VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

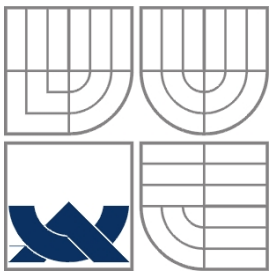
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁVÁNÍ POZIC A GEST

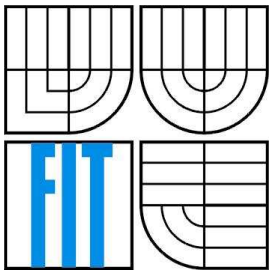
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE BC. LEOŠ JIŘÍK
AUTHOR

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

ROZPOZNÁVÁNÍ POZIC A GEST

RECOGNITION OF POSES AND GESTURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. LEOŠ JIŘÍK

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. DR. ING. PAVEL ZEMČÍK

BRNO 2008

Abstrakt

Práce se zabývá studiem současného stavu v problematice zpracování obrazu, zvláště s ohledem k rozpoznávání gest. Zmiňuje vybrané postupy jiných autorů a podrobuje je kritickému pohledu.
V druhé části se věnuje návrhu algoritmu, který by umožnil spolehlivé rozpoznávání gest v datech z projektů AMI a M4. Navrhují se prostředky zpřesnění informace o poloze účastníků a zpracování dynamických dat za účelem jejich přípravy k rozpoznávání. Je navržena možnost rozpoznávání gest pomocí směsi Gaussových funkcí a analýzy periodičnosti. Zkoumaná třída gest jsou gesta podporující řeč účastníka záznamu. Poslední část demonstruje výsledky a diskutuje další možný postup.

Klíčová slova

barevný model, segmentace, Gaussova funkce, normální rozložení, konvoluce, fitrování, exponenciální filtr, tracking, sledování regionů, směs Gaussových funkcí, rozpoznávání, klasifikace

Abstract

This thesis inquires the existing methods on the field of image recognition with regards to gesture recognition. Some methods have been chosen for deeper study and these are to be discussed later on. The second part goes in for the concenpt of an algorithm that would be able of robust gesture recognition based on data acquired within the AMI and M4 projects. A new ways to achieve precise information on participants' position are suggested along with dynamic data processing approaches toward recognition. As an alternative, recognition using Gaussian Mixture Models and periodicity analysis are brought in. The gesture class in focus are speech supporting gestures. The last part demonstrates the results and discusses future work.

Keywords

color model, segmentation, Gauss Function, normal distribution, convolution, filtering, exponencial filter, tracking, Gaussian Mixture Model, GMM, recognition, classification

Citace

Jiřík Leoš: Rozpoznávání pozic a gest. Brno, 2008, diplomová práce, FIT VUT v Brně.

Rozpoznávání pozic a gest

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Dr. Ing. Pavla Zemčíka.
Další informace mi poskytli dr. Mannes Poel, University of Twente, Nizození a Ing. Pavel Žák, FIT VUT, Brno.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Leoš Jiřík
10.5.2008

Poděkování

Rád bych poděkoval doc. Pavlu Zemčíkovi za vedení této práce. Dále Ing. Pavlu Žákovi za časté konzultace. V neposlední řadě bych chtěl poděkovat všem, kteří mi přímo i nepřímo pomáhali, aby tato práce vůbec vznikla. Velký dík vám všem.

Obsah

1 Úvod

Na poli počítačové vědy proběhl v uplynulých letech větší vývoj než kdykoli předtím za celou dobu její existence. Svět se globalizuje, vyvstávají do té doby nevídané možnosti komunikace, zpracování problémů a automatizace úkonů pomocí počítačů, nejen těch osobních. Tato věda se proto také více a více specializuje a jednotlivá její odvětví zkoumají do větších podrobností zatím nevyřešené problémy.

Jednou z velmi rychle se vyvíjejících částí informatiky – počítačové vědy – je zpracování multimediálních dat a hledání obecných informací v nich obsažených. Již nyní některé výsledky výzkumu v oblasti zpracování obrazových dat a rozpoznávání v nich slouží široké veřejnosti k zajištění bezpečnosti v místech velkého výskytu lidí, k ulehčení komunikace s počítači a i v dalších aspektech běžného života.

Cílem této práce je nalezení takových přístupů k řešení rozpoznávání v obrazových datech, které mohou sloužit k lepšímu popisu, které dynamické děje se v nich odehrávají. Statické informace nemohou do důsledku popsat časově závislé děje. A protože lidská gesta i mimika jsou ději časově závislými, bude tato práce převážně o časově proměnných dějích a jejich vyjádřením prostředky počítačové vědy a zpracování obrazu.

Jak uvádí první část této práce, úkol zpracování lidských gest je stále velmi otevřeným problémem, a tudíž tato práce nemůže popsat všechny možnosti, respektive vybrat tu obecně nejlepší. Již proto, že je v současném stavu poznání stále neznáma. Omezuje se tedy pouze na určitou část a vychází z předpokladů, které nemusí být obecně platné. Zadání samotné specifikuje jako vstup data získaná v rámci projektů *AMI* a *M4*, pro podrobnosti viz [1]. Již tím se otevírá možnost apriorních (dopředných) předpokladů, jako je počet osob ve scéně, stálost pozadí scény apod.

Jednotlivé části této práce popisují současný stav poznání problematiky rozpoznávání gest, výhody a nevýhody jednotlivých řešení a popis vlastního experimentu v intencích zadání.

Kapitola druhá (následující za touto úvodní) shrnuje zvolené přístupy autorů v podobné problematice. Zvláště se zabývá barevnými modely, segmentací obrazu na základě barvy, klasifikací (přiřazením tříd) nalezených regionů, metodami sledování regionů a rozpoznávání časově závislých sekvencí. Vybrané části přístupu různých autorů je možné také nalézt v příloze. Hodnocení jednotlivých popsaných metod spolu s technickým východiskem k praktické části této práce je obsaženo v části třetí.

Popis návrhu a implementace v rámci této práce vytvořeného algoritmu se nachází v kapitole 4. Postupováno je zde od základních aspektů, jako je volba barevného modelu, metod zpracování obrazu a regionů a sledování až k rozpoznávání časových sekvencí. Základní popis vytvořených programů

spolu s ukázkou fungování lze nalézt v kapitole páté. Podrobný popis včetně ovládání programů je uveden v příloze.

Závěrečná kapitola je shrnutím obsahu práce. Zmiňuje také možnosti jejího dalšího pokračování, hodnotí z různých pohledů dosažené výsledky a formuje celkový rámec vyznění tohoto díla.

2 Současný stav

V této kapitole bych rád popsal různé již existující přístupy nejen k rozpoznávání gest, ale též k segmentaci obrazu a rozpoznávání z jednoho snímku a k dalším důležitým aspektům této problematiky.

2.1 Zpracování a segmentace jednoho snímku

Přestože tato práce zaměřuje na dynamické děje v obrazových sekvencích, při téměř každém zpracování obrazu jsou nutné kroky, které vychází pouze z informace statické (jednoho snímku, půlsnímku, atd.).

Jednotlivé po sobě jdoucí kroky ve zpracování obrazu jsou jeho sejmutí a digitalizace, předzpracování, segmentace – nalezení jednotlivých objektů či jejich částí a posledně potom klasifikace či hlubší porozumění a reprezentace objektů.

2.1.1 Sejmutí a digitalizace obrazu

Obraz jako takový je vlastně měřením intenzity dopadu světla nebo jeho jednotlivých barevných složek z různých částí scény. Převedením signálu – naměřené veličiny – do digitální podoby vzniká digitalizovaný obraz skládající se z diskrétních hodnot obrazových elementů – *pixelů*.

V současné době existuje veliké množství hardwarových prostředků – kamer, čipů, optických soustav, které jsou nutné pro snímání obrazu a jejich popis či jen výčet je nad rámec rozsahu tohoto textu, navíc mnohé z jejich parametrů přímo neovlivňují řešení zadání práce.

2.1.2 Předzpracování obrazu

Metody předzpracování obrazu jsou prvním nutným krokem s cílem jeho porozumění. Těmito kroky se zvětšuje přesnost metod vyšších úrovní, např. odstraněním šumových hodnot, převedením vstupních dat do nejvýhodnější podoby apod.

Matematicky vzato je obraz dvourozměrnou funkcí s několika možnými obory funkčních hodnot:

$$f(x, y) \in D \tag{1}$$

kde x a y jsou diskrétní souřadnice pozice obrazového elementu a množina D může být množinou konečného počtu celých čísel – tzv. *černobílý obraz* (anglicky *gray-scale*), či konečnou množinou dvou-, tří- i čtyřprvkových vektorů.

Základním oborem hodnot obrazové funkce barevného obrazu je tzv. *RGB* model, kde první složka vektoru udává intenzitu červené, druhá zelené a třetí modré barevné složky. Navzdory

jednoduchosti práce s tímto modelem není tento vhodný pro detekci nebo rozpoznávání objektů barvy lidské pokožky. Za tímto účelem se zavádí modely jiné, vhodnější:

- *normalizovaný* RGB model,
- *HSV* model,
- *YCrCb* (chromační) model a jiné.

První výše uvedený model lze získat ze standardního RGB modelu pomocí vztahů:

$$r = \frac{R}{R+G+B} \tag{2}$$

$$g = \frac{G}{R+G+B} \tag{3}$$

Jelikož pro normalizované složky platí, že $r + g + b = 1$, je složka b redundantní. Tento model využil pro segmentaci objektů barvy lidské pokožky Ap-apid ve své práci o automatické detekci pornografických materiálů [2] nebo Jae a Suk v práci [3] na téma automatické segmentace barvy lidské pokožky a urychlení jejího výpočtu.

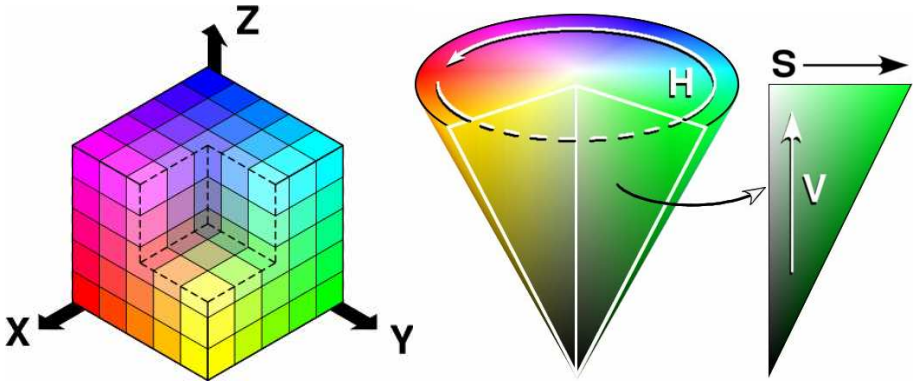
Druhým modelem je tzv. HSV, jehož jednotlivé složky jsou barevný odstín (anglicky *hue*), sytost barvy (*saturation*) a hodnota jasu (*value*). Převodní vztahy jsou následující:

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \tag{4}$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B} \tag{5}$$

$$V = \frac{1}{3}(R+G+B) \tag{6}$$

Vhodnost využití tohoto modelu pro detekci barvy kůže diskutuje práce Vezhnevetse a kol. [4]. Obecně lze říci, že tento model a modely jemu příbuzné jako je *HSL* či *HSI* v mnoha aplikacích hledání barvy pokožky využity nejsou. Jedním z příkladů takovéhoto přístupu je práce autorů Brashearové, Parka a Leea [5].



Obr. 2-1 – RGB a HSV barevné modely

Oproti tomu poslední zmiňovaný model – YCrCb – byl využit mnohokrát. Lze jej také převést ze základního RGB modelu, a to těmito vztahy:

$$Y = 0{,}299R + 0{,}587G + 0{,}114B \tag{7}$$

$$C_r = R - Y \tag{8}$$

$$C_b = B - Y \tag{9}$$

Vztah (7) je empirické převedení RGB modelu na jasovou hodnotu. Využít lze i pro převod obrazu do stupňů šedi. Z prací, v nichž autoři využívají tohoto modelu lze jmenovat např. Face Segmentation in Videophone Applications, Chai a Ngan [6], kde je využit fakt, že v tomto modelu zaujímá barva kůže jen určitou, dobře oddělitelnou část celého rozsahu barevného modelu. Ostatně tuto vlastnost v různých modelech využívají všechny přístupy segmentace objektů lidské kůže na základě barvy, jak je uvedeno v následující části.

2.1.3 Segmentace obrazu

Důležitým krokem k nalezení regionů barvy lidské pokožky je vytvoření modelu určitého rozsahu, který odpovídá (modeluje) obrazovým elementům z trénovací množiny. V současných pracích bývá využito několik základních principů.

Modelování jedním Gaussiánem

Gaussián (přesněji *Gaussova funkce*) je charakterizována *střední hodnotou* a *rozptylem*. Ve vícedimenzionálních případech je rozptyl nahrazen *kovarianční maticí* (11) a střední hodnota *vektorem středních hodnot* (10).

$$\mu_{skin} = \frac{1}{n}\sum_{j=1}^{n} c_j \tag{10}$$

$$\Sigma_{skin} = \frac{1}{n-1}\sum_{j=1}^{n} (c_j - \mu_{skin})(c_j - \mu_{skin})^T \tag{11}$$

Celkový počet vzorků c_j z trénovací množiny je n. Pro přiřazení neznámého vzorku c do buď do množiny *skin* nebo *non-skin* se využívá většinou přímo hodnota funkce hustoty pravděpodobnosti $p(c|skin)$:

$$p(c \mid skin) = \frac{1}{\sqrt{2\pi^d \left|\Sigma_{skin}\right|}} e^{-\frac{1}{2}(c-\mu_{skin})^T \Sigma_{skin}^{-1}(c-\mu_{skin})} \tag{12}$$

Směs více Gaussových funkcí

Jedná se o techniku sofistikovanější, nicméně zde roste riziko, že daný model bude přeučen, tzn. nebude dostatečně správně generalizovat – zevšeobecňovat – hledaný výřez barevného modelu. Využívá váženou směs funkcí popsaných v předchozím odstavci.

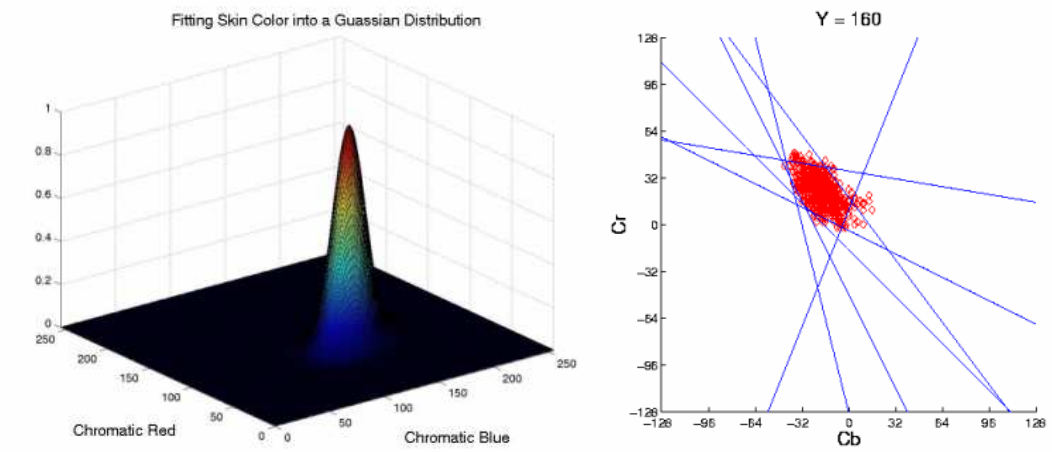
$$p(c \mid skin) = \sum_{i=1}^{k} \pi_i . p_i(c \mid skin) \tag{13}$$

Výslednou pravděpodobností je suma k pravděpodobností p_i násobených vahou π_i.

Model eliptický a jiné

Lze nalézt i modely složitější, například modelování shluku pixelů pomocí *eliptické hranice*, *vyhledávací tabulkou* (anglicky *Look-up Table*, LUT) či *samoorganizující mapou* (*Self Organizing Map*, SOM). První z výše jmenovaných autoři Lee a Yoo podrobně studují ve své práci z roku 2002. Navrhují v přípravném kroku provézt odstranění okrajových hodnot, tj. pixelů, které jsou v trénovací množině zastoupeny pouze malým počtem vzorků. Trénování probíhá podobně jako trénování modelu jednoho Gaussiánu. Podle autorů je nespornou výhodou práce s eliptickou hranicí rychlost vyhodnocení neznámého vzorku.

Popis a možnosti dalších modelů je možné najít spolu s diskusí jejich přesnosti v práci [3].



Obr. 2-2 –Příklady modelování barvy lidské pokožky

2.1.4 Klasifikace

Klasifikace je jedním z nejobecnějších problémů počítačového vidění; jedná se o úkol nalezení zobrazení přiřazující objektu O nějakou třídu C_k z N známých tříd, tj. $1 \leq k \leq N$:

$O \rightarrow C_k$

V problematice klasifikace objektů odpovídajících částem lidského těla bylo provedeno mnoho experimentů hodnotících implementace různých metod. Jedna skupina metod klasifikuje objekt analýzou *spojitého regionu*, zatímco jiné jej popisují a (následnou) klasifikaci provádějí pomocí různých druhů *charakteristik* nazývaných též *příznaky* (anglicky *features*).

Příznaky získané analýzou spojitých regionů

Autoři práce [7] navrhují detekci obličeje pomocí několika příznaků spojitého regionu objektu barvy lidské pokožky. Jsou jimi *kompaktnost*, *hustota* a *orientace*, v tomto pořadí:

$$C = \frac{A}{P^2} \tag{14}$$

$$S = \frac{A}{D_x D_y} \tag{15}$$

$$O = \frac{D_y}{D_x} \tag{16}$$

kde A je plocha (počet pixelů) spojitého regionu, P je jeho obvod, D_x a D_y jsou rozměry nejmenšího obalujícího obdélníku, jehož strany jsou rovnoběžné s osami (anglicky *bounding box*, BB). Zmíněná práce také uvádí empirické hodnoty pro jednotlivé příznaky.

Jinou možností popisu spojitého diskrétního objektu (anglicky *blob*) je využití tzv. *momentů* definovaných následovně:

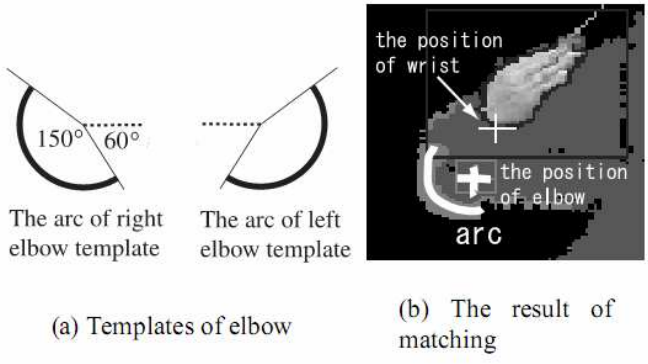
$$M_{ij} = \sum_S x^i y^j I(x, y) \tag{17}$$

$$S = \{(x, y) \mid (x, y) \in BLOB\} \tag{18}$$

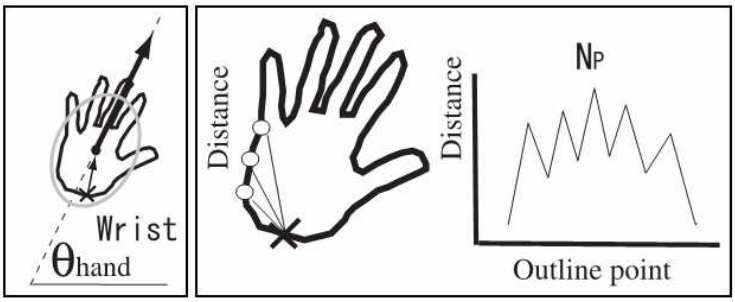
Někdy se v sumě (17) nahrazuje hodnota intenzity obrazu $I(x,y)$ za 1. Potom je objekt charakterizován jako uniformní a ve výpočtu jeho momentů nehrají roli jasové změny obrazu v místě jeho plochy.

Jednou z prací využívající momenty pro detekci částí lidského těla je [8], v níž autoři jako kritérium pro klasifikaci navrhují poměr délek hlavní a vedlejší poloosy elipsy určené pomocí normalizovaných momentů – tj. momentů vypočítaných z objektů, jejichž těžiště bylo přesunuto do středu souřadného systému.

Autoři práce na téma rozpoznávání japonského znakového jazyka [9] rozšiřují množinu příznaků pro detekci a klasifikaci rukou o pozici loketního kloubu a počet výběžků na hranici regionu ruky. Obrázek 2-3 demonstruje nalezení lokte jako oblouku určitého úhlového rozsahu a obrázek 2-4 nalezení počtu výběžků v regionu ruky.



Obr. 2-3 – Nalezení pozice loketního kloubu



Obr. 2-4 – Orientace ruky a nalezení počtu výběžků obvodu ruky

Další metody získávání charakteristik

Příznakové vektory nemusejí být vždy sekvence několika málo charakteristik, které intuitivně popisují region ruky, ale také dlouhé vektory vzniklé jako výsledek nějaké transformace. Příkladem může být *transformace Fourierova* dávající přesnou informaci o frekvencích obsažených v obraze.

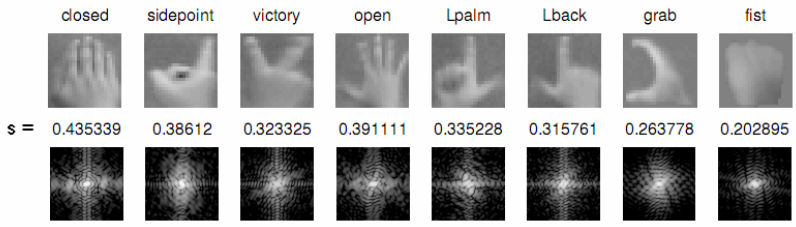
Autoři Kölsch a Turk v práci [10] navrhují klasifikátor pozice ruky následovně: nejdříve provedou Fourierovu transformaci výřezu obsahujícího ruku v určité pozici (19), následně odečtením frekvencí vzniklých ořezáním obrazu získávají rozdílovou transformaci *D(u, v)* (20). Rovnice (22) je normalizovaným součtem amplitud jednotlivých frekvencí a – jak je patrné z obrázku 2-5 – pro každou z uvažovaných pozic ruky má jinou hodnotu, je tedy možné klasifikovat regiony ruky podle ní.

$$F(u,v) = \frac{1}{N*M}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I(m,n)e^{-i2\pi(\frac{mu}{M}+\frac{nv}{N})} \tag{19}$$

$$D(u,v) = \log\left|F(u,v) - P(u,v)\right| \tag{20}$$

$$P(u,v) = \frac{1}{N*M}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\frac{1}{2}e^{-i2\pi(\frac{mu}{M}+\frac{nv}{N})} \tag{21}$$

$$s = e^{\frac{1}{k}\sum_{u,v} D(u,v)} \tag{22}$$



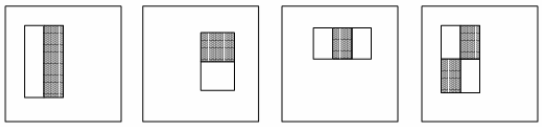
Obr. 2-5 Fourierův deskriptor

Jiný přístup využili Hoshino a Tanimoto v práci [11] Realtime Estimation of Human Hand Posture. Region ruky charakterizují hodnotami *autokorelační funkce*:

$$R(\vec{a}) = \sum_{\vec{r}} I(\vec{r}) I(\vec{r} + \vec{a}) \quad (23)$$

kde ***a*** a ***r*** jsou vektory pozic obrazových elementů. Takto je tedy signál (obraz) charakterizován vektorem hodnot, které značí, do jaké míry je obraz *soběpodobný*.

V poslední době se objevuje stále více přístupů založených na přístupu navrženém Violou a Jonesem [12], totiž využití příznaků získaných z *Haarových struktur* (anglicky *Haar-like Structures*). V podstatě jde o sumu přes hodnoty obrazové funkce vynásobené dvouhodnotovou masku. Obrázek 2-6 zobrazuje některé možné *masky*. V sumě se objeví hodnoty pixelů v bílých obdélnících s kladným znaménkem a hodnoty v šedých obdélnících se znaménkem záporným. Jelikož počet všech možných masek je neúměrně veliký, větší než počet bodů v obraze, je nutné nalézt takový algoritmus, který vybere nejvhodnější množinu masek, z níž sestaví příslušný klasifikátor. Autoři Viola a Jones využívají varianty algoritmu *AdaBoost*, který uvádím v příloze 1.



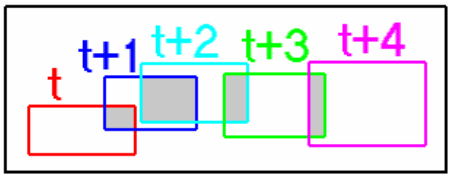
Obr. 2-6 – Příklady masek používaných Violou a Jonesem

2.2 Tracking – sledování objektů

Sledování regionů je krokem od statického k dynamickému zpracování. Řeší problém korespondence nalezených regionů v různých časech. Základní otázkou je, jaký region z *N* nalezených v čase *t* ($R^t_{1..N}$) považovat za odpovídající výskyt regionu *i* v čase *t-1* (R^{t-1}_i). Existuje více technik, z nichž dvě uvádím pro svou jednoduchost respektive robustnost.

2.2.1 Metoda překrývajících se boxů

Tento přístup vychází z jednoduchého předpokladu, že v časově přímo následných snímcích (časy *t-1* a *t*) dochází ke změně pozice a velikosti regionu tak, že korespondentní obalové obdélníky jednotlivých regionů se překrývají (obrázek 2-7).



Obr. 2-7 – Sledování obalových obdélníků

Vlastnost překrytí lze snadno detekovat tak, že jeden či více rohů jednoho obdélníku leží v obdélníku druhém či obráceně.

V diskusní části práce [13] autoři zmiňují způsoby řešení případů, kdy

1. jeden obdélník překrývá více obdélníků v následujícím čase,

2. obdélník v následujícím čase nepřekrývá žádný v předchozím čase a
3. obdélník v dřívějším čase nekoresponduje s žádným z obdélníků následujících.

Řešení těchto případů je věcí vlastní implementace, bod 3 lze považovat za okamžik, kdy došlo k zakrytí či vymizení objektu ze scény, oproti tomu bod 2 za případ jeho opětovného objevení. Nejsložitější je případ 1, kdy je buď nutné rozhodnout, který z více možných regionů odpovídá předchozímu (například pomocí klasifikace regionů – část 2.1.4) nebo i pomocí dynamické charakteristiky – vybere se ten region, jehož poloha nejlépe odpovídá předchozí trajektorii v časech *t-n* až *t-1*.

2.2.2 Sledování generováním a potvrzováním hypotéz

V práci [14] z roku 2004 navrhují Argyros a Lourakis novou metodu sledování objektů barvy kůže. Je založena na generování *hypotéz* výskytu nějakého objektu a jejich potvrzení a zpřesnění pomocí nalezených bodů pokožky.

Základem této metody je nalezení hypotézy výskytu objektu, kterou je elipsa o parametrech $h = (c_x, c_y, \alpha, \beta, \theta)$, kde první dva parametry jsou souřadnice středu, další dva délky hlavní a vedlejší poloosy a poslední potom úhel natočení elipsy (její hlavní osy). V každém obrazu existuje M spojených regionů b_j a sledováno je N objektů o_i. Definují se tři množiny

$$B = \bigcup_{j=1}^{M} b_j \tag{24}$$

$$O = \bigcup_{i=1}^{N} o_i \tag{25}$$

$$H = \bigcup_{i=1}^{M} h_i \tag{26}$$

tedy množina pixelů barvy pokožky, množina bodů objektu a množina elips (hypotéz) v tomto pořadí. Dále se definuje metrika $D(p, h)$ vzdálenosti bodu $p = (x, y)$ od elipsy h, po níž platí, že D(p, h) < 1 pokud bod leží uvnitř elipsy a D(p, h) > 1 pokud je bod mimo elipsu:

$$D(p,h) = \sqrt{\vec{v}.\vec{v}} \tag{27}$$

$$\vec{v} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{pmatrix} \frac{x - x_c}{\alpha} & \frac{y - y_c}{\beta} \end{pmatrix} \tag{28}$$

Nyní zbývá vyřešit problémy vytvoření hypotézy pro nově nalezený spojený region (blob), sledování regionů v čase, odstranění hypotézy v případě vymizení nebo zastínění objektu a posledně predikce hypotéz.

Vytvoření hypotézy

Platí-li pro nějaký blob *b* podmínka (29), je nutné vytvořit novou hypotézu *h*, tj. nalézt elipsu, která daný blob charakterizuje. Rovnice pro nalezení jejích parametrů jsou obsaženy v příloze 1. V každém časovém kroku *t* je každý blob podroben testování, zda uvedená podmínka platí, pokud ano, je pro něj vytvořena příslušná hypotéza.

$$\forall p \in b, \min_{h \in H}(D(p,h)) > 1 \tag{29}$$

Sledování hypotéz

Sledování vychází ze dvou základních pravidel, kterými se vytváří množiny obrazových bodů *o*, jež potvrzují danou hypotézu:

- **Pravidlo 1:** Pokud pixel barvy pokožky leží uvnitř nějaké elipsy (použitím rovnice (27)), je přiřazen této hypotéze.
- **Pravidlo 2:** Pokud takovýto pixel leží vně všech elips, potom je přiřazen té hypotéze, které leží nejblíže (opět použitím (27)).

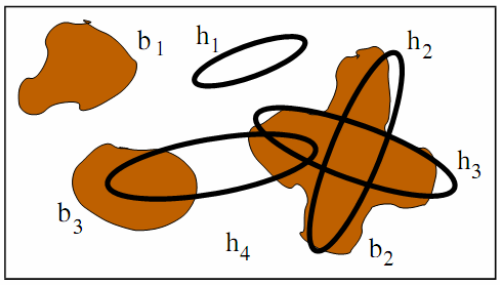
Formálně lze tedy napsat množinu bodů *o* nějakého objektu přiřazených hypotéze *h* následovně:

$$o = R_1 \cup R_2 \tag{30}$$

$$R_1 = \{p \in B \mid D(p,h) < 1\} \tag{31}$$

$$R_2 = \{p \in B \mid D(p,h) = \min_{k \in H}(D(p,k))\} \tag{32}$$

V praxi může nastat případ, kdy jedna hypotéza *h* je podporována body z více blobů (tedy body uvnitř této elipsy pochází z různých spojených regionů). Zde autoři navrhují následující postup. Pokud existuje právě jeden blob *b* podporující hypotézu *h*, je tato přiřazena *b*. Jinak je přiřazena regionu, se kterým sdílí nejvíce bodů (z nějž nejvíce bodů podporuje *h*). V obrázku je to případ hypotéhy h_4, jež je nakonec přiřazena blobu b_3.



Obr. 2-8 – Hypotéza h_1 je odstraněna, pro blob b_1 je vytvořena hypotéza nová, body regionu b_2 jsou rozděleny mezi h_2 a h_3 – některé budou podporovat obě hypotézy

.

Odstranění hypotézy

Pro existující hypotézu *h* může nastat případ, kdy žádný bod barvy pokožky nepodporuje přímo její existenci (33), tj. neleží uvnitř příslušné elipsy žádný takovýto bod.

$$\forall p \in B, D(p,h) > 1 \tag{33}$$

Potom takováto hypotéza musí být odstraněna z množiny hypotéz *H*. V praxi ovšem autoři takovouto hypotézu ponechávají po určitou dobu v platnosti, aby zamezili vlivu případné špatné segmentace barvy pokožky. Délku takového intervalu navrhují asi půl sekundy, tedy kolem 14 snímků sekvence.

Predikce

Pro každý následující snímek je třeba vytvořit hypotézy tak, aby odchylka hypotézy od skutečného zjištěného stavu byla co nejmenší. Argyros a Lourakis navrhují postup, který lze vágně vyjádřit jako „z nedávné minulosti se určí nedaleká budoucnost“, tedy z lineárního posunu v minulých snímcích určí hypotézy pro následující snímek.

Je-li $h_i = (c_x, c_y, \alpha, \beta, \theta)$ jistou hypotézou v čase *t-1*, potom tatáž hypotéza v čase *t* má tvar $h_i = (c_x', c_y', \alpha, \beta, \theta)$. Označme $(c_{xi}(t), c_{yi}(t)) = C_i(t)$. Střed nové elipsy – hypotézy – lze napsat následovně:

$$C_i(t) = \left(c_x', c_y'\right)_i = C_i(t-1) + (C_i(t-1) - C_i(t-2)) \tag{34}$$

Druhý člen součtu je v podstatě vektorem lineárního posunu v posledním snímku *t-1*.

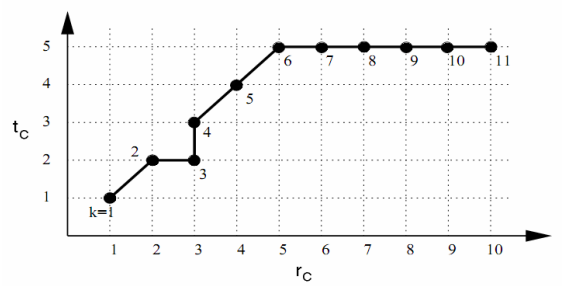
2.3 Rozpoznávání gest časově závislých

Každé gesto, respektive nějaká aktivita, je charakteristické svojí časovou závislostí. V následující části bych rád uvedl stávající přístupy k popisu dynamické závislosti aktivit a gest, tedy formálněji řečeno k časové závisti stavů nějakého systému.

2.3.1 Dynamické borcení času

Tento přístup je jedním ze základních. Hlavní myšlenkou je porovnání, tj. pravděpodobnostní ohodnocení, dvou různých sekvencí symbolů (příznakových vektorů). Není zde kladen přímý požadavek na délku těchto sekvencí.

Mějme dvě sekvence symbolů ***r*** a ***t***. Definujeme cestu D_C charakterizovanou jednoznačně její délkou K_C a průběhem funkcí $r_C(k)$ a $t_C(k)$, kde $1 \le k \le K_C$. Viz obrázek 2-9.



Obr. 2-9 – Jedna z možných cest v DTW

Sekvence ***r*** je *referenční* a ***t*** je *testovaná* sekvence. Jednotlivé třídy sekvencí jsou reprezentovány buď jedním nebo více referenčními vzory. V případě, že vzorů – referencí – je více, existuje několik přístupů k určení výsledného ohodnocení. Je možné z několika vzorů stejné třídy určit průměrný vzor nebo výsledné ohodnocení určit jako minimum z ohodnocení porovnání testované sekvence se všemi referenčními.

Vzdálenost sekvencí přes nějakou cestu $C = (K_C, r_C(k), t_C(k))$ je dána takto:

$$D_C(\vec{t},\vec{r}) = \frac{\sum_{k=1}^{K_C} d\left[\vec{t}(t_C(k)),\vec{r}(r_C(k))\right] W_C(k)}{N_C} \tag{35}$$

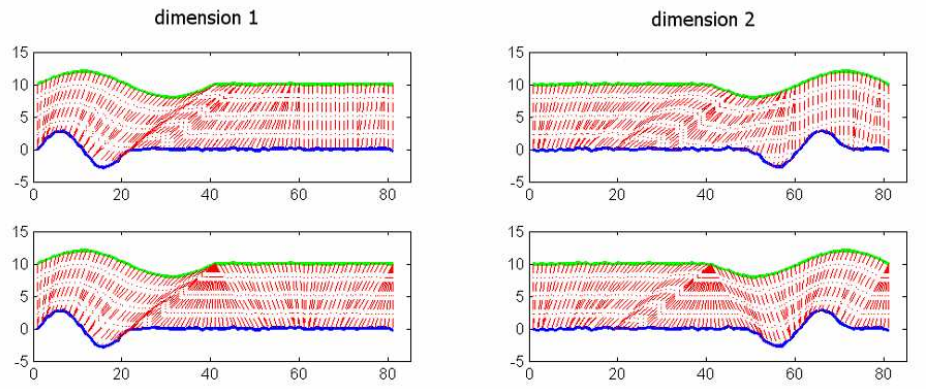
kde $d[s_1,s_2]$ je vzdálenost dvou symbolů – vektorů, $W_C(k)$ je váha odpovídající k-tému kroku cesty a N_C je normalizační faktor závislý na vahách cesty $W_C(k)$.

Vzdálenost sekvencí je definována jako minimální vzdálenost přes všechny možné cesty:

$$D(\vec{t},\vec{r}) = \min_C D_C(\vec{t},\vec{r}) \tag{36}$$

Existuje několik aspektů, jež je třeba zvážit při návrhu rozpoznávacího algoritmu založeného na *DTW*, jako je omezení cesty jen na ty blízké diagonálnímu průběhu, návrh vah a normalizačního faktoru. V materiálu [15] jsou příslušné problémy spojené s tímto diskutovány. V příloze 1 uvádím pseudokód demonstrující práci rozpoznávače DTW pomocí určení vzdálenosti dvou sekvencí.

Metody dynamického borcení času využilo několik autorů ve svých výzkumech s cílem rozpoznávání gest, nicméně často se jedná o určité varianty DTW. V nezměněné formě využívá DTW například Corradini [16]. V práci [17] demonstrují autoři vhodnost *vícedimenzionálního* DTW (anlgicky *Multi-Dimensional Dynamic Time Warping*, MD-DTW). Jedná se o variantu, která řeší problém nalezení pouze *suboptimální* cesty v případě sekvencí složených z vícedimenzionálních stavových vektorů. Navržený přístup vkládá před vlastní proces nalezení optimální cesty DTW kroky normalizace referenční i testované sekvence na nulovou střední hodnotu a jednotkový rozptyl. Obrázek 2-10 zobrazuje rozdíly obou přístupů: horní řádek DTW bez normalizace a spodní s normalizací; modrý průběh je referenční a zelený testovaný, červené spojnice značí porovnané vzorky.



Obr. 2-10 – Suboptimální (horní) a optimální (spodní) cesta DTW ve dvou dimenzích

2.3.2 Skryté Markovovy modely

Skrytý Markovův model (SMM, anglicky *Hidden Markov Model*, HMM) je statistický model, u nějž předpokládáme, že modeluje systém, který je *Markovovým procesem* (formálně – znalost několika minulých stavů nepřináší o rozložení pravděpodobnosti současného stavu více informace nežli znalost jediného – toho posledního z nich).

HMM je pětice $(\pi, \Omega_X, \Omega_O, A_{NxN}, B_{MxN})$, kde $\pi = (\pi_1, \ldots \pi_N)$ je pravděpodobnost počátku v odpovídajícím stavu, Ω_X je množina stavů, Ω_O množina pozorovaných symbolů, A_{NxN} je matice pravděpodobností přechodů (a_{ij} je pravděpodobnost přechodu ze stavu i do stavu j), B_{MxN} je matice pravděpodobností vyslání daného symbolu v daném stavu (b_{ji} je pravděpodobnost, že ve stavu i byl vyslán symbol j). Platí zde několik omezení:

$$\sum_j a_{ij} = 1, \forall i \tag{37}$$

$$\sum_j b_{ji} = 1, \forall i \tag{38}$$

tedy: součet pravděpodobností, že ze stavu i systém přejde do nějakého stavu, je roven jedné a součet pravděpodobností, že ve stavu i vyšle nějaký symbol, je také roven jedné.

Model, v němž platí, že hodnota všech pravděpodobností návratu do předcházejícího stavu je rovna nule, se nazývá *dopředný* model. Platí zde, že je-li model v čase t ve stavu i, pak v čase $t+1$ musí být některém ze stavů j takových, že $i \leq j$.

Základní úkoly při práci s HMM jsou následující (předpokládáme znalost struktury modelu, tj. počet stavů a počet vysílaných symbolů):

1. Určení pravděpodobnosti vyslání sekvence pozorování $\boldsymbol{O}$, jsou-li dány parametry modelu $\lambda = (\pi, A, B)$.
2. Nalezení nejpravděpodobnější sekvence stavů, jimiž projde HMM, je-li dána sekvence symbolů $\boldsymbol{O}$.
3. Nalezení parametrů modelu $\lambda = (\pi, A, B)$ tak, aby byla maximalizována pravděpodobnost, že přijme známou sekvenci $\boldsymbol{O}$, tj. maximalizace $P(\boldsymbol{O}|\lambda)$.

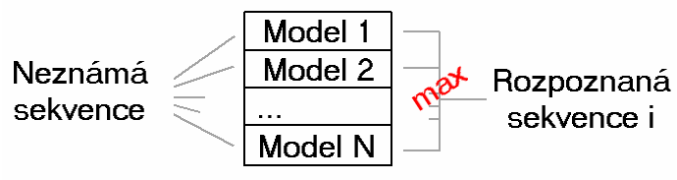
První problém je možné řešit pomocí výpočtu částečných dopředných či zpětných věrohodností (anglicky *likelihood*), druhý potom řeší tzv. *Viterbiho algoritmus* výběru maxima přes všechny možné přechody. Nejsložitějším problémem je trénování modelu, tedy úprava parametrů modelu tak, aby byla maximalizována pravděpodobnost přijetí dané sekvence. Zatím není znám způsob jak analyticky nalézt globální maximum, tedy model λ^* takový, že

$$\lambda^* = \arg\max_{\lambda} P(\vec{O} \mid \lambda) \tag{39}$$

Nicméně existuje způsob, jak nalézt model λ' takový, že $P(\mathbf{O}|\ \lambda') \geq P(\mathbf{O}|\ \lambda)$. Je jím tzv. *forward-backward algoritmus*, nazývaný též *Baum-Welchův algoritmus* (obdoba *EM algoritmu* spočívající v opakování kroků výpočtu dopředných a zpětných částečných pravděpodobností a reestimace parametrů modelu ze spočtených pravděpodobností).

Přesný popis včetně matematického pozadí k výše zmíněným problémům je možné nalézt v [15] na stranách 112-123.

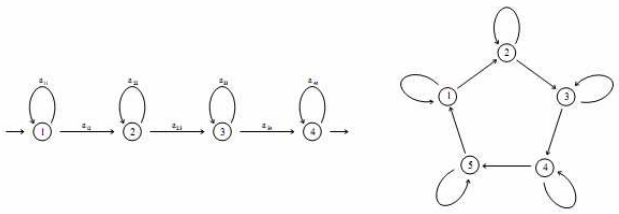
Typické schéma rozpoznávače využívajícího HMM je znázorněno na obrázku 2-11. Spočívá v natrénování jednoho modelu pro každou z rozpoznávaných tříd a výběru modelu s maximální pravděpodobností vyslání neznámé sekvence:



Obr. 2-11 – Rozpoznávání pomocí HMM výběrem třídy s maximální pravděpodobností

Způsobů využití skrytých Markovových modelů v problematice rozpoznávání dynamických procesů je mnoho. Objevují se v *diskrétní* i *spojité* verzi (pravděpodobnosti vyslání daného symbolu nejsou dány výčtem pro jednotlivé symboly, ale Gaussovou funkcí). Diskrétní HMM využívají například Fraile a Maybank [18] ke klasifikaci trajektorií projíždějících vozidel. Nam a Wohn [31] využili tentýž pro rozpoznávání deseti tříd trajektorií rukou. Ve shrnutí své práce diskutují vhodnost rozšíření trénovací množiny na více než 200 příkladů. Přesnost HMM výrazně vzrůstá s počtem příkladů nad sto.

Rigoll a kolektiv [19] oproti tomu nezvolili zvolili cestu apriorní diskretizace vektorů vysílaných HMM, ale implementovali spojité modely (anglicky *Continuous* HMM). Cílem jejich práce bylo rozpoznávání gest a aktivit celých postav v sekvencích s uniformním (jednotným) pozadím. Topologii modelů volí pro každé gesto jinou – pro lineární gesta dopředné a pro periodická gesta *cyklické* modely (obrázek 2-12).



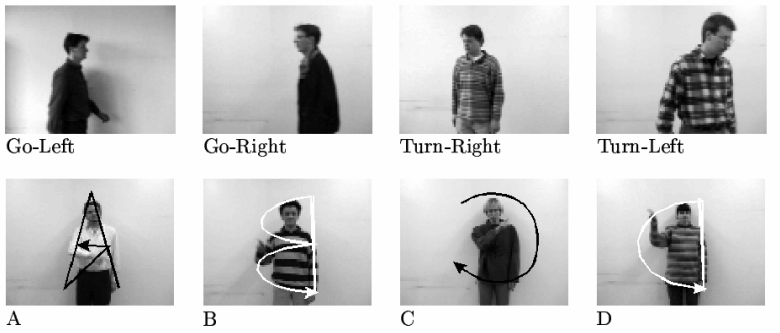
Obr. 2-12 – Dvě různé topologie HMM

Metod zpracování obrazu či videosekvence a získávání z nich informací existuje mnohem více než bylo uvedeno, jejich popis je však mimo možnosti této práce. V další kapitole jsou současné výše zmíněné přístupy a práce diskutovány a podrobeny kritickému pohledu.

3 Zhodnocení současného stavu

Z výše uvedeného dílčího výčtu přístupů různých autorů k problematice rozpoznávání gest je zřejmé, že mnohé systémy (či výzkumné projekty) vycházejí z předpokladů, které je sice možné v laboratorních podmínkách simulovat – použít vhodně nahrané sekvence – nicméně jejich splnění v reálných podmínkách nebo podmínkách blížících se skutečné v praxi využitelné podobě je velmi obtížné až nemožné.

Většina prací se zabývá rozpoznáváním dostatečně výrazných gest, jejichž rozlišení pouhým pohledem je zřejmé. Na stránkách [20] je uveden podrobný výčet systémů rozpoznávání gest založený na rozpoznávání obrazu. Více než 80% z uvedených prací vedených za účelem výzkumu či vývoje aplikací pro reálná použití je svázáno určitými omezeními jako je uniformní pozadí, optimální osvětlení atd., či jsou závislá na konkrétním uživateli.



Obr. 3-1 – Příklad rozpoznávání gest v laboratorních podmínkách

Rozpoznávání gest a aktivit libovolného uživatele v reálných podmínkách je v mnohém specifické. Nejedná se pouze o problém klasifikace trajektorie do několika tříd, ale je třeba využít většího množství informací získaných z obrazu, jako jsou část obrazu, v níž byla příslušná aktivita detekována, vzájemná poloha různých částí těla aktéra, a to nejenom těch, které v dané aktivitě účastní.

Jedním z problémů v rozpoznávání reálných gest může být fakt, že velká část autorů vedena cílem základní klasifikace získané trajektorie přistupuje k danému úkolu jako k popisu změny (dynamiky) jistých charakteristik – příznakového vektoru. Jak ovšem ukázali autoři práce [21] Hassink a Schopman, jistou výhodou může být přístup, v němž je daná aktivita rozdělena do částí – bloků, v nichž se předpokládá neměnný charakter sledovaných vlastností objektu – regionu zájmu. Tyto statické části nazývají *stavebními bloky gest* (anglicky *Gesture Building Blocks*, GBB) a jejich vzájemná časová následnost (či pořadí) je věcí dynamického rozpoznávání.

Autoři mnohých prací sice vykazují vysokou úspěšnost rozpoznávání i velkého množství tříd jednotlivých gest, čehož ovšem dosahují využitím informací z různých přídavných *senzorů* umístěných na těle aktéra jako jsou „inteligentní rukavice“, akcelerační čidla apod.

V problematice rozpoznávání gest v datech AMI je třeba zvážit mnohé aspekty, které v pracích jiných autorů nejsou řešeny nebo jsou řešeny způsobem, který nelze využít.

V první řadě se jedná o fakt, že pro větší přesnost je třeba zvážit možnost nalezení přesné pozice těch částí těla sledovaných osob, které vykonávají příslušnou aktivitu. Tato práce se zaměřuje na aktivity rukou, a proto se jeví jako potřebné lokalizovat část dlaně, jejíž pohyb odpovídá přesněji dynamické charakteristice dané aktivity než pohyb středu sledovaného bounding boxu, jak kupříkladu navrhl a později implementoval autor práce [22].

Získání naprosto přesné pozice sledovaného objektu je obtížně dosažitelné. Nicméně lze počítat s tím, že vhodným vyhlazením se dosáhne lepší aproximace. Jelikož žádná aktivita sledovaných osob není vždy provedena naprosto stejně, může toto vyhlazení odstranit některé nechtěné odlišnosti v časové závislosti.

Nespornou potřebou v problematice rozpoznávání gest je sledování příslušných regionů (objektů) odpovídajících částem těla aktérů v zachycených sekvencích. Různí autoři prací na téma rozpoznávání gest vždy sledovali jen jeden či dva objekty, u nichž nepředpokládali možnost, že daný objekt může ze zachycené scény dočasně zmizet, může se spojit s jiným objektem, být jím zakryt apod. Tyto dílčí problémy bude třeba zohlednit při návrhu vhodného rozpoznávací algoritmu, respektive v části sledování regionů.

Jistou výhodou klasifikace výrazných jednoznačných gest v laboratorních podmínkách – nazývejme je „umělá“ gesta – je to, že je možné jednoduše získat velké množství trénovacích příkladů. Přirozená gesta v obecných podmínkách lze získat mnohem hůře, je třeba mít k dispozici dlouhé záznamy, v nichž aktéři přirozeně vykonávají z (blíže nespecifikované) aktuální pozice příslušnou aktivitu. Nutnost získání vysokého počtu trénovacích příkladů pro správné fungování rozpoznávacího algoritmu je zmíněna v části 2.3.2 o skrytých Markovových modelech, nicméně možnosti jsou omezené množstvím dostupných dat. Jak autoři [21] uvádějí, většina složitých gest (psaní, naznačování směru, velikosti atd.) se v AMI datech vyskytuje v malé míře.

4 Návrh a implementace algoritmu

4.1 Základní popis

Při návrhu rozpoznávacího algoritmu vycházím ze své předchozí práce na téma zpracování dat z projektu AMI. Ve své bakalářské práci jsem implementoval segmentaci objektů částí lidského těla pomocí modelování barev obrazových bodů pokožky Gaussovou funkcí.

Obraz pravděpodobností příslušnosti dané barvy pixelu k modelu barvy lidské pokožky (anglicky *Skin Probability Image*, SPI) je nutné zpracovat tak, aby bylo možné analyzovat každý spojitý objekt. Toho lze dosáhnout prahováním a následným použitím algoritmu spojených komponent (anglicky *Connected Component Labelling*, CCL). Spojité objekty je možno vybírat tak, že se zanedbají (odstraní z dalšího zpracování) ty objekty, jejichž velikost (například jenom v nějakém rozměru) je příliš malá.

V kapitole 3 je diskutována nutnost nebo alespoň vhodnost nalezení přesnějších pozic těch částí rukou účastníků zachycených v nahrávkách, které jsou podstatné z hlediska analýzy dynamiky. Prakticky tedy se jedná o nalezení alespoň přibližné polohy dlaně. Jelikož lidská dlaň vykazuje větší variabilitu v jasové oblasti než ostatní části lidské ruky z důvodu stínů nebo pozadí mezi jednotlivými prsty, lze předpokládat, že konvolucí s vhodnými jádry nebo aplikací hranových operátorů spolu s prahováním bude možné tyto části lokalizovat.

Dalším krokem k rozpoznávání gest je nalezení algoritmu sledování objektů v čase – tzv. *tracking*. Část 2.2 zmiňuje metodu trackingu sledováním minimálních obalových obdélníků – bounding boxů. Řešení situací, kdy dochází ke spojení či rozdělení některých sledovaných objektů, lze provézt intuitivně podle charakteru většiny situací nacházejících se v AMI datech.

Vzhledem k tomu, že lokalizace objektu, případně jeho části, není vždy naprosto přesná, je nutné zvážit vhodnost *filtrace* detekovaných pozic v časové závislosti. Vyhlazená data by měla lépe odpovídat skutečnosti, a proto vézt k přesnějším výsledkům v dalším zpracování – rozpoznávání.

4.2 Nalezení regionů zájmu

Vytvoření modelu barvy lidské pokožky je základním krokem k další analýze. Zvolena byla metoda modelování jednou Gaussovou funkcí v YCrCb barevném modelu. Pro větší přesnost byl také implementován (podle výsledků práce [23]) výpočet hodnoty pravděpodobnosti, že daný pixel náleží do množiny obrazových bodů barvy jiné než pokožky (také v YCrCb barevném modelu).

Pomocí dvou spočtených vektorů středních hodnot a dvou kovariančních matic – μ_{skin}, $\mu_{non\text{-}skin}$ a Σ_{skin}, $\Sigma_{non\text{-}skin}$ – můžeme odvodit výslednou pravděpodobnost, že daný pixel je pixelem pokožky. Za hodnotu pravděpodobnosti nějakého obrazového bodu $\boldsymbol{v}$ lze považovat hodnotu funkce hustoty

pravděpodobnosti dané vzorcem (12), stejný vzorcem lze vypočítat pravděpodobnost, že daný pixel patří do množiny *non-skin*. Výsledná pravděpodobnost je dána vzorcem

$$p(skin \mid \vec{v}) = \frac{p(\vec{v} \mid skin)}{p(\vec{v} \mid skin) + p(\vec{v} \mid non - skin)} \tag{40}$$

odvozeným z tzv. *Bayesova teorému*.

Nalezení jednotlivých regionů odpovídajících rukám a obličeji je implementováno tak, že nejdříve je sestaven SPI, který je následně prahován. SPI je definován jako

$$\phi(i, j) = a.p(skin \mid \overrightarrow{v_{ij}}) \tag{41}$$

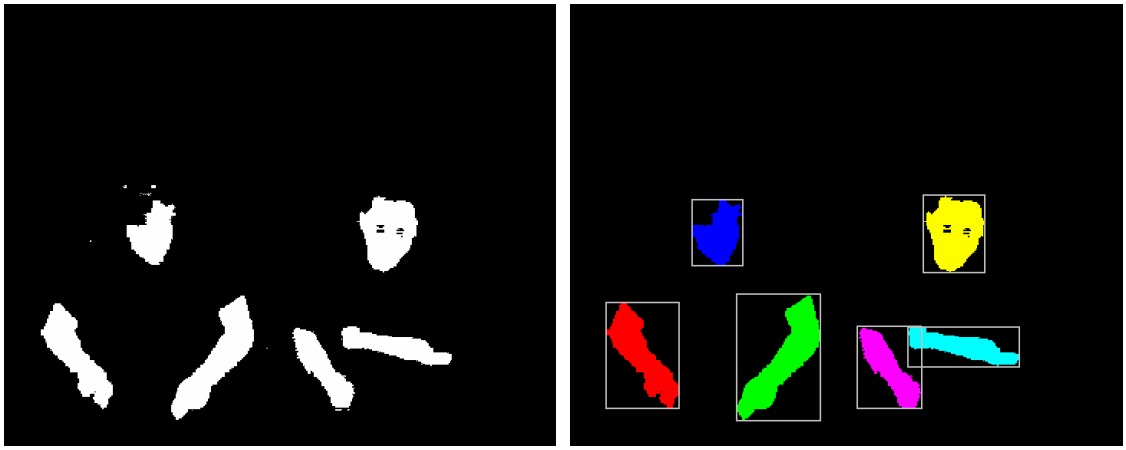
kde *a* je libovolná (tedy vhodná) konstanta. Za hodnotu prahu *T* a konstanty *a* autoři práce posledně zmíněné volí *a = 255* a *T = 60*, nicméně tyto hodnoty vedly k příliš velkému množství nežádoucího šumu, a proto volím empiricky zjištěné konstanty *a = 255* a *T = 127*.

Obr. 4-1 – Originální obraz a SPI

Dalším krokem pro analýzu regionů je nalezení spojených částí. Jeho princip spočívá v označení každého prahovaného bodu hodnoty 1 (tj. větší než práh *T*) přirozeným číslem společným pro celý region (anglicky *label*). Existuje několik algoritmů, například pomocí *semínkového vyplňování* nebo níže uvedeného algoritmu:

```
for (each pixel P from upper left corner)
      if (value of P == 1)
            if (all neighbors of P == 0) assign a new label to P;
            else if (only one neighbor of P == 1) assign its label to P;
            else if (more than one of the neighbors of P == 1)
                  assign one of the labels to P;
                  make a note of the equivalence of these labels;
```

Tento algoritmus byl implementován z důvodu své jednoduchosti a obecné použitelnosti. Poté, co jsou jednotlivé spojené regiony nalezeny, provede se odstranění těch, jejichž velikost je příliš malá. Jako hranici pro odstranění regionu volím podmínku velikosti méně než sto obrazových bodů.



Obr. 4-2 – Prahovaný SPI a nalezené spojené regiony – po odstranění těch příliš malých (každá barva odpovídá právě jednomu regionu), sedě vyznačené jsou bounding boxy

4.3 Zpracování regionů

Jak bylo uvedeno v úvodu této kapitoly, sledování středů obalových obdélníků nemusí vždy podávat přesnou informaci o trajektorii té části těla, kterou aktér danou aktivitu provedl. Zvláště markantní je tento rozdíl v případě rukou, kdy není nalezeno pouze zápěstí, ale celý loket respektive paže účastníka – viz obrázek 4-2.

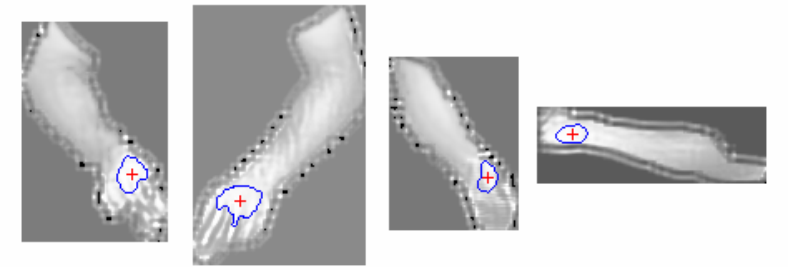
Nalezení pozice dlaně jsem implementoval pomocí *konvoluce* regionu ruky s jádry kosinové vlnky v horizontální (α_{NxN}) a vertikální (β_{NxN}) orientaci. Vzniklý konvoluční obraz Π lze zapsat následovně (Ω je původní šedotónový obraz vynásobený hodnotami SPI pro odstranění vlivu pozadí):

$$\Pi(i,j) = \sum_{n=1}^{N}\sum_{m=1}^{N}\Omega(i-\left\lfloor\frac{N}{2}\right\rfloor+n,\ j-\left\lfloor\frac{N}{2}\right\rfloor+m)\alpha(n,m) + \sum_{n=1}^{N}\sum_{m=1}^{N}\Omega(i-\left\lfloor\frac{N}{2}\right\rfloor+n,\ j-\left\lfloor\frac{N}{2}\right\rfloor+m)\beta(n,m) \quad (42)$$

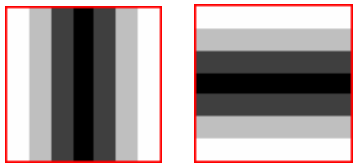
V místech velké variability v šedotónové oblasti lze předpokládat, že bude poměrně větší odezva na tyto jádra. Velikost jader je nutné zvolit tak, aby se blížily velikosti vzorů vyskytujících se v obraze v kýženém místě. V tomto případě jde o prsty rukou a stíny mezi nimi. Empiricky bylo zjištěno, že optimální odezvy se dosáhne jádry velikosti 7x7 (míněno v datech AMI).

Prahováním takto vzniklého obrazu se určí body, které mohou patřit do shluku vysokých hodnot nacházejících se v oblasti zápěstí. *Těžiště* (anglicky *gravity center*) největšího spojeného regionu vzniklého prahováním lze považovat za střed dlaně. Důležitou otázkou je volba hodnoty výše zmíněného prahu. Vyzkoušením několika hodnot prahu a zhodnocením celkového výsledku lokalizace jsem dospěl k hodnotě $T = min_{\Pi} + 0{,}9 * (max_{\Pi} - min_{\Pi})$.

Obrázek 4-3 ukazuje výsledný konvoluční obraz, červené body jsou těžiště modře vyznačených největších shluků prahovaných hodnot. Následující obrázek jsou konvoluční jádra, která byla využita.



Obr. 4-3 – Nalezení pozice dlaně konvolucí



Obr. 4-4 – Konvoluční jádra (zvětšeno 10x)

4.4 Sledování regionů

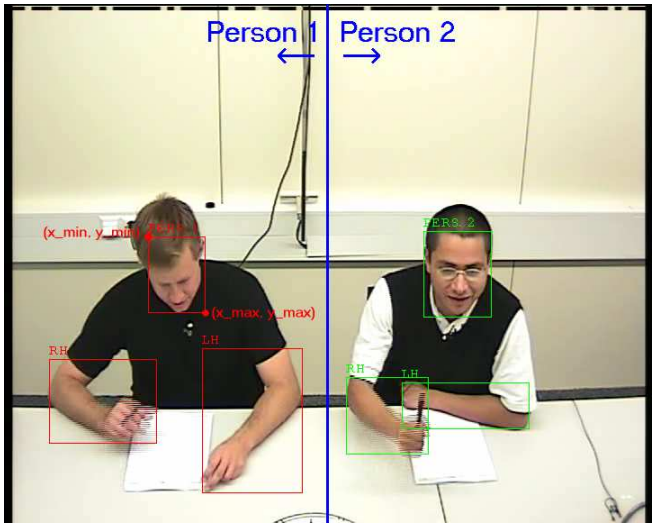
4.4.1 Výchozí předpoklady a inicializace trackingu

V rámci zjednodušení celého algoritmu lze vyjít z jistých počátečních předpokladů o pozici a póze účastníků v záznamu a také o jejich počtu a počtu detekovaných částí jejich těla. Na základě těchto předpokladů je možné určit, které nalezené regiony je nutné sledovat v čase.

Scéna zaznamená v jedné videosekvenci AMI dat zachycuje vždy právě dvě osoby z čelního pohledu. Analýzou dostupných dat bylo zjištěno, že nedochází k výskytu více než šesti regionů barvy pokožky, někdy ovšem méně vlivem zakrytí některých z nich za objekty jiné barvy než pokožky. K detekci menšího počtu regionů dochází také v tom případě, že některé regiony se spojí do jednoho.

V implementaci algoritmu sledování vycházím z předpokladu, že (jak zachycuje obrázek 4-5) jsou regiony hlavy a rukou každého účastníka rozloženy buď jen v pravé nebo jen v levé části scény. Za hlavu/tvář účastníka je považován ten z regionů nalezených v příslušné polovině obrazu, který leží nejvýše (přesněji řečeno hodnota y_{min} nejmenšího obalového obdélníka $bb_{tvář} = (x_{min}, y_{min}, x_{max}, y_{max})$ je minimální).

Dalšímu předpokladu podléhá pozice rukou v inicializační části. Regiony zbývající po nalezení tváře mohou být pouze ruce. Proto jsou za ruce považovány dva největší (co do počtu bodů) regiony nalezené v dané části. Jejich rozlišení na pravou a levou ruku je možné podle hodnoty x_{min} bounding boxů.



Obr. 4-5 – Demonstrace předpokladů pro sledování regionů

4.4.2 Algoritmus sledování

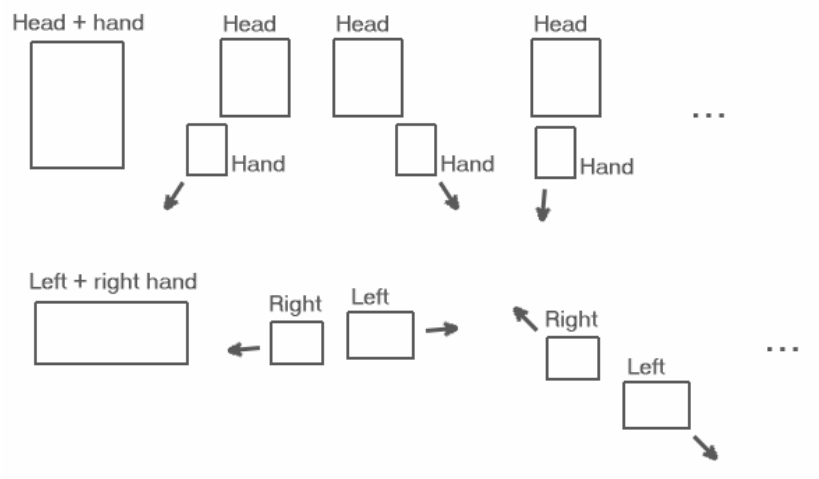
V případě, že pohyb osob a jejich částí je prost velkých rychlostí, kdy se změny pozic sledovaných regionů blíží velikostem jejich obalových obdélníků, je možné využít metodu sledování pomocí nich. Popsána byla v části 2.2.1.

Problémem k vyřešení ovšem jsou situace, kdy dochází ke spojení, rozdělení a zakrytí regionů. V dostupných AMI datech nedochází ke všem obecně možným situacím překrytí a rozpojení sledovaných regionů, proto níže uvádím základní předpoklady pro tyto situace, které byly využity při implementaci.

Předně jsou zde situace, kdy sledovaný region vymizí z pole záběru kamery. Vzhledem k tomu, že takovéto situace nastávají pouze založením rukou účastníka (účastníků) za okraj stolu či vlastní záda, je možné takovouto situaci řešit tak, že ve snímcích po vymizení regionu bude algoritmus v místě posledního jeho výskytu a v blízkém okolí kontrolovat, zda znovu nedošlo k objevení nového regionu, který je poté prohlášen za opětovný výskyt regionu chybějícího.

Velmi časté jsou také případy, kdy v nějaké sekvenci dochází ke spojování a následnému rozdělování regionů. Toto bylo vyřešeno tak, že v případě dvou blízkých regionů je sledováno, jestli v oblasti dané příslušnými dvěma regiony v minulém snímku se nachází jeden či více regionů ve snímku aktuálním. Pokud se zde vyskytuje pouze jeden, došlo ke spojení těchto dvou oblastí.

Následné rozdělení regionů je opět nutné řešit zvláštním přístupem. Jako postačující pro většinu situací vyskytujících se v datech AMI se ukázalo tyto případy řešit tak, že oddělivší se region je považován za ten z těch, které patřily v minulých snímcích do společné oblasti, který vůči druhému zaujímá pozici člověka v klidové poloze (hlava nejvýše, levá ruka nalevo, pravá napravo). Náčrt 4-6 výše popsané situace zobrazuje.



Obr. 4-6 – Vybrané situace opětovného dělení regionů

Situací rozdělení v předchozích snímcích splynutých regionů je samozřejmě velké množství, vždy však ve snímcích, kde se tyto regiony dělí, předpokládám platnost rovnic (43) a (44). Uvedené rovnice jsou pro počátek souřadnic v levém horním rohu, v případě jiného počátku je třeba je adekvátně upravit.

$$y_{\min}^{bb\_head} < y_{\min}^{bb\_hand} \tag{43}$$

$$x_{\min}^{bb\_right} < x_{\min}^{bb\_left} \tag{44}$$

4.5 Úprava dynamických dat

Vzhledem k tomu, že lokalizace pozice objektů není téměř nikdy naprosto přesná, je třeba provézt vyhlazení dat tak, aby odchylka byla minimální. Jedním z vhodných aparátů vyhlazení dat je tzv. *exponenciální filtr*.

4.5.1 Exponenciální filtrování

Tento filtr existuje v několika modifikacích. Základní je ve své podstatě pouze váženým průměrem vyhlazeného (již průměrovaného) minulého vzorku a vzorku současného. Jeho nevýhodou je dlouhá *doba relaxace* – ustálení – při změně dat, která na nějakém intervalu vykazuje určitý *trend*.

Rozšířením exponenciálního filtru pro data vykazující trend je tzv. *dvojitý* exponenciální filtr. Vyhlazená hodnota vzorku je dána váženým průměrem současného (nevyhlazeného) vzorku a součtu minulého (vyhlazeného) vzorku s trendem v minulém vzorku.

Označme S_t vyhlazenou (anglicky *smoothed*) hodnotu vzorku v čase t, nevyhlazenou hodnotu v témž čase r_t a trend změny dat b_t v čase t. Způsob výpočtu je následující:

$$S_t = \alpha r_t + (1-\alpha)(S_{t-1} + b_{t-1}) \tag{45}$$

$$b_t = \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1} \tag{46}$$

Pro stanovení hodnot konstant α a β neexistuje žádný analytický nástroj, je nutné je vybrat empiricky, případně provézt analýzu přesnosti vyhlazení pro jejich různé hodnoty.

Stejně tak neexistuje jediný správný přístup k inicializaci hodnoty trendu na začátku série vzorků. Některé často používané výpočty jsou uvedeny v rovnicích (47) až (49).

$$b_1 = r_2 - r_1 \tag{47}$$

$$b_1 = \frac{(r_2 - r_1) + (r_3 - r_2) + (r_4 - r_3)}{3} \tag{48}$$

$$b_1 = \frac{r_n - r_1}{n - 1} \tag{49}$$

Podrobné informace o jednoduchém, dvojitém i trojitém exponenciálním filtru je možné nalézt na stránkách [24].

4.5.2 Využití v rozpoznávacím algoritmu

Vzhledem k tomu, že algoritmus popsaný v části 4.3 vykazuje jistou míru nepřesnosti, je vhodné využít exponenciálního filtrování pro pozice nalezené výše zmíněným algoritmem. Tím by mělo ve většině případů dojít k ustálení pozice v klidových fázích, tj. těch, kdy ruka účastníka nevykonává žádnou aktivitu, a vyhlazení průběhu pozice dlaně v těch částech, kde je nějaká aktivita konána.

Exponenciální filtr je původně určen k vyhlazení diskrétních jednorozměrných veličin, a proto bude muset být upraven do tvaru pro dvousložkové vektory pozice. Skaláry S_t a b_t jsou nahrazeny vektory $\boldsymbol{S_t}$ a $\boldsymbol{b_t}$, operace součtu, rozdílu a násobení skalárem jsou dány obecnými vlastnostmi vektorů.

4.5.3 Stanovení hodnot konstant α a β

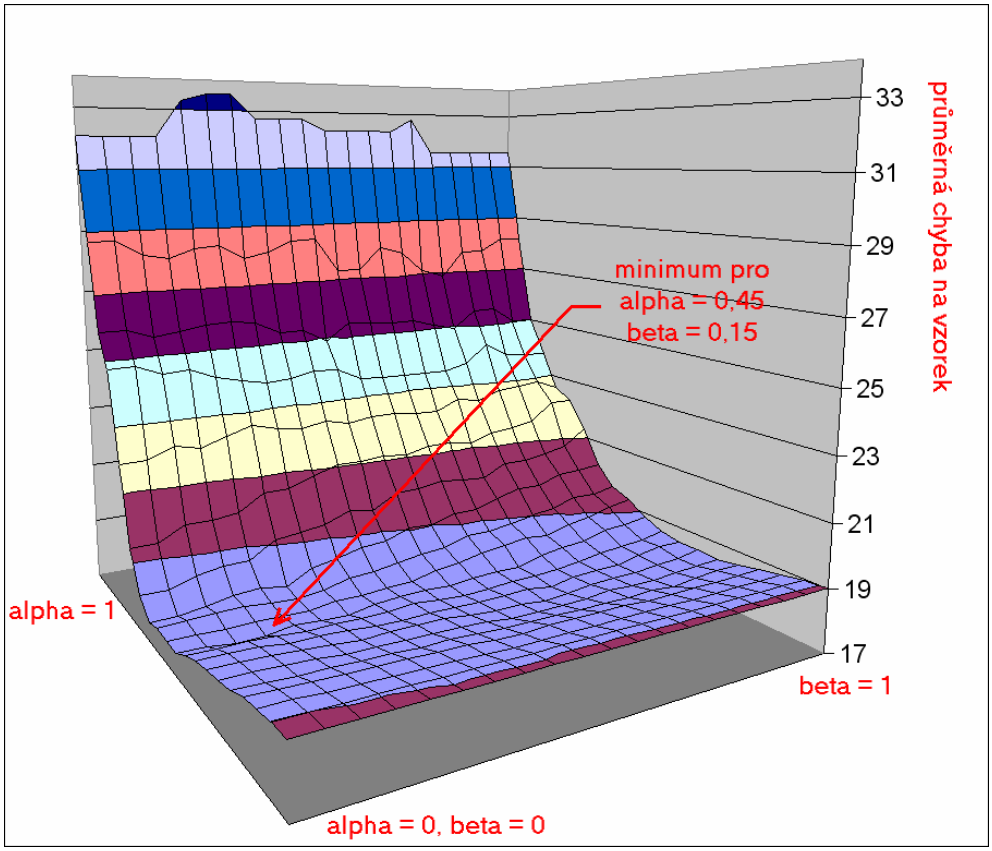
Pro správné fungování filtru je nutné vhodně zvolit (nalézt) hodnoty konstant obsažených ve vzorcích (45) a (46). Tyto konstanty určují rychlost změny vyhlazených hodnot a rychlost změny trendu ve vzorcích.

Jedním ze způsobů nalezení α a β je anotace vhodné – tj. reprezentativní – sekvence a poté měření celkové chyby (pro různé volby konstant) vyhlazené trajektorie s trajektorií anotovanou. Vybrána byla sekvence asi sedmdesáti snímků zachycující nejdříve klidovou fázi ruky (přibližně polovina snímků) a následně opakovaný pohyb ruky ve vertikálním směru.

Metrika chyby byla zvolena průměrná *Euklidova vzdálenost* jedné pozice v anotované trajektorii $\boldsymbol{p_a}$ a odpovídající pozice v trajektorii $\boldsymbol{p_v}$ vyhlazené exponenciálním filtrem s konstantami α a β, N je počet vzorků trajektorií:

$$Err_{\alpha,\beta} = \frac{\sum_{i=1}^{N} \left| \overrightarrow{p_a(i)} - \overrightarrow{p_v(i)} \right|}{N} \tag{50}$$

Měření jsem provedl pro hodnoty $\alpha, \beta = 0{,}0 - 1{,}0$ s krokem $0{,}05$. Výsledek je zobrazen v grafu na obrázku 4-7. Výsledné hodnoty koeficientů filtru, který nejlépe vyhladil analyzovanou sekvenci, jsou $\alpha = 0{,}45$ a $\beta = 0{,}15$.



Obr. 4-7 – Výsledky měření rozdílu anotované trajektorie a trajektorie vyhlazené exponenciálním filtrem pro různé hodnoty parametrů filtru

Výsledky vzešlé z výše popsaného experimentu budu používat pro vyhlazování všech pozic dlaně v sekvencích exponenciálním filtrem.

4.6 Rozpoznávání

V rozvaze v kapitole 3 předcházející tuto část bylo uvedeno, že reálná gesta nejsou pouze přesně vymezenou trajektorií, již je třeba za účelem rozpoznání posoudit, ale celkový souhrn charakteristik pozic a pohybu různých částí těla účastníků. Autoři Hassink a Schopman v [21] volí přístup shlukování příznakových vektorů, na základě kterého dělí danou aktivitu na bloky typické malou variabilitou charakteristik. Tento postup, pomocí nějž lze modelovat různé části gest, jsem zvolil také.

4.6.1 Shluková analýza

Jedním ze silných nástrojů k *analýze shluků* je modelování pomocí *směsi Gaussových funkcí* (anglicky *Gaussian Mixture Model*, GMM) v daném *d*-rozměrném prostoru. Velkou výhodou takovéhoto modelování je zvláště fakt, že pro daný neznámý vektor $\boldsymbol{v}$ a natrénovaný model M je možné určit věrohodnost (likelihood), že $\boldsymbol{v}$ patří do modelu M.

Proces trénování je vlastně maximalizováním věrohodnosti, že trénovací data patří do trénovaného modelu.

Trénování modelu GMM

Nejčastěji používaný (a v této práci implementovaný) algoritmus trénování parametrů modelu GMM je obdoba tzv. EM algoritmu (anglicky *Expectation-Maximization Algorithm*). Spočívá v opakování dvou kroků, z nichž první (E-krok) je přiřazením vektorů k určitému shluku (*Gaussově normálnímu rozložení*) a druhý (M-krok) aktualizuje parametry modelu (Gaussových rozložení) podle vektorů k daným shlukům přiřazeným.

Pro určení počátečního přiřazení vektorů ke shlukům je možno využít několik přístupů (např. náhodné přiřazení), implementovaný algoritmus z důvodu rychlé konvergence a dosažení přesnosti využívá pro inicializaci algoritmu *k-means*, jak je popsáno v [25] a také zřejmo z uvedeného pseudokódu trénování GMM.

Model směsí Gaussových funkcí je dán parametry n vektorů středních hodnot μ_i a n kovariančními maticemi Σ_i:

$$\forall i = 1,2,..n$$

$$\mu_i \in \Re^d \tag{51}$$

$$\Sigma_i \in \Re^{dxd} \tag{52}$$

Algoritmus trénování lze popsat kroky inicializace, opakování E-kroku a M-kroku a vyhodnocování ukončovací podmínky po provedení každé iterace E (přiřazení vektorů) a M (aktualizace parametrů) fáze:

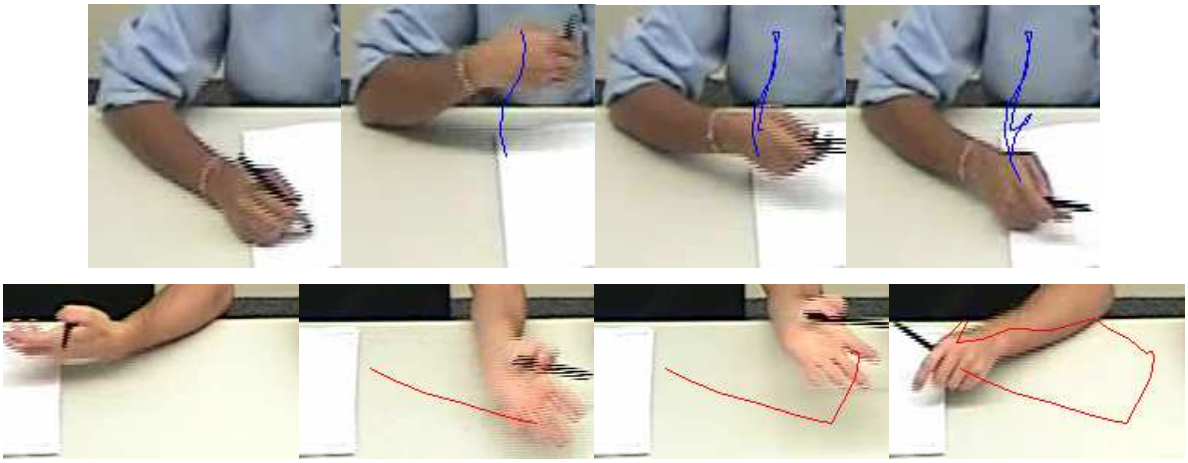
```
calculate initial means μ1 to μn using k-means algorithm;
initialize covariance matrices to identity Σi = Id;
E_step:
        assign the m training samples to the n clusters Γi using the minimum
        Mahalanobis distance rule:
                i = argmini [log(det(Σi)) + (x – μi)T (Σi)-1 (x – μi)];
M_step:
        compute new means and new covariance estimates:
                μi = (∑x in Γi x) / |Γi|
                Σi = (∑x in Γi (x - μi)(x - μi)T) / |Γi|
        where |Γi| denotes the number of vectors assigned to cluster Γi in the
        E_step;
if (changes in the means and covariances in the M_step are small enough)
        stop;
else
        goto E_step;
```

Volba složek příznakového vektoru a tříd trajektorií

Existuje více možností, jaké příznakové vektory zvolit. Základní variantou je volba vektoru ***v***, který obsahuje složky rychlostí v jednotlivých směrech souřadného systému v daném okamžiku t. Tato možnost sice nezohledňuje další aspekty (charakteristiky), které by mohly zvýšit přesnost modelování daných aktivit, nicméně je dostatečně robustní pro zachycení orientace a směru aktivity.

Základními třídami aktivit rukou účastníků v datech AMI – s ohledem na rozpoznávání gest jako jsou řeč podporující gesta (anglicky *Speech Supporting Gestures*, SSG), psaní (pořizování poznámek) aj. – jsou jejich *vertikální* nebo *horizontální* charakter. Vzhledem k tomu, že tyto trajektorie vykazují velkou variabilitu a potenciálních trénovacích příkladů pro jednotlivé případy existuje malé množství, je nutné vyjít z obecnějších předpokladů.

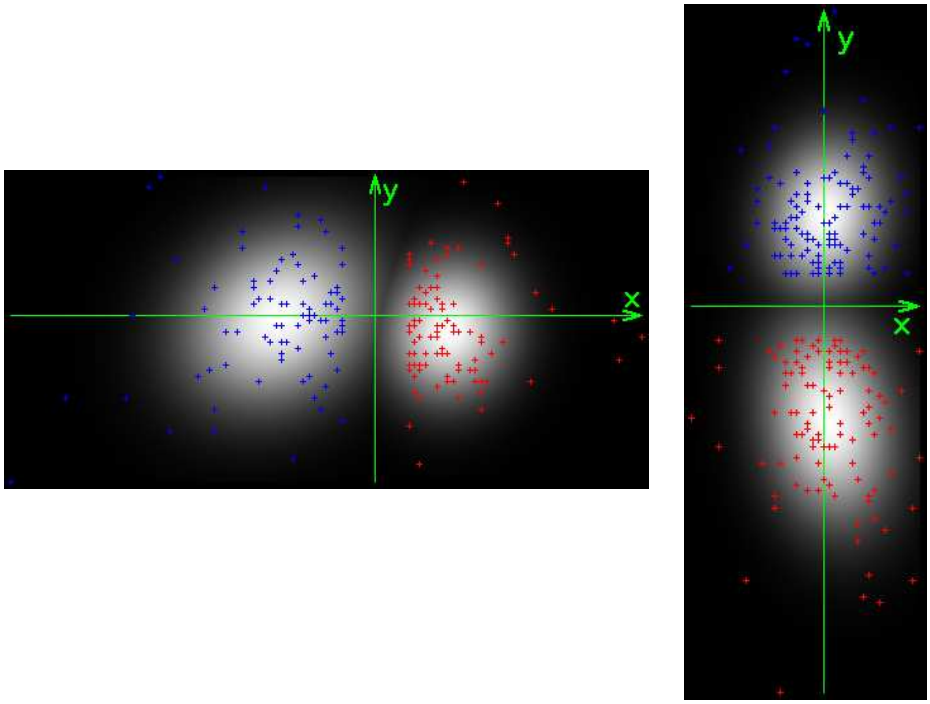
Gesta podporující řeč jsou charakteristická přímou trajektorií pohybu dlaně aktéra a také svou periodičností. Obrázek 4-8 uvádí příklad.



Obr. 4-8 – Příklady řeč podporujících gest (různé barvy trajektorie značí vertikální respektive horizontální orientaci gesta)

Mějme dvě třídy potenciálních gest: gesta vedená horizontálně a vertikálně. Lze předpokládat, že existují modely (směsi Gaussových funkcí) M_V a M_H, které je v prostoru vektorů $\boldsymbol{v}$ o složkách v_x a v_y (tedy rychlosti v jednotlivých souřadných směrech) charakterizují. Natrénování modelů na vhodných příkladech lze provézt dle algoritmu zmíněného výše.

Předpokládejme dále, že každý model je směsí dvou Gaussových funkcí. Natrénované modely demonstruje obrázek 4-9. Červeně a modře jsou vyznačeny vzorky patřící k jednotlivým shlukům, jejichž hodnoty funkce rozložení pravděpodobnosti jsou zobrazeny stupni šedé.



Obr. 4-9 – Modely horizontálních a vertikálních gest

4.6.2 Segmentace a určení třídy podsekvence

Spojitou trajektorii rukou je třeba rozdělit na ty části, kde dochází k nějaké aktivitě. Autoři práce [21] k tomu využili metodu prahování velikosti rychlosti regionu (anglicky *Activity Measure*, AM). Pro potřeby této práce navrhuji modifikaci jejich přístupu tak, že za hranici nějaké aktivity bude považován vektor v čase *t*, pro který platí

$$\left|\overrightarrow{v_{t-1}}\right| < thr_1 \tag{53}$$

$$\min_{i=0..N}\left(\left|\overrightarrow{v_{t+i}}\right|\right) > thr_1 \tag{54}$$

pro určení $\boldsymbol{v_t}$ za počátek aktivity a

$$\left|\overrightarrow{v_{t-1}}\right| > thr_2 \tag{55}$$

$$\min_{i=0..M}\left(\left|\overrightarrow{v_{t+i}}\right|\right) < thr_2 \tag{56}$$

pro $\boldsymbol{v_t}$ konec aktivity. Operace |...| značí výpočet Euklidovy vzdálenosti. Hodnota *N* je délka okna, která udává minimální délku doby, po kterou musí hodnota rychlosti regionu na začátku aktivity přesáhnout práh thr_1. Hodnota *M* potom minimální délka doby, po kterou musí aktivita klesnout pod práh thr_2, aby byl detekován konec aktivity. Zřejmě jsou tedy hodnoty *N-1* a *M-1* délky podsekvencí, po které může aktivita vzrůst nad práh, aby aktivita nebyla detekována, respektive klesnout pod práh, aby nedošlo k jejímu rozdělení.

Ty části trajektorií, které byly získány výše zmíněným způsobem, lze předpokládat, že náleží do nějaké z tříd modelovaných příslušnou směsí Gaussových funkcí – modelem GMM.

Pravděpodobnost, že nějaký vektor ***v*** patří k modelu *M* lze nahradit hodnotou funkce rozložení pravděpodobnosti pro tento vektor, tedy hodnotou věrohodnosti. Věrohodnost, že daná sekvence ***V*** = ***(v₁, v₂, ... vₙ)*** patří k modelu *M* lze spočítat součtem záporných logaritmických věrohodností jednotlivých prvků sekvence:

$$p(\vec{V} \mid M) = \frac{1}{n}\sum_{i=1}^{n} -\log\left(\frac{1}{(2\pi)^{d/2}\left|\Sigma\right|^{1/2}} e^{-\frac{1}{2}(\vec{v_i}-\mu)^T \Sigma^{-1}(\vec{v_i}-\mu)}\right) \tag{57}$$

kde μ a Σ jsou parametry modelu *M* a *d* je dimenze vektoru ***v***.

„Nejvěrohodnější“ model *M‘* sekvence ***V*** je ten, pro který platí, že průměrná záporná logaritmická věrohodnost je minimální:

$$M' = \arg\min_M (p(\vec{V} \mid M)) \tag{58}$$

Jak bylo řečeno v části předcházející (tj. 4.6.1), gesta podporující řeč jsou charakteristická svojí *periodičností*. Ostatně tato vlastnost je odlišuje od pouhých změn pozice ruky, které samozřejmě nejsou gesty. Proto je nutné nalezené podsekvence, jimž již byl přiřazen buď horizontální nebo vertikální model, $\boldsymbol{V_H}^{T1,\,T2}$ a $\boldsymbol{V_V}^{T1,\,T2}$ dále podrobit analýze periodičnosti.

Počet period pro danou podsekvenci $\boldsymbol{V_X}^{T1,\,T2} = (\boldsymbol{v}_{T1}, \boldsymbol{v}_{T1+1}, \ldots \boldsymbol{v}_{T2})$ definuji následovně. Vektor $\boldsymbol{W_V} = (w_{T1}, w_{T1+1}, \ldots w_{T2})$ je vektorem hodnot z množiny *{1, 2, ... N}*, kde *N* je počet shluků v modelu M_X. Pro všechna $i = 0 .. (T_2 - T_1)$ platí, že

$$w_{T_1+i} = \arg\min_{j=1}^{N}(D(\mu_j^X, \Sigma_j^X, \overrightarrow{v_{T_1+i}})) \tag{59}$$

$$D(\mu, \Sigma, \vec{x}) = \sqrt{(\vec{x}-\mu)^T \Sigma^{-1}(\vec{x}-\mu)} \tag{60}$$

kde $D(\mu, \Sigma, \boldsymbol{v})$ je tzv. *Mahalanobisova vzdálenost* vektoru ***v*** od množiny bodů, jejíž střední hodnota je μ a kovarianční matice Σ; μ_j^X a Σ_j^X jsou střední hodnota respektive kovarianční matice *j*-tého shluku modelu M_X .

Periodičnost je potom dána počtem souvislých úseků w_i až w_j, pro něž platí

$$i \geq T_1, j \leq T_2, i < j, j - i \geq C_T \tag{61}$$

$$\forall k = i..(j-1): w_k = w_{k+1} \tag{62}$$

$$w_{i-1} \neq w_i \tag{63}$$

$$w_{j+1} \neq w_j \tag{64}$$

Poslední podmínka v (61) definuje konstantu C_T jako minimální délku takovéhoto úseku. Neformálně lze tedy periodičnost popsat jako počet částí trajektorie, ve kterých má pohyb stejnou orientaci (vektory ***v*** přísluší ke stejné distribuci směsového modelu M_X).

Volba konstant segmentace a klasifikace

Výše zmíněný popis metody segmentace a klasifikace zavádí několik konstant, jimž je třeba pro vlastní implementaci přiřadit nějaké (vhodné) hodnoty. Volbu provádím empiricky a to tak, že pro konstanty v rovnicích (53) až (56) volím $thr_1 = 5$, $thr_2 = 7$, $N = 3$ a $M = 3$.

Hodnotu minimální délky souvislého úseku periody gesta definovanou v (61) navrhuji $C_T = 5$.

Poslední konstantou, kterou je třeba zvážit, je počet period spočtených podle výše uvedeného schématu, aby daná aktivita byla řeč podporujícím gestem. Je zřejmé, že k rozlišení periodických gest od pouhého pohybu rukou jedním směrem by mělo stačit, aby každá trajektorie s periodicitou větší než 1 byla považována za detekované gesto. Samozřejmě mohou existovat takové trajektorie, které nejsou gesty, ale vykazují periodicitu větší než jedna. Oproti tomu prahování hodnotou větší než 1 může vést k tomu, že některá gesta nebudou rozpoznána. Tento parametr je třeba zvolit s ohledem na oba výše zmíněné aspekty. V části 5.3 této práce jsou demonstrovány výsledky a celkové shrnutí je začleněno do závěru (kapitola 6).

5 Popis funkce programů a algoritmu

V rámci implementace algoritmu nastíněného v minulé kapitole byly vytvořeny tři programy plnící určitou část funkcí rozpoznávání. Popsány jsou v následujících podkapitolách a také v přílohách 2 a 3.

5.1 Implementační prostředí

Všechny programy byly vytvořeny v jazyce *C++* ve vývojovém prostředí *MS Visual Studio 2005*, které je dostupné pro studenty FIT v Brně z *MSDN Academy* firmy *Microsoft*. Zvolený jazyk i prostředí jsou v dnešní době standardně používané nástroje, přenositelnost kódu programů v C++ je ve velké míře možná.

Při práci na programech bylo využito některých funkcí z volně šířitelné knihovny pro oblast počítačového vidění *OpenCV* dostupné ze zdroje [26].

5.2 Vytvořené programy

5.2.1 Program DP_track

Tento program je dávkovou aplikací, která zadanou videosekvenci ve formátu *AVI* analyzuje podle uživatelem zadaných parametrů, což zahrnuje funkce popsané v částech 4.2 až 4.5 včetně, tedy fáze

- segmentace snímku v YCrCb modelu na základě modelování barvy Gaussovou funkcí,
- zpracování regionů – nalezení pozice dlaně ruky,
- sledování regionů a
- vyhlazení dynamických hodnot

Výstupem programu mohou být buď trajektorie v textovém formátu, nebo snímky ze vstupní AVI sekvence s vyznačenými obalovými obdélníky.

5.2.2 Program param_fit

Z důvodu přehlednosti vznikla aplikace *param_fit*, jež by tématicky samozřejmě mohla být začleněna do předchozího programu. Jedná se o relativně jednoduchý program, jehož jedinou funkcí je výpočet chyby filtrování (vyhlazení) dynamických hodnot dvojitým exponenciálním filtrem. Pomocí tohoto programu byly určeny parametry diskutované v části 4.5.3 a také vytvořen graf na obrázku 4-7.

Vstupem jsou dva textové soubory obsahující anotovanou trajektorii a trajektorii nevyhlazenou. Výstupem je tabulka průměrných chyb na vzorek pro parametry α a β od *0,0* do *1,0* s krokem *0,05*. Výstupní tabulku je možné jednoduše upravit ve zdrojovém kódu.

5.2.3 Program DP_gmms

Program *DP_gmms* implementuje shlukování pomocí GMM, tj. nejdříve natrénování modelů, poté derivaci trajektorie dané pozicemi (x_i, y_i) sledovaných objektů, následnou segmentaci podsekvencí a jejich klasifikaci.

Vstupem jsou jednak příklady pro trénování modelů a také analyzovaná sekvence; výstupem snímky ze vstupní AVI sekvence s vyznačenými nalezenými aktivitami/gesty.

Přesný popis parametrů, vstupních a výstupních dat a jejich formátů je obsažen v příloze č. 2. Náhled do jejich struktury (podstatné funkce a třídy aj.) dává příloha č. 3.

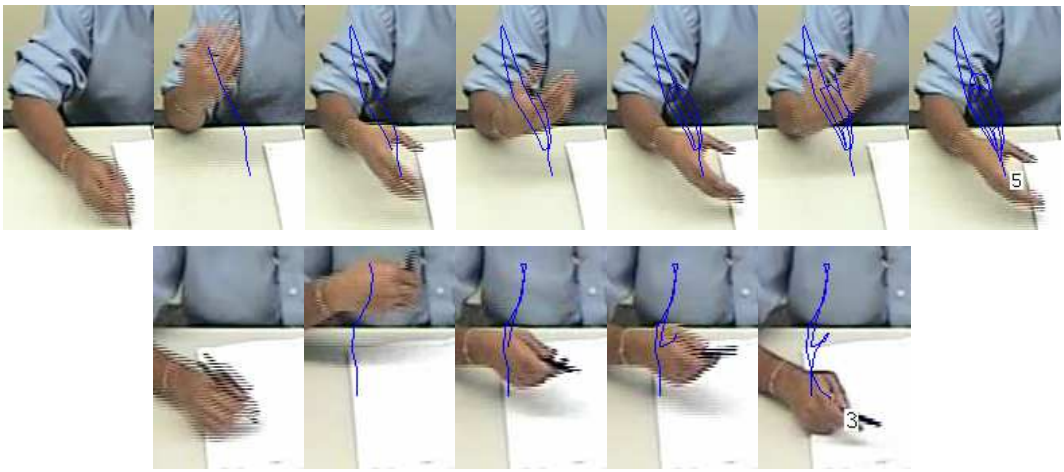
5.3 Demonstrace na příkladech

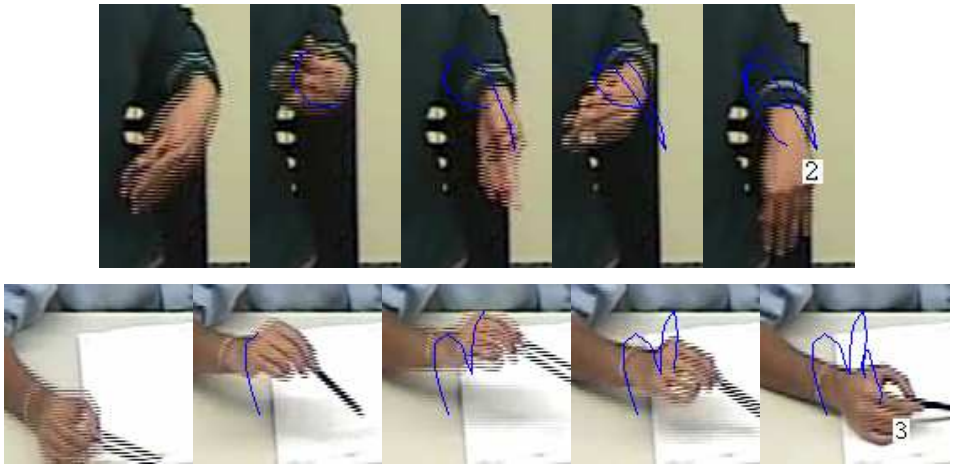
V této části bych rád ukázal principy fungování algoritmu popsaného v kapitole 4. Pro přesnost předesílám, že ve všech níže uvedených obrázcích jsou trajektorie klasifikované jako vertikální vyznačeny modře, červeně potom trajektorie horizontální. Čísla vepsaná na konci každé trajektorie příslušné detekované aktivity (gesta) jsou počty period, tak jak je definováno v části 4.6.2.

Budeme-li pokládat za řeč podporující gesto takovou aktivitu, která vykazuje periodicitu 2 a vyšší, lze rozdělit detekované aktivity do následujících kategorií podle úspěšnosti.

5.3.1 Úspěšná klasifikace gesta

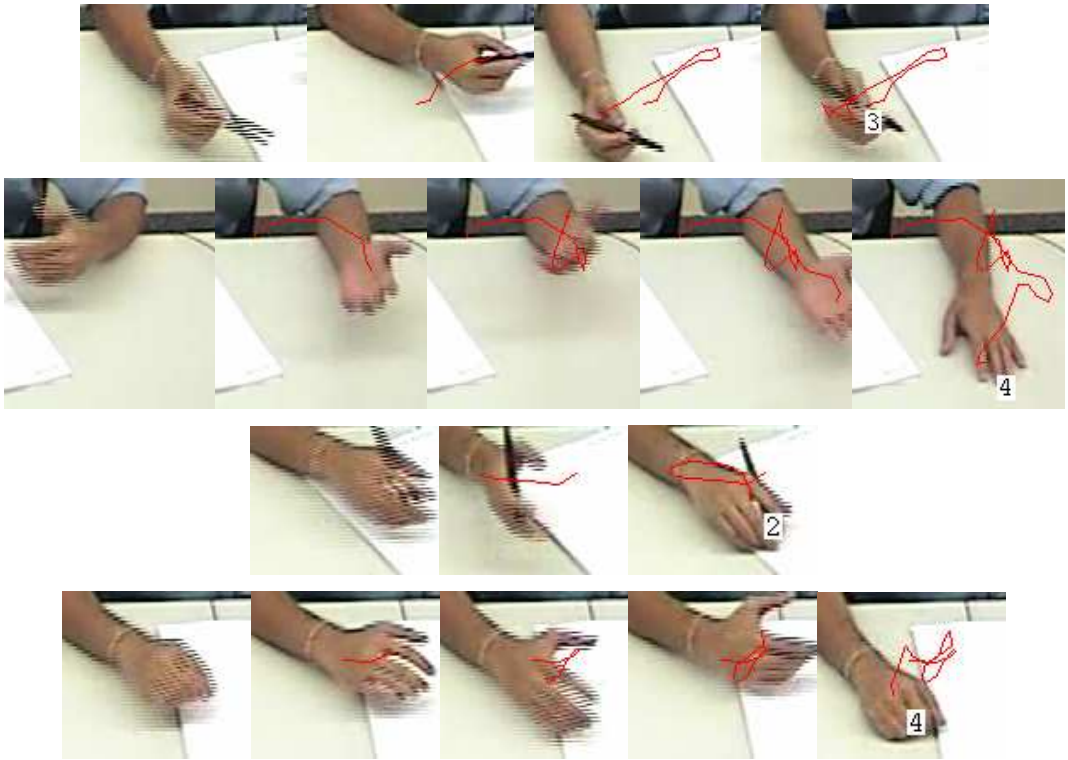
Příklady gest, která byla úspěšně rozpoznána, jsou na obrázku 5-1 a 5-2. První jsou gesta, jejichž trajektorie byla přiřazena modelu vertikálnímu M_V.





Obr. 5-1 – Gesta klasifikovaná vertikálně

Další skupinou jsou gesta horizontálního charakteru. Pozornosti doporučuji výraznost jednotlivých příkladů, která je dále diskutována.

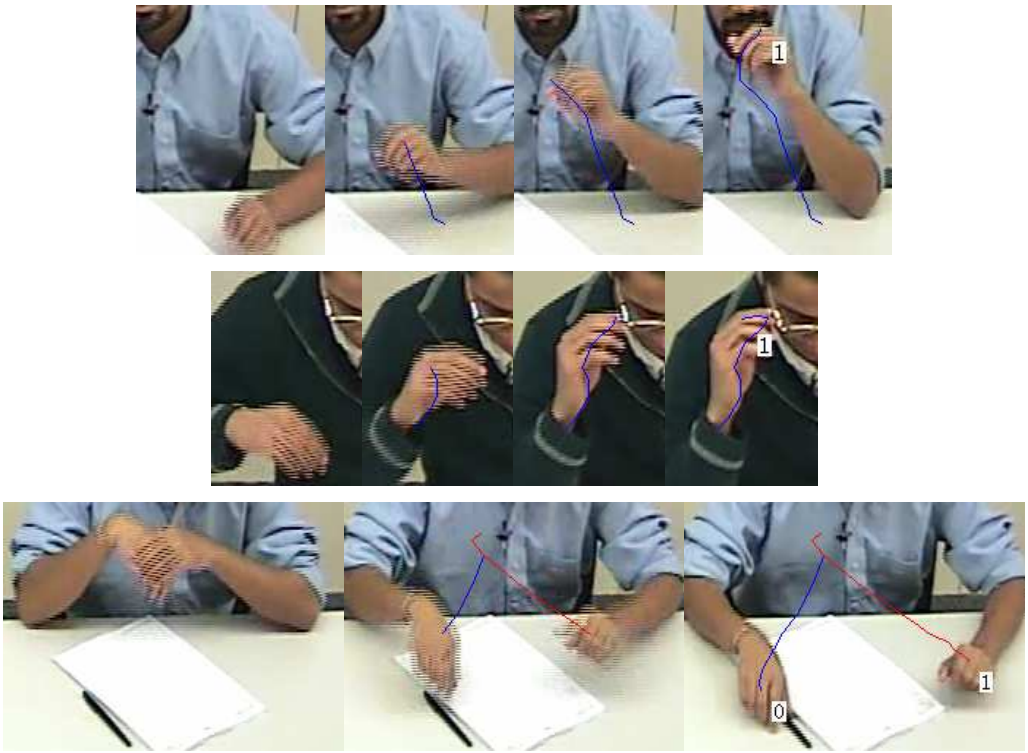


Obr. 5-2 – Gesta klasifikovaná horizontálně

První dva příklady v obrázku 5-2 jsou gesta co do amplitudy výrazná. U takovýchto gest je mnohem menší pravděpodobnost, že nebudou vysegmentována např. proto, že nebudou splněny podmínky (53) – (56). U gest nevýrazných (další dva příklady) tato pravděpodobnost roste. Dále však roste i nebezpečí, že tak, jak je navrženo v části 4.6.2, nebudou všechny části intuitivně považované za periody (souvislé části trajektorií) nalezeny. Tento fakt ovlivňuje volba konstanty C_T (61). Poslední příklad ukazuje gesto nevýrazné amplitudy, kdy byly nalezeny všechny čtyři takovéto části.

5.3.2 Jiná správně klasifikovaná aktivita

Jak již bylo řečeno, základním předpokladem, že nějaká aktivita je gestem, je určitá míra periodicity. Trajektorie, které by neměly vykazovat periodicitu, jsou různé jednorázové přesuny rukou z jedné klidové polohy do druhé. Obrázek 5-3 ukazuje takovéto správně klasifikované aktivity.



Obr. 5-3 – Klasifikovaná neperiodická aktivita

5.3.3 Neúspěšná klasifikace

Do této třídy můžeme zařadit jednak případy, kdy dané gesto není vůbec nalezeno – tedy došlo k chybě segmentace (obr. 5-4). Dále případy, kdy je trajektorie nějakého gesta chybnou segmentací rozdělena do více nezávislých aktivit, které poté nejsou „shledány“ periodickými (obr. 5-5) a také případy, kdy nějaká aktivita (např. zvednutí nebo posun ruky) je klasifikována jako gesto (2 a více period, obr. 5-6).

Obr. 5-4 – Nedetekovaná aktivita

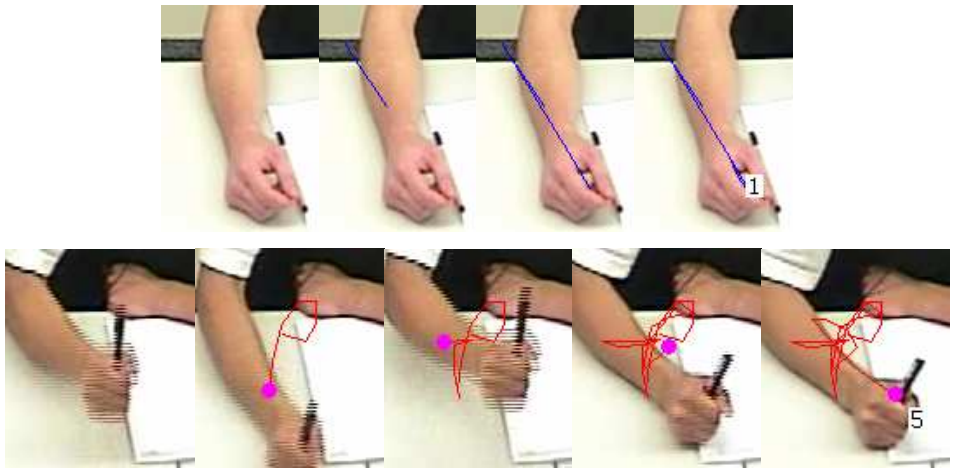
Obr. 5-5 – Rozdělená trajektorie

Obr. 5-6 – Chybně detekovaná periodicita

5.3.4 Další aspekty

Na neúspěchu rozpoznávání aktivit mohou samozřejmě mít podíl i výsledky předchozích kroků než jen vlastní rozpoznání. Důležitým aspektem je jistě přesnost a robustnost detekce dlaně účastníka. Například trajektorie na obrázku 5-6 je na počátku a konci pohybu zatížena „šumem“, tj. nepřesnou lokalizací, která může být příčinou špatného rozpoznání.

Obrázek 5-7 ukazuje dva případy: v prvním byla vlivem špatné lokalizace (úskoku detekce dlaně a zvětšení jeho amplitudy vlivem exponenciálního filtrování, které po nějakou dobu danou parametry udržuje směr pohybu) k detekci aktivity. V druhém případě došlo sice k rozpoznání gesta, ale daná trajektorie neodpovídá trajektorii skutečné.



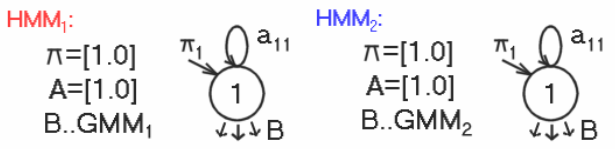
Obr. 5-7 – Chybná lokalizace dlaně (v druhém příkladu vyznačuje fialový bod aktuální detekovanou pozici dlaně)

6 Závěr

Tato práce ve své první části shrnuje metody, které využili různí autoři v dílech zabývajících se (převážně) rozpoznáváním gest ve videosekvencích pomocí analýzy obrazu. Výčet přístupů není samozřejmě úplný, nicméně dává relevantní vhled do dané problematiky, tak jak bylo uvedeno v prvním bodě zadání. Diskuze různých úskalí, výhod a nevýhod postupů navržených jmenovanými autory vytváří podklady pro návrh a implementaci vlastního přístupu – algoritmu – v druhé části práce.

Vytvořený algoritmus na základě natrénovaných modelů směsí Gaussových funkcí vytváří pravděpodobnostní rámec pro nalezení gest podporujících řeč na základě jejich orientace a analýzy počtu nezávislých částí trajektorie označovaného jako periodicita a jejich oddělení od takových trajektorií, které nemají periodický charakter (jsou pouze pohybem rukou z jedné klidové polohy do druhé).

Výše zmíněné Gaussovy modely klasifikují nalezená místa aktivity rukou do dvou tříd (horizontální a vertikální) podle pravidla o průměrné záporné logaritmické pravděpodobnosti, stejně jako by tomu mohlo být při využití skrytých Markovových modelů s jedním stavem a maticí pravděpodobností vyslání (spojitých) symbolů danou rozložením pravděpodobností příslušného Gaussova modelu.



Obr. 6-1 – Analogie navrženého algoritmu

Následným krokem klasifikace je určení periodicity jako počtu souvislých úseků trajektorie, které byly generovány jedním shlukem příslušného Gaussova modelu.

Poslední část této práce popisuje vytvořené programy implementující navržený algoritmus a demonstruje fungování algoritmu na příkladech, stejně jako zmiňuje případy selhání algoritmu z různých příčin.

Práce tedy řeší všechny body uvedené v zadání, a proto z pohledu obsahu jej plní celé. S ohledem na cíl daný v úvodu, tedy nalezení takových přístupů k získávání informací z videodat, které rozšíří prakticky využitelnou množinu informací o nich, lze říci, že gesta podporující řeč účastníka jsou jistě vhodným rozšířením; mohou například podat informaci o místech v promluvě, jimž autor přikládá větší váhu nebo v nich naopak naznačuje nejistotu v podávaných skutečnostech.

Na výsledky navrženého algoritmu je možné nahlížet z více úhlů. Z kvantitativního pohledu lze říci, že sice byla pozornost věnována pouze jedné třídě gest, nicméně navržený přístup po alespoň zběžném uvážení stejně jako uvážení všech s ním souvisejících přípravných kroků a jejich analýze – přičemž nutnost takového zkoumání a její míra se ponechává k dalším úvahám – může poskytnou

podklad pro rozsáhlejší práci v oblasti rozpoznávání gest. „Znovuvyužitelnost“ algoritmu navrženého v této práci a jeho případné provázání s jinými přístupy či začlenění do globálnějšího systému rozpoznávání gest jsou samozřejmě možné. Dále je třeba říci, že vlastní rozpoznání řeč podporujících gest není z pohledu této práce pouhou informací typu *ano – ne*, ale že také dává další informace o celkové orientaci rozpoznaného gesta. Praktická použitelnost takovéto informace je však problémem spíše filozoficko-sociologickým: tedy posouzení přesného významu různě vedených gest.

Pro získání přesného vyjádření kvalitativních vlastností výsledků navrženého algoritmu by bylo třeba provézt srovnávací testy jeho výsledků s již anotovanými sekvencemi. Z důvodu jejich nedostupnosti a také časových omezení této práce nebyly zatím provedeny.

Je tedy zřejmé, že se zde otvírá široké pole působnosti pro další rozšíření této práce. Předně je to zvážení vhodnosti provedení analýzy přesnosti rozpoznání pomocí popsané metody, případně její provedení po vhodném rozšíření algoritmu. Dále určitě zvážení možností zpřesnění výsledků rozšířením příznakového vektoru na vstupu shlukové analýzy například o úhel natočení ruky (zápěstí) nebo složky zrychlení v jednotlivých souřadných směrech. Je třeba ovšem zmínit, že volba dalších složek příznakového vektoru nemusí vždy vézt k větší přesnosti, naopak někdy může způsobit její pokles. Proto by tato možnost měla být důkladně přezkoumána, stejně jako parametry modelu směsí Gaussových funkcí tak, aby nedošlo k přeučení či přílišné generalizaci rozpoznávacího algoritmu a byl zachován fakt, že v daném modelu příslušná gesta vykazují periodičnost.

Možností rozšíření je také využití jiného modelu gest, než je Gaussův model, například skrytých Markovových modelů či metody dynamického borcení času. Za zvážení pak ale určitě stojí dostupnost dostatečného množství trénovacích příkladů pro tyto modely, které jsou jistě silným nástrojem, ovšem jejich trénování pro co nejpřesnější fungování je věcí správného výběru dat a získání dostatečného množství dostatečně reprezentativních příkladů.

Co se týká jiných částí práce, než je vlastní klasifikace, během testování funkčnosti bylo zaznamenáno, že zvolená metoda sledování regionů v některých případech může selhat vlivem (především) neplatnosti apriorních předpokladů o spojování a rozdělování regionů barvy pokožky. Lze proto uvažovat o výběru jiné metodě sledování nebo zpřesnění stávající například implementací zotavení ze stavu, kdy nějaký region – část těla účastníka – je po dlouhou dobu „ztracen“ trackovacím algoritmem.

Soustavná činnost na rozsáhlém díle jako je tato práce jistě umožňuje člověku získat velkou osobní zkušenost. Je třeba zvažovat mnohé možné postupy a přijímat dílčí rozhodnutí, jejichž správnost nebo v menší či větší míře chybnost může mít své důsledky v úspěšném dosažení kýžených výsledků v dalších fázích. Nedílnou částí plnění práce podobné této je rozšíření obzorů na poli odborných znalostí, povědomí o různých přístupech a řešeních, jejich spolehlivosti a úskalí. Praktická část vývoje aplikací dává další cenné zkušenosti a empirické znalosti návrhu programů, jejich tvorby, ladění a odhalování příčinných souvislostí.

Literatura

[1] AMI Consortium: *Augmented Multi-party Interaction*. Projektové stránky, 3.5.2008. URL: http://www.amiproject.org/

[2] Ap-apid, R. *An Algorithm for Nudity Detection*. Manila, Filipíny, 2005. URL: http://www.math.admu.edu.ph/~raf/pcsc05/proceedings/AI4.pdf

[3] Jae, Y. L., Suk, I. Y. *An Elliptical Boundary for Skin Color Detection*. Soul, Korea, 2002. URL: http://ailab.snu.ac.kr/publication/down/CISST02-169CT.pdf

[4] Vezhnevets, V., Sazonov, V., Andreeva, A. *A Survey on Pixel-Based Skin Color Detection Techniques*. Moskva, Rusko, 2004. URL: http://graphics.cs.msu.ru/en/publications/text/gc2003vsa.pdf

[5] Brashear, H., Park, K.-H., Lee, S. *American Sign Language Recognition in Game Developement for Deaf Children*. Atlanta, Georgia, USA, 2006. URL: http://www.cc.gatech.edu/~brashear/pubs/ASSETS2006.pdf

[6] Chai, D., Ngan, K. N. *Face Segmentation Using Skin Color Map in Videophone Applications*. IEEE Transactions on Circuits and Systems for Video Technology, svazek 9, strany 551-564. Perth, Austrálie, 2003.

[7] Prem, K., Prassd, G., Subbanna, P. B., Sumam D. S. *Human Face Detection and Tracking Using Skin Color Modeling and Connected Component Operators*. Surathkal, Indie, 2002. URL: http://www.ece.arizona.edu/~pgsangam/ietepaper.pdf

[8] Lu, S., Tsechpenakis, G., Metaxas, D. N. *Blob Analysis of the Head and Hands: A Method for Deception Detection*. Proceedings on the 38th Annual Hawaii International Conference on System Sciences, strany 21-31. Havaj, USA, 2005.

[9] Tanibata, N., Shimada, N., Shirai, Y. *Extraction of Hand Features for Recognition of Sign Language Words*. Osaka, Japonsko, 2002. URL: http://www.cipprs.org/vi2002/pdf/s7-7.pdf

[10] Kölsch, M., Turk, M. *Robust Hand Detection*. Santa Barbara, California, USA, 2004. URL: http://www.movesinstitute.org/~kolsch/handvu/KolschTurk2004RobustHandDetection.pdf

[11] Hoshino, K., Tanimoto, T. *Realtime Estimation of Human Hand Posture for Robot Hand Control*. IEEE International Symposium on Computational Intelligence in Robotics and Automation, strany 99-104. Espoo, Finsko, 2005.

[12] Viola, P., Jones, M. *Robust Real-time Object Detection*. Vancouver, Kanada, 2001. URL: http://www.wisdom.weizmann.ac.il/~vision/courses/2003_2/ICCV01-Viola-Jones.pdf

[13] Wahde, M. a kol. *Computer Vision Based System for Dynamic Gesture Recognition*. Göteborg, Švédsko, 2007. URL: http://www.ituniv.se/program/isd/Thesis/rapport_0425.pdf

[14] Argyros, A. A., Lourakis, M. I. A. *Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera*. Heraklion, Kréta, 2004. URL: http://www.ics.forth.gr/~argyros/mypapers/2004_05_eccv_hand_tracking_2d.pdf

[15] Černocký, J. *Zpracování řečových signálů – studijní opora*. Studijní materiál, VUT Brno, Fakulta informačních technologií, 2006.

[16] Corradini, A. *Dynamic Time Warping for Off-line Recognition of a Small Gesture Vocabulary*. Proceedings of the IEEE ICCV Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, strany 82-89. 2001.

[17] ten Holt, G. A., Reinders, M. J. T., Hendriks, E. A. *Multi-Dimensional Dynamic Time Warping for Gesture Recognition*. Delft, Nizozemí, 2007. URL: http://ict.ewi.tudelft.nl/pub/gineke/DTW-vASCI.pdf

[18] Fraile, R., Maybank, J. S. *Vehicle Trajectory Approximation and Classification*. Reading, Velká Británie, 1998. URL: http://www.bmva.ac.uk/bmvc/1998/pdf/p144.pdf

[19] Rigoll, G., Kosmala, A., Eickeler, S. *High Performance Real-Time Gesture Recognition Using Hidden Markov Models*. Duisburg, Německo, 1998. URL: http://citeseer.ist.psu.edu/cache/papers/cs/13837/http:zSzzSzwww.fb9-ti.uni-duisburg.dezSzpublzSz97zSz137101gw.pdf/rigoll98high.pdf

[20] Kohler, M. *Vision Based Hand Gesture Recognition Systems*. Prezentační stránky výzkumu na Computer Graphics, University of Dortmund, 10.4.2008. URL: http://ls7-www.cs.uni-dortmund.de/research/gesture/vbgr-table.html

[21] Hassink, N., Schopman, M. G. *Gesture Recognition in a Meeting Environment*. Enschede, Nizozemí, 2006. Diplomová práce, University of Twente, Department of Electrical Engineering, Mathematics and Computer Science.

[22] Procházka, D. *Uživatelská rozhraní založená na rozpoznávání obrazu*. Brno, 2005. Diplomová práce, Ústav počítačové grafiky a multimédií FIT VUT Brno.

[23] Jiřík, L. *Gesture Recognition*. Valladolid, Španělsko, 2005. Bakalářská práce, FIT VUT Brno.

[24] NIST/SEMATECH e-Handbook of Statistical Methods: *Double Exponential Smoothing*. Online příručka U.S. Commerce Department's, 2.5.2008. URL: http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc433.htm

[25] Sbalzarini, I. F., Theriot, J., Koumoutsakos, P. *Machine Learning for Biological Trajectory Classification Applications*. Curych, Švýcarsko, 2002. URL: http://www.cbl.ethz.ch/research/docs/Sbalzarini2002b.pdf

[26] Intel Corporation: *Open Source Computer Vision Library (OpenCV)*. Knihovna pro práci s obrazem a videem, 11.4.2008. URL: http://opencvlibrary.sourceforge.net/

[27] Wikipedia: *Color model*. Internetová encyklopedie, 4.5.2008. URL: http://en.wikipedia.org/wiki/Color_model

[28] Chang, H., Robles, U. *Face Detection*. Stránky o segmentaci barvy pokožky a detekci objektů, 4.5.2008. URL: http://www-cs-students.stanford.edu/~robles/ee368/skincolor.html

[29] Garcia, C., Tziritas, G. *Automatic Face Detection in Complex Color Images*. Stránky o detekci tváře, 4.5.2008. URL: http://www.csd.uoc.gr/~cgarcia/FACE/Face.html

[30] Wikipedia: *Dynamic time warping*. Internetová encyklopedie, 4.5.2008. URL: http://en.wikipedia.org/wiki/Dynamic_time_warping

[31] Nam, Y., Wohn, K. *Recognition of Space-Time Hand-Gestures using Hidden Markov Model*. Taejeon, Korea, 1996. URL: http://citeseer.ist.psu.edu/cache/papers/cs/13148/http:zSzzSzdangun.kaist.ac.krzSz~nyhzSzcamera1.pdf/nam96recognition.pdf

Seznam obrázků a jejich zdrojů

Není-li uveden zdroj odkazem do seznamu literatury, jedná se o obrázek vlastní nebo o výstup z vytvořeného programu.

Číslo obrázku	**Popis, zdroj**
5-6	Chybně detekovaná periodicita
5-7	Chybná lokalizace dlaně
6-1	Analogie navrženého algoritmu v HMM

Seznam příloh

Příloha 1

Vybrané algoritmy a vzorce z teoretické části

Modifikace algoritmu AdaBoost využitý Violou a Jonesem v práci [12]

dány vzorové obrazy $(x_1, y_1) .. (x_n, y_n)$, kde $y_i = 0, 1$ pro negativní resp. pozitivní příklady;

inicializuj váhy $w_{1,i} = 1 / 2m$ pro $y_i = 0$; $w_{1,i} = 1 / 2l$ pro $y_i = 1$, kde m je počet negativních a l pozitivních příkladů;

for $t = 1 .. T$:

normalizuj váhy

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

aby w_t bylo pravděpodobnostní rozložení;

pro každou charakteristiku j natrénuj klasifikátor h_j, který využívá pouze jednu charakteristiku, chyba je dána vzorcem

$$\varepsilon_j = \sum_i w_{t,i} \left| h_j(x_i) - y_i \right| ;$$

vyber klasifikátor h_t s nejmenší chybou ε_t;

aktualizuj váhy

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

kde $e_i = 0$ pokud vzor x_i je klasifikován správně, jinak $e_i = 1$, $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$;

natrénovaný (silný) klasifikátor je dán

$$h(x) = \begin{cases} \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \ldots 1 \\ jinak \ldots 0 \end{cases} ;$$

kde $\alpha_t = log(1 / \beta_t)$;

Rovnice pro nalezení parametrů elipsy (c_x, c_y, α, β, θ) podle autorů [14]

jsou dány body p_i nějakého blobu a kovarianční matice Σ jejich rozložení

$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix}$$

potom parametry elipsy jsou

(c_x, c_y)...*gravity _ center*

$$\alpha = \sqrt{\lambda_1}$$

$$\beta = \sqrt{\lambda_2}$$

$$\theta = \tan^{-1}\left(\frac{-\sigma_{xy}}{\lambda_1 - \sigma_{yy}}\right)$$

kde

$$\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Delta}{2}$$

$$\lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Delta}{2}$$

$$\Delta = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 - (2\sigma_{xy})^2}$$

Pseudokód algoritmu pro výpočet vzdálenosti dvou sekvencí symbolů pomocí DTW [30]

s a *t* jsou porovnávané sekvence, *d* je matice vzdáleností jednotlivých symbolů z řetězců *s* a *t*

```
int DTW_Dist(char s[1..n], char t[1..m], int d[1..n, 1..m]){
      int DTW[0..n, 0..m];
      int i, j, cost;

      for (int i = 1 .. m)
            DTW[0, i] := infinity;
      for (int i = 1 .. n)
            DTW[i, 0] := infinity;

      DTW[0,0] = 0;

      for (int i = 1 .. n)
            for (int j = 1 .. m){
                  cost = d[s[i], t[j]];
                  DTW[i,j] := cost + min(DTW[i-1, j]
                                         DTW[i, j-1],
                                         DTW[i-1, j-1]);
            }
      return DTW[n, m];
}
```

Příloha 2

Popis ovládání programů

Program DP_track

Tento program je dávkovou aplikací, která provádí jednotlivé kroky přípravy sekvence k rozpoznávání gest. Jsou jimi

- segmentace obrazu,
- elliipse-fitting (proložení elipsy) – původně zamýšleno pro využití v algoritmu sledování regionů pomocí hypotéz (viz 2.2.2),
- nalezení pozice dlaně (palm localization),
- sledování regionů a
- grafickou prezentaci výsledků.

Běh programu je rozdělen do několika módů, které lze specifikovat prvním parametrem aplikace. Dále uvádím jejich seznam a popis:

`SEGM` provede segmentaci obrazu vytvořením SPI, prahováním a algoritmem spojených komponent; výsledek uloží do binárního souboru, který je nutný pro další zpracování; tento binární soubor je uložen do adresáře, v němž je zpracovávaná sekvence; druhým parametrem musí být název adresáře, v němž leží segmentovaná AVI sekvence, jejíž název je třetím parametrem.

`ELLIPS` načte jednotlivé vysegmentované regiony z binárního souboru vytvořeného parametrem `SEGM`, každým proloží elipsu a výsledky uloží do nového binárního souboru, pro nějž platí totéž jako v popisu `SEGM`, taktéž jsou požadovány druhý a třetí parametr za stejných podmínek.

`ELLIPSSAVE` načte z příslušného binárního souboru vypočtené parametry elips, zakreslí je do jednotlivých snímků sekvence a uloží do adresáře, jehož jméno je požadováno jako čtvrtý parametr; druhý a třetí parametr jsou opět cesta k adresáři a název sekvence ve formátu AVI.

`PALMS` načte jednotlivé vysegmentované regiony z příslušného binárního souboru, provede nalezení dlaní rukou a výsledky uloží do binárního souboru; požadovány jsou také druhý a třetí parametr – cesta k adresáři a název videosekvence v něm.

`PALMSSAVE` načte jednotlivé nalezené pozice dlaní z binárního souboru, zakreslí je do jednotlivých snímků sekvence a uloží do adresáře, jehož jméno je požadováno jako čtvrtý parametr; jako obvykle druhý a třetí jsou jméno adresáře a název sekvence.

`TRACK` je parametrem, jímž je možné uložit snímky ze sekvence s vyznačenými bounding boxy slodovaných regionů, dále uložit do textového souboru trénovací trajektorie podle dané anotace pro program *DP_gmms* nebo uložit trajektorie všech sledovaných regionů v celé sekvenci pro další rozpoznávání týmž programem do textového souboru; ve všech případech je nutné zadat druhý

parametr cestu k adresáři se zpracovávanou sekvencí, jejíž název je parametrem třetím; všechny výše zmíněné binární soubory musí v tomto adresáři již existovat.

Uložení snímků se provede spuštěním aplikace se čtvrtým parametrem názvem adresáře, do nějž mají být sníky uloženy.

Vytvoření trénovacích trajektorií se provádí spuštěním aplikace se čtvrtým parametrem názvem anotačního textového souboru včetně cesty k němu jako jeden parametr a zadáním pátého parametru **SAVEEXAM**; v anotačním souboru musí být na každém řádku uloženy parametry jednoho gesta v tomoto pořadí:

`start_time_in_frames end_time_in_frames gesture_locus gesture_class\n`

`start_time_in_frames` je počátek gesta označený číslem snímku od začátku, první snímek má hodnotu 0;

`end_time_in_frames` je konec gesta označený číslem snímku od začátku;

`gesture_locus` je lokus (část těla vykonávající příslušné gesto), pro pravou ruku prvního účastníka (vlevo) má hodnotu 1, pro levou ruku prvního účastníka hodnotu 2, pro pravou ruku druhého účastníka (vpravo) má hodnotu 3 a pro levou ruku druhého účastníka hodnotu 4;

`gesture_class` je třída gesta (hodnota 1..N, kde N je počet tříd);

soubor musí být ukončen hodnotou -1 na samostatném řádku.

Výsledkem je M textových souborů s pozicemi zadané části těla v určených časech – framech; M je počet gest zadaných ve vstupním souboru. Každý soubor má následující tvar:

`x_velocity y_velocity gesture_class\n` (v čase T_{start} + 1)

`x_velocity y_velocity gesture_class\n` (v čase T_{start} + 2)

`...`

`0 0 0` (ukončovací zarážka)

Uložení trajektorií všech objektů lze uskutečnit zadáním jména textového souboru, do nějž mají být trajektorie uloženy (opět včetně cesty), jako čtvrtého parametru a pátý nastavit na **SAVETRAJ**; ve výsledném souboru potom budou trajektorie dány v tomto formátu:

`x_R1 y_R1 x_L1 y_L1 x_R2 y_R2 x_L2 y_L2\n` (snímek 0)

`x_R1 y_R1 x_L1 y_L1 x_R2 y_R2 x_L2 y_L2\n` (snímek 1)

`...`

`-2 -2` (zarážka)

kde x a y značí x-ovou a y-ovou souřadnici, R pravou ruku, L levou ruku, 1 první osobu (vlevo) a 2 druhou osobu (vpravo).

Pro jednodušší provedení všech důležitých kroků v jednom běhu byly implementovány ještě akce:

ALLTMP, která provede segmentaci, výpočet aproximačních elips a lokalizaci dlaní a uložení příslušných binárních souborů; vyžaduje zadání adresáře (druhý parametr) a kýžené videosekvence (třetí); a

ALL, která provádí to samé jako ALLTMP a navíc výsledky uloží do adresáře daného čtvrtým parametrem.

Program param_fit

Program vypíše tabulku hodnot průměrné chyby na vzorek při vyhlazení dvojitým exponenciálním filtrem o parametrech α a β. Hodnota na pozici ($i_{řádek}$, $j_{sloupec}$), i, j = 1..20 je chyba filtru s parametry α = (i – 1)/20, β = (j – 1) / 20. Aplikace nepřebírá z příkazové řádky žádné parametry, názvy textových souborů s anotovanou a původní trajektorií je možné provézt přímo ve vstupní metodě `main(...)` zdrojového kódu.

Program DP_gmms

Trénování modelu GMM, vlastní rozpoznání gest v trajektoriích a prezentaci výsledků provádí program *DP_gmms*. Spuštěn musí být vždy s pěti parametry:

parametr1 je jednotným názvem souborů s trénovacími daty včetně cesty číslovaných od nuly, kde číslo souboru je nahrazeno znakem * (asterisk); například pro soubory *trajectory0.txt* až *trajectory18.txt* uložené v adresáři *D:\train_data* je tímto parametrem `parametr1` = *D:\train_data\trajectory*.txt*); tyto příklady musí mít tvar popsaný v této příloze v části o programu *DP_track* (akce TRACK s přepínačem SAVEEXAM);

parametr2 udává počet trénovacích souborů (např. pro soubory *trajectory0.txt* až *trajectory18.txt* je `parametr2` = *19*);

parametr3 definuje cestu a název k textovému souboru s trajektoriemi všech sledovaných objektů; soubor musí mít formát definovaný v části o programu *DP_track* (akce TRACK s přepínačem SAVETRAJ);

parametr4 je název souboru AVI včetně cesty, do kterého mají být zakresleny trajektorie; zadávat jiný soubor než jaký byl použit k vygenerování trajetorií daných souborem `parametr3` ztrácí smysl a může vézt k pádu programu;

parametr5 udává název adresáře, do nějž mají být uloženy snímky sekvence včetně vyznačených trajektorií.

Příloha 3

Náhled do struktury programů

Programy *DP_track* a *DP_gmms* používají třídy pro práci s barevným obrazem *ImgRGB*, podstatnou část jejíž deklarace uvádím a popisuji níže.

```
class ImgRGB{
public:
	typedef struct T_Color {                                              (3)
		unsigned char R, G, B;
	};
	typedef struct T_Point {                                              (6)
		int x, y;
	};
	typedef vector<T_Point> T_Points;

	ImgRGB(void);                                                         (11)
	ImgRGB(const int x_s, const int y_s);                                 (12)
	ImgRGB(const char * filename);                                        (13)
	ImgRGB(const ImgRGB& img);                                            (14)
	ImgRGB(const char * filename, const unsigned char r, const unsigned   (15)
		char g, const unsigned char b);

	T_Color getPix(const int x, const int y) const;                       (18)
	void setPix(const int x, const int y, const T_Color value);          (19)

	void draw_point(const int x, const int y, const int size, const T_Color (21)
			color);
	void drawStrAt(const int x, const int y, const char * str, T_Color col);(23)
	void line(int x0, int x1, int y0, int y1, const T_Color color);       (24)
	void ellipse(int x, int y, const float a, const float b, const float  (25)
			angle, const T_Color color, const T_Color color2);

	int getX() const;
	int getY() const;

	void RGBtoYCbCr();                                                    (31)
	void gray_scale();                                                    (32)
	void threshold(const unsigned char intensity);                        (33)

	void saveFile(const char * filename) const;                           (35)

	void setPix_F(const int i, const int j, const float value);           (37)
	float getPix_F(const int i, const int j) const ;                      (38)
	void saveFloat(const char * filename) const;                          (39)
	void thrFloat(const float thr);                                       (40)

	int CCA();          // vraci pocet labelu                             (42)
	int CCA2();         // vraci label nejvetsiho                         (43)
	T_Points get_region(const int label);                                 (45)

	~ImgRGB(void);

private:
	unsigned char * R;                                                    (49)
	unsigned char * G;                                                    (51)
	unsigned char * B;                                                    (52)

	float * fl;                                                           (54)
	bool * thr;                                                           (55)
	int * labels;                                                         (56)
};
```

Třída vyváží základní struktury, jako je bod a třísložková barva – řádky (3) a (6). Umožňuje volání konstruktorů, které vytvoří obraz daných dimenzí (12), respektive načtou soubor ze zadaného souboru (13) či načtou soubor a uloží do paměti masku danou barvou v RGB modelu (15). Implementovány jsou také implicitní a kopírovací konstruktor (11) a (14).

Uložení obrázku do souboru typu *BMP* provádí metoda (35).

Definovány jsou základní metody pro přístup k obrazovým datům (18) a (19). Metody (21) až (25) provádí vykreslení základních primitiv do obrazových dat.

Důležitou skupinou metod jsou metody pro převod obrazu do modelu YCbCr, do stupňů šedi a pro jeho prahování (31) – (33).

Třída též umožňuje práci s reprezentací obrazu pomocí hodnot s plovoucí čárkou (floating point operations) definicí základních metod (37) – (40). Implementován byl také algoritmus spojených komponent na prahovaném obrazu. Metoda `CCA` (42) vrací počet nalezených regionů a `CCA2` (43) label největšího regionu. Přistupovat k jednotlivým spojeným regionům lze pomocí funkce (45), která vrací vektor bodů regionu zadaného parametrem.

Privátní proměnné typu ukazatel definované na řádcích (49) až (56) uchovávají adresy v paměti s uloženými obrazovými a dalšími daty. Jejich inicializaci provádí konstruktor.

Dále uvádím popis funkcí a tříd využitých pouze v jednom programu.

Program DP_track

Vstupní metoda tohoto programu je rozdělena do několika částí, které provádějí jednotlivé kroky popsané v příloze o ovládání programu. Využita je zde také třída `Bayes`, která implementuje natrénování Gaussova rozložení množiny bodů:

```
class Bayes {
public:
	struct T_Vect2 {                                                (3)
		float v1, v2;
	};
	typedef vector<T_Vect2> T_Train_Set;                            (6)

	Bayes();
	Bayes(const T_Train_Set& input_set);                            (9)

	float prob(const T_Vect2& point);                               (11)

private:
	float mean1, mean2;                                             (14)
	float cov11, cov22, cov12;                                      (15)
	float koeficient;                                               (16)
};
```

Třída vyváží strukturu dvouprvkového vektoru a typ vektoru těchto vektorů (trénovací množina) – (3) a (6). Uživatelský konstruktor (9) provede výpočet parametrů modelu (14) – (16) z množiny dané parametrem. Poté je možné určovat pravděpodobnost příslušnosti neznámého vektoru do spočteného Gaussova rozložení pomocí funkce hustoty pravděpodobnosti (11).

V souboru `track_fnc.h` jsou definovány funkce využité při implementaci trackingu, jako je např. funkce výpočtu vzdálenosti dvou bounding boxů nebo predikát jejich překrytí. Za zmínku jistě také stojí, že je zde definována funkce implementující vyhlazení trajektorie `traj` pomocí dvojitého exponenciálního filtru s danými parametry `alpha` a `beta`:

```
void smooth(ImgRGB::T_Points& traj, const float alpha, const float beta);
```

Dalším souborem definujícím důležité funkce je `video_track.h`, obsahuje jak funkce pro tracking, tak aproximaci elipsou, lokalizaci dlaně a struktury pro reprezentaci segmentovaného snímku, parametrů aproximačních elips regionů, pozice dlaně aj.

Nalezení dlaně provádí funkce (1) a (2) volané v tomto pořadí. První z těchto funkcí z obrazu daného prvním parametrem vezme výsek definovaný strukturou `section` a provede konvoluci s dvěma jádry – viz část 4.3 – velikosti `vel_jadro` a výsledek vrátí jako druhý parametr.

```
void find_finger_cluster(const ImgRGB * obraz, ImgRGB& obraz_out,
                         const BBox section, const int vel_jadro);          (1)
void localize_fingers(ImgRGB& pict, Palm& plm);                             (2)
```

Funkce druhá prahuje obraz `pict` a nalezne těžiště největšího spojitého regionu. Dále také algoritmem k-means najde dva shluky (pro využití v případech spojení dvou regionů ruky – řeší funkce trackingu). Tyto informace vrací ve struktuře `plm`.

Důležitými definovanými funkcemi jsou také `initTrack` a `doTrack`. První z nich vychází z apriorních předpokladů na začátku sekcence o pozici osob a provede prvotní přiřazení regionů k jednotlivým částem těla osob. Druhá potom sleduje regiony po zbytek sekvence. Obě vracejí nalezené stavy osob ve strukturách typu `Person_state`, jejíž zjednodušenou definici s komentáři uvádím zde:

```
// reprezentuje polohu regionu jedné osoby v daném čase
struct Person_state {
       char label[20];           // popiska
       BBox head_reg;            // BB hlavy

       BBox lhand_reg;                          // BB levé ruky
       ImgRGB::T_Point lhand_center;            // pozice levé dlaně
       BBox rhand_reg;                          // BB pravé ruky
       ImgRGB::T_Point rhand_center;            // pozice pravé dlaně

       // příznaky spojení
       unsigned char tokens_con;
       // 1 .. spojení LH a RH
       // 2 .. spojení LH a HD
       // 3 .. spojení RH a HD
       // 4 .. spojení LH + RH + HD
       // 0 .. jinak

       // příznaky zakrytí
       unsigned char tokens_occ;
       // 1 .. zakrytí LH
       // 2 .. zakrytí RH
       // 3 .. zkarytí LH a RH
       // 4 .. zakrytí HD
       // 5 .. zakrytí HD a LH
       // 6 .. zakrytí HD a RH
       // 0 .. jinak
};
```

Zkratky `LH`, `RH` a `HD` značí levou ruku (left hand), pravou ruku (right hand) a hlavu (head) v tomto pořadí.

Funkce `drawTrackState`, která přebírá referenci na `ImgRGB` a struktury `Person_state`, vykresluje stav v daném okamžiku.

Program DP_gmms

Za účelem implementace GMM vznikla knihovna základních funkcí pro práci s maticemi a vektorem, jejíž hlavičkový soubor `mat_vect.h` zpřístupňuje funkce pro výpočet determinantu čtvercové matice, inverzní matice a násobení schémat $\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}$ a $\boldsymbol{v}\boldsymbol{v}^T$, kde $\boldsymbol{v}$ je vektor a $\boldsymbol{M}$ matice s odpovídajícími rozměry.

Nejdůležitější třídou využitou v tomto programu je `gauss_mix`:

```
class gauss_mix {
public:
	gauss_mix();                                                          (3)
	gauss_mix(const gauss_mix& ref);                                      (4)
	gauss_mix(const int dimensions, const int n_gauss_fnc);               (5)

	void feed(const float* point);                                        (7)
	void train(const int iterations);                                     (8)

	float pdf(const float* point);                                        (10)
	int win_distribution(const float* point);                             (11)

	~gauss_mix();

private:
	struct class_point {                                                  (16)
	public:
		float * data;
		int cls;
	};

	vector<class_point> train_set;                                        (22)
	vector<float *> mean_vals;                                            (23)
	vector<float *> cov_matrices;                                         (24)
};
```

Uživatelský konstruktor (5) vytvoří model pro práci s daty dimenze zadané prvním parametrem s počtem shluků daným parametrem druhým. Dostupné jsou i implicitní a kopírovací konstruktor (3) a (4). Metoda `feed` přidá do trénovací množiny (22) o prvcích typu struktury (16) vektor z parametru. Voláním metody `train` lze provézt učení modelu, tj. přiřazení tříd trénovacím bodům (16), čímž se získají hodnoty vektorů středních hodnot (23) a kovariančních matic (24). Parametr udává maximální počet iterací.

V souboru `recog_fnc.h` jsou definovány konstanty z rovnic (53) až (56) v části 4.6.2. Tedy délky oken pro segmentaci, hodnoty prahů segmentace a minimální délka jedné periody gesta SSG. Dále je zde definována funkce

```
void recog(gauss_mix& c1, gauss_mix& c2, vector<T_Gesture>& gestures,
		const T_Points traj_velocity, const int locus);
```

Ta vloží do vektoru s položkami typu struktury `T_Gesture` všechny detekované aktivity, jak je popsáno v části o rozpoznávání. Vstupy jsou dva natrénované modely HMM, vektor rychlostí a locus gesta (který je přiřazen všem gestům nalezeným v jednom volání této funkce).

Příloha 4

Popis dat na přiloženém DVD

Struktura dat na přiloženém DVD je následující:

- `Data`
  - `AVI – binarni soubory – trajektorie`
    - *vytvořené pomocné binární soubory a textové soubory s trajektoriemi pro 16 videosekvencí, v nichž se vyskytují gesta*
  - `Trajektorie pro Exp Filter`
    - *anotovaná a původní trajektorie využitá pro nalezení parametrů exponenciálního filtru*
  - `Trenovaci data pro GMM`
    - *sada 27 trajektorií v textové podobě*
  - `Vysledky`
    - `Rozpoznavani`
    - `Ukazka tracking`
      - *obojí ve formě sekvence JPG souborů*
- `Programy`
  - `DP_gmms`
  - `DP_track`
  - `param_fit`
- `Technicka zprava`
  - `Literatura`
    - `Barevne modely`
    - `DTW`
    - `HMM`
    - `Klasifikace`
    - `NNets`
    - `Ostatni`
    - `Tracking`
      - *online získané materiály k problematice, většinou ve formě PDF*
  - `Obrazky`
    - *všechny obrázky využité v tomto textu*
  - `Ostatni`