nedochází již k výraznému zlepšení kvality rozpoznávání. Naopak vyšší rozlišení s sebou přináší vyšší nároky na zpracování a tím dochází ke zvýšení času potřebného pro zpracování jednoho snímku.

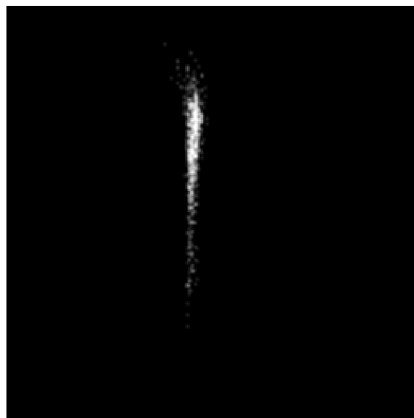
Dalším výrazným parametrem videa je počet snímků za vteřinu. Pro detekci gest je plně vyhovující rychlost patnácti snímků za vteřinu. Vyšší rychlost přináší větší nároky na rychlost zpracování, naopak při nižší rychlosti dochází k problémům s rychlejšími gesty.

Důležitým parametrem je členitost scény. Pokud je ve videu spousta objektů, které by svým odstínem mohli připomínat barvu lidské kůže, tak se můžou ztratit důležité informace. Dalším faktorem je rovněž vzdálenost snímané osoby od kamery. Pokud je osoba příliš daleko, tak jsou rozpoznávané oblasti kůže velmi malé a můžou být považovány za šum. V takovém případě se trajektorie nenačte vůbec, nebo je přerušovaná. V obou dvou případech to vede k nerozpoznání gesta.

3.2.2 Detekce pokožky

Pro detekci částí těla v obraze byla zvolena detekce pomocí pravděpodobnostního ohodnocení. Výstupem této detekce je pravděpodobnostní mapa vyznačující pravděpodobnost pokožky v jednotlivých bodech. Jako barevný model jsem vybral model HSV popsany v kapitole 2.1.2.

Převod mezi jednotlivými barevnými modely je v knihovně OpenCV poměrně dobře optimalizován a tak nepředstavuje výrazné prodloužení doby pro zpracování jednoho snímku. Protože jasová složka není pro detekci pokožky příliš důležitá, knihovna pracuje pouze s odstínem a sytostí. Pro jednoduché ohodnocování pixelů je předem vygenerovaný histogram. Tento histogram je vlastně bitmapa, která má rozměry 256 na 256 pixelů. Jeden rozměr bitmapy tedy znázorňuje odstín a druhý sytost. Příklad vygenerovaného histogramu je na obrázku 4.



obrázek 4: Příklad histogramu modelu kůže

Pro výpočet jednotlivých pravděpodobností je použita sada obrázků, na kterých jsou části pokožky (obrázek 5). Části obrázků, které nejsou pokožkou jsou začerněny a při zpracování se neuvažují. Tyto obrázky jsou procházeny bod po bodu tak, že se spočítá četnost všech kombinací odstínu a sytosti.



obrázek 5: Příklad výřezů pokožky

Dalším nepovinným vstupem jsou bitmapy, které pokožkou určitě nejsou (obrázek 6). Četnosti pixelů z těchto bitmap jsou odečteny. Výhodou tohoto nepovinného vstupu je možnost odfiltrvat konkrétní pozadí scény, které může být blízké barvě pokožky.



obrázek 6: Příklad obrázku s body, které nejsou pokožkou

Tyto četnosti jsou dále přepočítány tak, aby nejčtetnější barvy byly označeny v histogramu vyšším jasem a barvy, které nebyly vůbec nalezeny nejnižším. Nulová četnost je v histogramu znázorněna nulovou hodnotou a četnost, která je větší než průměrná četnost z nenulových hodnot je znázorněna maximální hodnotou. Četnosti, které jsou mezi těmito dvěma extrémy jsou rozděleny lineárně mezi nulovou a maximální hodnotou.

Tento způsob tvorby histogramu pro ohodnocení pokožky má několik nesporných výhod. Pokud použijeme rozsáhlou sadu vstupních obrázků, získáme tak histogram pro univerzální detekci pokožky. Je ale možné vytvořit histogram jen z omezeného množství obrázků, které korespondují s používanou kamerou a snímanou scénou. Výsledný histogram tak nepůjde použít univerzálně, ale pro konkrétní kameru a člověka bude mít detekce mnohem vyšší přesnost.

Další výhodou je to, že je tento způsob univerzální a lze použít nejen pro detekci pokožky, ale i pro detekci libovolné jiné barvy, či skupin barev.

3.2.3 Lokalizace rukou

Po vytvoření pravděpodobnostní mapy je třeba v ní detekovat jednotlivé oblasti. Pro tuto lokalizaci jsem vybral Mean shift algoritmus, který je popsán v kapitole 2.1.3.

Samotná lokalizace je implementována pomocí funkce CamShift. Tato funkce je součástí knihovny OpenCV a pro jednoduchou lokalizaci je plně dostačující. Funkce využívá Mean shift algoritmus a obrazové momenty.

Jako vstup vyžaduje funkce CamShift počáteční oblast v pravděpodobnostní mapě, kde začne hledat. To je její největší nevýhoda, protože nalezení počátečních oblastí je problém. Pro nalezení počátečních oblastí jsem opět zvolil funkci z knihovny OpenCv a to konkrétně funkci FindContours, která nalezne všechny oblasti v bitmapě. Omezením této funkce je nutnost stanovit hodnotu pixelu, která bude brána jako hraniční. Vstupní bitmapu jsem gaussovsky vyhladil a dále jsem zvolil dostatečně nízkou hraniční hodnotu, u které je jistota, že bude detekováno co nejvíce oblastí. Takto získané oblasti jsou přibližné a jsou tedy vhodné jako počáteční oblasti pro funkci CamShift. Dále jsem jako vstupní oblasti zvolil oblasti nalezené v předchozím snímku videa.

Funkce CamShift pracuje tak, že nejprve zvětší vstupní oblast tak, že z každé strany přidá určitý počet pixelů. Pro tuto zvětšenou oblast spočítá obrazové momenty. Tyto obrazové momenty funkce přepočítá na polohu a rozměry nové oblasti. Celý postup se opakuje dokud není zajištěna požadovaná přesnost (tj. starší a nová oblast jsou shodné, nebo velmi podobné), nebo není vyčerpán předem zvolený počet iterací. Výstupem funkce je oblast, která je umístěna a má rozměry, které odpovídají oblasti nalezené v mapě.

3.2.4 Filtrování oblastí

Při hledání oblastí je detekováno spousta shodných, nebo podobných. To je dáno vstupními oblastmi pro funkci CamShift. Je tedy nutné nalezené oblasti vyfiltrovat.

Nejjednodušším filtrem je odstranění duplicitních oblastí, tj. naprosto shodných, nebo velmi podobných oblastí. Dále je možné odfiltrovat oblasti, které neodpovídají svými parametry. To jsou zejména příliš úzké, nebo malé oblasti.

Poslední filtr se týká částečně se překrývajících oblastí. V tomto případě se odfiltruje oblast, která je méně výrazná, tj. oblast s výrazně nižší pravděpodobností výskytu pokožky. V případě že jsou pravděpodobnosti u obou porovnávaných oblastí podobné, dochází k jejich spojení v jednu oblast.

3.3 Analýza oblastí ruky

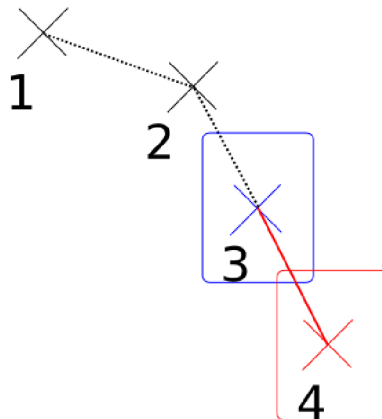
Analýza oblastí je implementována velice jednoduše. Pro další sledování se uvažuje vždy střed detekované oblasti. Dále je uchovávána informace o rozměrech oblasti, která se využívá pro hledání kontextu mezi oblastmi v čase.

3.3.1 Spojitost mezi jednotlivými oblastmi v čase

Protože je třeba každou oblast sledovat v čase, je třeba najít kontinuitu mezi jednotlivými snímky. Po sobě jdoucí oblasti jsou považovány za shodné, pokud se alespoň částečně překrývají. Pokud by takových oblastí bylo více, bere se jako odpovídající ta oblast, která je nejbližší.

Tento přístup má hlavní nevýhodu při protnutí rukou. Pokud se například ruce dostanou do zákrytu, tak ze dvou oblastí vzniká jedna. Následně při oddělení rukou vznikají opět dvě oblasti. Není možné určit která ruka po oddělení odpovídá ruce před zákrytem. Proto v takovémto případě nelze gesto s jistotou rozpoznat.

Analýza obsahuje i jednoduchý mechanismus opravy chyby ve videu. Pokud se sledovaná ruka ztratí na jeden jediný snímek a pak se objeví, tak se chybějící snímek nahradí předpokládanou polohou oblasti. Tato předpokládaná poloha je vypočítána z posledních dvou poloh a to tak, že se předpokládá stejný směr i rychlost pohybu. Předpokládaná poloha dalšího bodu je pozice předchozího bodu zvětšená o rozdíl posledních dvou bodů. Pro ilustraci odhadu je k dispozici obrázek 7.



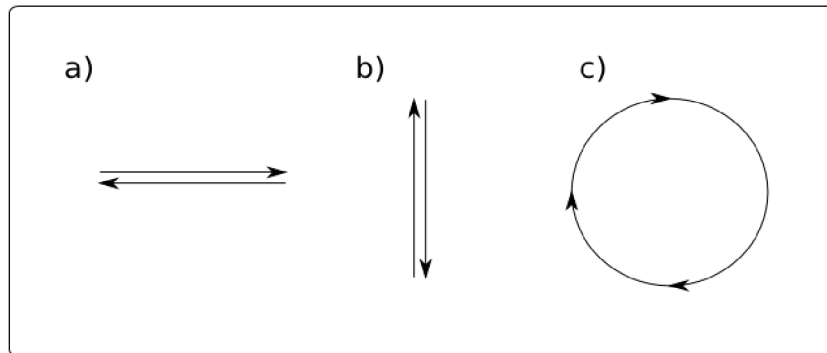
obrázek 7: Odhad dalšího bodu trajektorie

Předpokládaná poloha oblasti je použita i pro odstranění problémů s příliš rychlým pohybem. Pokud nebyla k oblasti nalezena vhodná následující oblast, tak se oblast hledání rozšíří i na předpokládanou polohu oblasti.

3.4 Rozpoznávání gest ruky

Pro rozpoznávání gest existuje spousta přístupů. Ne všechny jsou vhodné pro všechny typy gest. Gesta na kterých jsem testoval knihovnu jsou znázorněna na obrázku 8. Společnými rysy všech gest, která jsou pro systém akceptovatelná jsou následující:

- Gesto začíná i končí na stejném místě
- Je určena minimální velikost vykovávaného gesta a to relativně k rozlišení videa
- U gesta se nerozlišuje směr pohybu – analyzuje se tedy pouze vzniklá trajektorie. Například pokud je gestem pohyb rukou do kruhu, tak se nerozlišuje směr pohybu, ani to jestli bylo gesto započato na spodní, nebo vrchní straně kruhu
- Jedno gesto smí být vykonáváno jednou rukou, systém tedy neregistruje interakci mezi rukama navzájem



obrázek 8: Přehled testovaných gest

- a) *horizontála*
- b) *vertikála*
- c) *kruh*

Zvolená omezení pro gesta umožňují řadu zjednodušení. Tím základním je převod trajektorie na jednodušší seznam vektorů se kterými se dále pracuje. Převodu trajektorie na vektory se věnuje kapitola 3.4.1.

Dalším zjednodušením je normalizace těchto vektorů. Vektory jsou transformovány tak, že všechna gesta jsou převedena na taková, která by byla vykonána pohybem ve směru hodinových ručiček. Druhou změnou je že jsou posunuta tak, aby začátek těchto gest byl vždy vlevo dole a první pohyb byl veden vzhůru. Třetí změnou je změna velikosti gesta a to tak, aby gesta zabírala stejný prostor. Normalizaci vektorů se věnuje kapitola 3.4.2.

Normalizované vektory lze již poměrně snadno porovnávat s vytvořenými modely gest. Tvorbě modelů a klasifikaci gest je vyčleněna kapitola 3.4.3.

3.4.1 Převod trajektorie na sadu vektorů

Před samotnou detekcí gest dochází k převodu trajektorie na sadu vektorů. U každého vektoru je evidován jeho směr a velikost. Vybrané směry jsou nahoru, dolů, doleva a doprava. Trajektorie je postupně procházena a jsou postupně vypočítávány rozdíly mezi sousedními body. Tyto rozdíly jsou postupně sčítány dokud nepřesáhnou zvolenou hranici, která je odvozena od rozlišení videa.

Po přesáhnutí této hranice je zvolen jeden ze čtyř směrů a k němu odpovídající velikost. Pokud rozdíly mezi body označíme jako x (rozdíl na vodorovné ose) a y (rozdíl na svislé ose), tak jsou směry vektorů a jejich velikosti vybírány podle tabulky 1.

Podmínky		Směr vektoru	Délka vektoru
$ x > y $	$x > 0$	Vpravo	$ x $
$ x > y $	$x \leq 0$	Vlevo	$ x $
$ y \geq x $	$y > 0$	Nahoru	$ y $
$ y \geq x $	$y \leq 0$	Dolů	$ y $

Tabulka 1: Podmínky pro vektory

Vzniklé vektory jsou spojovány tak, aby dva po sobě jdoucí vektory nebyly stejného směru. Počet vzniklých vektorů se tak u drtivé většiny gest snížil na velmi malý počet.

3.4.2 Normalizace sady vektorů

Vektory jsou rovněž normalizovány. Normalizaci vektorů je popsána v následujících krocích:

1. Změna velikosti.

Je vypočítán minimální obdélník, do kterého by tyto vektory šli vepsat. Při výpočtu velikosti tohoto obdélníku se postupně procházejí všechny vektory a postupně je prováděn jejich součet. Program ukládá extrémy tohoto součtu a jejich rozdíl pak tvoří rozměry tohoto obdélníku. Dále je tento obdélník zvětšen tak, aby vyplnil prostor o velikosti rozlišení videa. Tímto je možné zaručit, že všechna shodná gesta budou přibližně stejně velká.

2. Spojení shodných krajních vektorů

Protože vycházíme z předpokladu, že gesta mají shodný začátek i konec, je možné s vektory dělat další operace, jako je rotace a otočení. Před těmito operacemi je ještě nutné zjistit, zda první i poslední vektor nejsou stejného směru. Pokud ano, tak poslední vektor odstraníme a velikost prvního zvětšíme o jeho velikost.

3. Posun vektorů

Pokud mezi vektory existuje vektor se směrem vzhůru, tak se největší z těchto vektorů zvolí jako první. Pokud vektor směrem vzhůru nenalezneme, řídíme se podle vektoru vpravo.

4. Otočení pořadí vektorů

Posledním krokem normalizace vektorů je převod vektorů tak, aby gesto bylo prováděno ve směru hodinových ručiček. Toho je dosaženo tak, že se postupně procházejí vektory a hledají se dostatečně velké vektory se směrem vlevo, nebo vpravo. Pokud první dostatečně velký vodorovný vektor je vpravo, tak se hledání ukončí, protože vektory jsou ve správném pořadí. Pokud první takový vektor vlevo, tak se provede změna pořadí vektorů.

Příklad normalizace vektorů na jednoduchém gestu pro video s rozlišením 320 na 240 bodů:

- | | | |
|----|--|-------------------------------|
| 0. | dolu 20; vpravo 30; nahoru 25; vlevo 25; dolu 10 | (gesto před normalizací) |
| 1. | dolu 160; vpravo 240; nahoru 200; vlevo 200; dolu 80 | (po změně velikosti) |
| 2. | dolu 240; vpravo 240; nahoru 200; vlevo 200; | (po spojení krajních vektorů) |
| 3. | nahoru 200; vlevo 200; dolu 240; vpravo 240; | (po posunutí vektorů) |
| 4. | nahoru 200; vpravo 240; dolu 240; vlevo 200; | (po otočení pořadí) |

3.4.3 Klasifikace gest

Pro klasifikaci gest je třeba nejprve natrénovat modely gest. Model gesta je sada vektorů, které jsou třeba před prvním použitím normalizovat podobně jako vektory, které se budou rozpoznávat.

Trénování modelu gest má podobný princip jako samotná klasifikace gest. Základem je funkce pro ohodnocení shody dvou sad vektorů. Před jejím použitím je třeba vektory upravit tak, aby odpovídající si směry byly na shodných pozicích. Tohoto je docíleno vkládáním prázdných vektorů. Je vygenerována sada dvojic, které jsou si podobné. U každé dvojice je určena míra shody. Při klasifikaci gesta je jako výstup brána nejvyšší míra shody. Při trénování modelu je aktuální model nahrazen součtem vektorů s nejvyšší shodou.

Dvojice sad vektorů jsou generovány rekurzivním způsobem. Předem jsou připravené prázdné sady vektorů pro výstup. V prvním kroku se najdou všechny shodné dvojice vektorů ve vstupních sadách. Při tomto hledání nezáleží na pozici vektoru. Pro tyto dvojice jsou vygenerovány výstupní sady vektorů a to z vektorů, které jsou před nimi. Algoritmus vždy upřednostňuje co největší počet shodných dvojic vektorů, tak aby se minimalizovala nutnost použití pomocných prázdných vektorů. Dále se do výstupních vektorů přidávají nalezené shodné vektory. Vektory, které jsou za nalezenými shodnými vektory se použijí jako vstup pro rekurzivní funkci. Generování

podobných sad končí v okamžiku, kdy již na vstupu nejsou žádné vektory. Pro celé toto generování dvojic je zapotřebí pouze informace o směru vektoru. S délkou vektoru se pracuje až při určení míry shody.

Při výpočtu míry shody dvou podobných vektorů nejprve dochází k pokusu o přerozdělení délky vektoru mezi sousední. K tomuto dochází pouze v případě, že jeden z dvojice vektorů je prázdný. V takovém případě se program pokusí doplnit sousední vektory stejného typu tak, aby byl rozdíl délek těchto vektorů co možná nejmenší.

Samotná míra shody sad vektorů v_1 a v_2 se vypočítá podle následujícího vztahu:

$$Shoda = 1 - \frac{\sum (|v_{1_i} - v_{2_i}|) * v_{max_i} / v_{min_i}}{\sum v_{1_i} + v_{2_i}}$$

kde v_{max_i} je větší z vektorů v_1 a v_2 s indexem i a v_{min_i} je menší.

Při klasifikaci vektorů není posuzována pouze kompletní sada vektorů. Testována je kompletní sada vektorů, která je postupně zkracována od nejstarších vektorů. Po každém pokrácení dochází k nové normalizaci vektorů. Nejmenší počet testovaných vektorů jsou dva. Dalším omezením pro testování je velikost gesta. Příliš malá gesta, která by bylo při normalizaci potřeba neúměrně zvětšovat se netestují. Zvolená hranice pro zvětšení gesta, která je ještě akceptovatelná je desetinásobné zvětšení. Z toho vyplývá, že nejmenší gesto, které je zpracováváno dosahuje alespoň v jedné ose velikosti jedné desetiny rozměru videa. Ze všech testů vybíráme nejpravděpodobnější gesto.

4 Vyhodnocení systému

Funkčnost každého systému se ukáže až na konkrétních datech. Při tvorbě systému byly testovány a optimalizovány všechny části programu. Skutečná použitelnost systému se ale ukáže teprve tehdy, pokud systém testujeme jako celek. Následující kapitola hovoří o způsobech testování a o konkrétních dosažených výsledcích.

4.1 Způsoby hodnocení

Pro testování aplikace bylo natočeno celkem 34 video sekvencí. Testování se zúčastnili tři osoby. Dvě osoby neznali strukturu, ani přesnou funkčnost aplikace, takže se řídili pouze stručným popisem jednotlivých gest. Pro natočení testovacích videí bylo použito celkem jedenáct různých prostředí.

Jak se při testování ukázalo, tak byly zvoleny příliš členité scény s mnoha rušivými elementy v záběru. Proto byl pro každé video vygenerován vlastní histogram pokožky. Při použití histogramu vygenerovaného na míru byla detekce oblastí poměrně uspokojivá. Matematicky tuto úspěšnost nelze vyjádřit, protože se jedná o subjektivní záležitost.

Pro konkrétní testy byly z nasnímaných videí uloženy fragmenty trajektorií:

- pro vertikální gesto 137 trajektorií
- pro horizontální gesto 139 trajektorií
- pro kruhové gesto 156 trajektorií

4.2 Výsledky testů

První test je nejobecnější. Referenční modely gest byly vygenerovány ze všech uložených trajektorií. Následně byly podle těchto modelů klasifikovány všechny trajektorie. Výsledky tohoto testu jsou v tabulce 2.

Gesta	Počet	Úspěšně rozpoznaná	Chybně rozpoznaná	Úspěšně rozpoznané se shodou vyšší než 0.85	Průměrná míra shody
Horizontální	139	138 (99.3%)	0	138 (99.3%)	0.97
Vertikální	137	137 (100%)	0	137 (100%)	0.98
Kruhové	156	142 (91.0%)	14 (9.0%)	86 (55.1%)	0.85
Souhrn	432	417 (96.5%)	14 (3.2%)	361 (83.5%)	0.93

Tabulka 2: Výsledky testování na všech trajektoriích

Pro druhý test byly modely vygenerovány jen z trajektorií gest vykonaných jednou osobou. Pro testování byla opět využita všechna gesta. Tabulka 3 shrnuje výsledky tohoto testu pro jednotlivé osoby a gesta.

Testovaná gesta vykonaná stejným člověkem jako gesta referenční					
Gesta	Počet	Úspěšně rozpoznaná	Chybně rozpoznaná	Úspěšně rozpoznané se shodou vyšší než 0.85	Průměrná míra shody
Horizontální	39	39 (100%)	0	39 (100%)	0.98
Vertikální	38	38 (100%)	0	38 (100%)	0.99
Kruhové	51	50 (98.0%)	1 (2%)	38 (74.5%)	0.87
Souhrn	128	127 (99.2%)	1 (0.8%)	115 (89.8%)	0.94
Testovaná gesta vykonaná jinými lidmi než gesta referenční					
Gesta	Počet	Úspěšně rozpoznaná	Chybně rozpoznaná	Úspěšně rozpoznané se shodou vyšší než 0.85	Průměrná míra shody
Horizontální	100	99 (99%)	0	99 (99%)	0.97
Vertikální	99	99 (100%)	0	99 (100%)	0.98
Kruhové	105	88 (83.8%)	17 (16.2%)	47 (44.8%)	0.83
Souhrn	304	286 (94.0%)	17 (5.6%)	245 (80.5%)	0.93

Tabulka 3: Výsledky testování pro natrénovaná gesta jedním člověkem

Pro třetí test bylo náhodně vybráno 40 gest pro natrénování každého modelu. Tato gesta byla vytvořena všemi třemi osobami, které se podíleli na testování. Gesta, která pro tvorbu modelů použita nebyla posloužila jako vstupní hodnoty testu. Výsledky shrnuje tabulka 4.

Gesta	Počet	Úspěšně rozpoznaná	Chybně rozpoznaná	Úspěšně rozpoznané se shodou vyšší než 0.85	Průměrná míra shody
Horizontální	99	99 (100%)	0	99	0.97
Vertikální	97	97 (100%)	0	97 (100%)	0.98
Kruhové	116	110 (94.8%)	6 (5.2%)	73 (62.3%)	0.86
Souhrn	312	306 (98.1%)	6 (1.9%)	269 (86.2%)	0.93

Tabulka 4: Výsledky pro modely natrénované na 40ti gestech pro každý model. Testovaná gesta jsou odlišná od gest použitých pro tvorbu modelu.

Posledním testem je test výkonu aplikace. Testováno bylo na videích s rozlišením 320 na 240 pixelů při třech natrénovaných modelech gest. Na počítači s procesorem s frekvencí 1.4 GHz a s operační pamětí 704 MB byla průměrná rychlost zpracování snímků 16.8 snímku za vteřinu. Tato rychlost je dostatečná pro detekci gest v reálném čase.

Testování ukázalo závislost složitosti gesta na pravděpodobnosti jeho rozpoznání. Jednoduchá vertikální a horizontální gesta složená pouze ze dvou pohybů jsou rozpoznávána téměř bezchybně. Naopak u kruhového gesta je úspěšnost rozpoznání gesta závislá na kvalitě provedení gesta a to zejména na jejich rozdílných proporcích.

Z výsledků druhého testu rovněž vyplývá, že systém vykazuje lepší výsledky v případě, kdy je gesto natrénováno stejnou osobou která pak systém používá. Naopak vysoký počet trénovacích gest nepřináší vyšší úspěšnost detekce gest. Pro kruhové gesto byly natrénovány modely z 20, 40, 60 a všech dostupných gest. Výsledky pro 40 a více trénovacích gest byly srovnatelné. Pro 20 trénovacích gest byla úspěšnost detekce nižší.

Dosažené výsledky jsou o něco vyšší, než reálné výsledky, protože v testech nebyly uvažovány chyby při rozpoznávání pokožky a hledání spojitosti mezi oblastmi. Testovány byly již takové trajektorie, které byly detekovány celé. Chyby při detekci trajektorie jsou závislé na konkrétních videích. U některých testovacích videí k nim téměř nedocházelo, u jiných byla tato chybovost již výraznější. Je obtížné tuto chybovost vyjádřit matematicky, ale hrubým odhadem založeným na počítání přerušovaných a nepřerušovaných trajektorií, které mohli být potenciálními gesty je tato chybovost kolem osmi procent.

5 Závěr

Předem stanoveným cílem této bakalářské práce bylo navrhnout a implementovat aplikaci pro rozpoznávání gest ve videu. Cíl této práce byl splněn. Byly vytvořeny celkem čtyři aplikace. První aplikace implementuje celou detekci gest ve videu. Dále byly vytvořeny podpůrné aplikace pro tvorbu modelu pokožky a modelů jednotlivých gest. Poslední aplikace je určena pro testování úspěšnosti detekce gest.

Systém pro rozpoznávání gest v reálném čase musí splňovat několik základních požadavků. Tím prvním je rychlost potřebná pro zpracování jednoho snímku. Testováním bylo prokázáno, že je aplikace schopná zpracovat více než patnáct snímků za vteřinu. Tato rychlost je pro nasazení v praxi dostačující.

Dalším požadavkem je přesnost rozpoznávání gest. Pro jednoduché horizontální a vertikální gesto je přesnost dostatečná. Při testování se ukázalo, že tato gesta jsou rozpoznávána téměř se stoprocentní přesností. U složitějšího kruhového gesta je úspěšnost nižší. Pro nasazení v praxi jsou tak použitelná jednodušší gesta.

Předností zvoleného způsobu detekce gest je vyřešení problému nestejně rychlých gest. Mezi další přednosti systému patří možnost vytvoření nových gest a nového modelu barvy kůže. Program pro tvorbu modelu kůže umožňuje zvolit pro detekci i jinou barvu, než je barva kůže. Je tak možné systém používat tak, že si uživatel vezme do ruky dostatečně velký předmět, který bude mít ve snímané scéně výjimečnou barvu. Na tuto barvu se pak vytvoří nový model pro detekci oblastí. Tento přístup je vhodný, pokud scéna obsahuje rušivé elementy, které korektní detekci barvy kůže znemožňují.

Další výhodou je možnost definování nových gest. Aplikace disponuje prostředky pro uložení zvolených trajektorií z videa a tvorbu nového gesta z těchto trajektorií.

Nepřesná detekce složitějších gest dává možnost pro další rozvoj aplikace. Zejména by bylo možné změnit trénování modelu gest. Trénovací gesta by se mohla dělit na sobě podobná a výstupem by pak nebylo jedno referenční gesto, ale celá skupina těchto gest.

Další prostor pro pokračování práce je v řešení problému ztráty trajektorie při přechodu jedné ruky přes druhou. Také je možné zvětšit počet základních směrů u vektorů, které by vedlo k možnosti definovat složitější gesta.

Jako nejvhodnější aplikaci výsledků této bakalářské práce vidím možnost využití ovládní počítače gesty ruky v jednoduché počítačové hře. Hra by pro svůj provoz nepotřebovala složité zařízení, k jejímu provozu by stačil počítač a webová kamera, která se v poslední době řadí k stále běžnější výbavě počítače.

Literatura

- [1] Vezhnevets V., Sazonov V., Andreeva A. A Survey on Pixel-Based Skin Color Detection Techniques. Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003, dostupný z WWW <<http://graphics.cs.msu.su/en/publications/text/gc2003vsa.pdf>>
- [2] Wikipedia Color space [online] <http://en.wikipedia.org/wiki/Color_space>
- [3] IEEE Transaction on pattern analysis and machine intelligence, vol 24, no.5, may 2002, dostupný z WWW <<http://courses.csail.mit.edu/6.869/handouts/PAMIMeanshift.pdf>>
- [4] Wikipedia Momentová vytvořující funkce [online] <http://cs.wikipedia.org/wiki/Obecn%C3%BD_moment>
- [5] Wikipedia Hidden Markov model [online] <http://en.wikipedia.org/wiki/Hidden_Markov_model>
- [6] OpenCV Wiki [online] <http://opencvlibrary.sourceforge.net/>

Seznam příloh

Příloha 1. Manuál

Příloha 2. CD obsahující zdrojové texty a testovací data

Přílohy

1. Manuál k implementované aplikaci

Po zkompilování systému vzniknou celkem 4 spustitelné soubory. Pro korektní chování je třeba ve složce s těmito soubory vytvořit dva adresáře pro výstupní data. Adresář 'img' pro ukládání snímků videa a adresář 'traj' pro uložení trajektorií.

Následuje stručný přehled jednotlivých programů:

hist_ibp.exe

Popis:

Program pro generování modelu pokožky. Výsledný histogram se uloží do souboru 'histogram.bmp'.

Parametry:

1. cesta k bitmapě s výřezy pokožky, začerněné oblasti jsou přeskočeny
2. (nepovinný): cesta k bitmapě kde pokožka není, začerněné oblasti jsou přeskočeny

model_ibp.exe

Popis:

Program pro generování modelu gest. Výsledný model se uloží do souboru se shodným názvem jako je soubor se seznamem trajektorií doplněným o koncovku '.model'

Parametry:

1. cesta k souboru s trajektoriemi. Na každém řádku je očekávána cesta k jednomu souboru s trajektorií uloženou aplikací main_ibp.exe
2. šířka videa ze kterého byly trajektorie uloženy
3. výška videa ze kterého byly trajektorie uloženy

test_ibp.exe

Popis:

Program pro testování gest. Pro gesto je vybrán nejbližší model je vypsána do konzole pravděpodobnost shody. Zároveň jsou výsledky ukládány do souboru 'test.csv' pro další analýzu.

Parametry:

1. cesta k souboru s natrénovanými modely gest. Na každém řádku je očekávána cesta k jednomu modelu gest vytvořeného aplikací main_ibp.exe
2. šířka videa ze kterého byly trajektorie uloženy
3. výška videa ze kterého byly trajektorie uloženy

main_ibp.exe

Popis:

Hlavní program pro detekci gest v obraze. Vstupem je video, model pokožky a modely gest. Výsledky jsou po detekování každého gesta vypisovány do konzole. Program také disponuje prostředky pro ukládání trajektorií a snímků videa.

Parametry:

1. cesta k video souboru, nebo výraz 'cam' pro načítání videa z kamery
2. cesta k modelu pokožky vygenerovanému programem hist_ibp.exe
3. cesta k textovému souboru se seznamem natrénovaných gest. Na každém řádku tohoto souboru je očekávána cesta k modelu gesta vygenerovaného programem model_ibp.exe

Ovládání programu:

Klávesy a jejich funkce

p – zastavení / spuštění videa

k – zastavení videa a posun o jeden snímek dopředu

s – uložení aktuálního snímku videa a pravděpodobnostního ohodnocení pokožky do obrázků ve složce img

q / escape – ukončení programu

h – nápověda se seznamem příkazů

t – pozastavení videa a přechod do módu pro ukládání trajektorií

< , - přechod na předchozí trajektorii

> . - přechod na další trajektorii

s – uložení aktuální trajektorie do složky 'traj', dále je uložen obrázek trajektorie

v – přidání bodu trajektorie na její začátek

b – ubrání bodu trajektorie z jejího začátku

n – ubrání bodu trajektorie z jejího konce

m – přidání bodu trajektorie na její konec

h – nápověda se seznamem příkazů pro tento mód

q – ukončení režimu pro ukládání trajektorií