ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Detekce rtů ve videozáznamech

2012

Vypracoval: Bc. Miroslav Hlaváč
Vedoucí práce: Ing. Marek Hrúz

!!! Originál zadání !!!

Místo tohoto listu bude originál zadání…

Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum: 18. 5. 2012

. . . . . .
podpis

Poděkování

Děkuji vedoucímu diplomové práce Ing. Marku Hrúzovi za cenné rady, vedení a veškerou další pomoc při zpracování této práce. Dále bych chtěl poděkovat Ing. Petru Stanislavovi za spolupráci při nahrávání potřebných videozáznamů.

Anotace

HLAVÁČ, M. *Detekce rtů ve videozáznamech*, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky, Plzeň, 2012.

Klíčová slova: detekce rtů, active appearance model, active shape model, tracking, fitting

Tato práce se zabývá detekcí a sledováním tvaru rtů ve videozáznamech. Pro tento účel je vytvořen statistický model tvaru rtů a následně i statistický model textury rtů. Algoritmus pro výpočet i testování modelu byl vytvořen při zpracování této práce a je přiložen na DVD. V závěru jsou otestovány různé možnosti trénování úprav parametrů modelu.

Summary

HLAVÁČ, M. *Detection of lips in video sequences*, University of West Bohemia in Pilsen, Faculty of Applied sciences, Department of Cybernetics, Pilsen, 2012.

Klíčová slova: lips tracking, active appearance model, active shape model, , model fitting

This thesis deals with detection and tracking of lips in video sequences. Statistical model is used for modeling lips shape and appearance. The algorithm for training and detection was created as a part of this work and is availible on the attached DVD. Various experiments are made with training of displacement parameters.

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

V úloze strojového rozpoznávání řeči je kvalita rozpoznávání závislá na míře okolního šumu. Tento šum může být způsobem například rušným prostředím nebo postižením řečníka. Rozpoznávací algoritmus pak nedává přesné výsledky a je vhodné využít dodatečných informací ke zvýšení přesnosti rozpoznávání. Takovouto informací muže být i vizuální záznam pohybu řečníkových rtů. Při audiovizuálním rozpoznávání řeči slouží vizuální složka jako dodatečná informace k audio signálu. Motivací k detekci rtů je tedy získání dodatečné informace pro zlepšení rozpoznávání řeči z audio signálu. Cílem této práce je nalézt a otestovat vhodnou metodu k detekci tvaru rtů řečníka ve videozáznamu.

K detekci rtů v obraze se v současnosti využívá několika metod. Jednou z používaných metod je například upravený H∞ filtr. Tento filtr je založený na principu stavového modelu, kde dynamika signálu je modelována stavovou rovnicí, zatímco pozorování včetně šumu je dáno rovnicí pozorování [2]. Další používanou metodou je vylepšený Snake model (Active countour model). Tento model pracuje na principu energii minimalizující spline funkce, která je inicializována v blízkosti požadovaného lokálního minima a dále kontrolována vnější a vnitřní omezující silou [2]. Ke sledování tvaru rtů se dále využívá například Adaptivní fuzzy částicový filtr (Adaptive fuzzy particle filter). Tento filtr je založen na Bayesovské metodě, která rekurzivně odhaduje stav sledovaného objektu jako aposteriorní rozdělení konečného souboru vážených vzorků [4]. Metody založené na statistickém modelování hledaného byly zkoumány a rozvíjeny Timem Cootesem od začátku devadesátých let dvacátého století. Jedná se o dvě metody, Active shape model a Active appearance model. Určitým přechodem mezi Active appearance modelem a Snake modelem je vyhledávání založené na algoritmu TASOM (Time Adaptive Self Organizing Map). Výhodou tohoto algoritmu je nezávislost modelu na rečníkovi a také možnost fungovat v reálném čase [5].

Stejně jako předchozí zmíněné postupy, kromě Active shape modelu a Active appearance modelu, je i tento určen k nalezení vnějšího tvaru rtů. Jak ale vyplývá z práce Ing. Petra Císaře, Ph.D., je nejpodstatnější část informace obsažena v textuře, která je vymezená vnitřní konturou úst, popisující přítomnost a pozici jazyka a zubů [6].

Metody Tima Cootese byly vybrány pro účely řešení této práce vzhledem k možnosti modelovat i vnitřní tvar rtů a také vzhledem k jejich srozumitelnosti a dostupnosti informačních zdrojů. Pokud není uvedeno jinak, vychází teoretické informace z publikace Tima Cootese [1].

Kapitola 2

Statistické modely tvaru a vzhledu

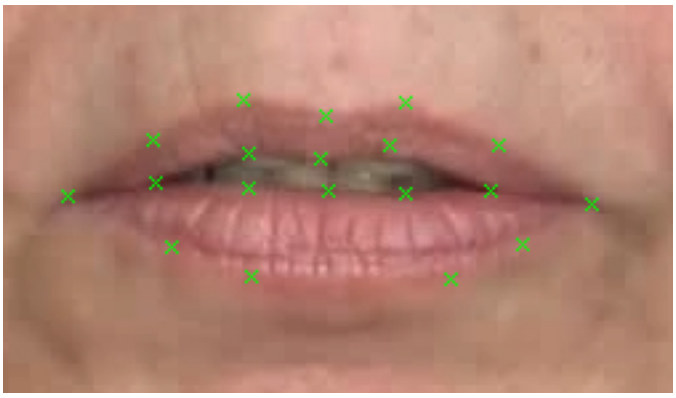
V této kapitole nastíním problematiku statistických modelů tvaru, jejich využití a metod potřebných pro jejich výpočet a použití. Existují dvě hlavní metody; Active Shape Model, který využívá pouze informaci o tvaru objektu, a Active Appearance model, který přidává navíc informaci o textuře objektu.

2.1 Statistické modely tvaru

Statistické modely tvaru se využívají k reprezentaci objektů v obrázcích. Statistická analýza se aplikuje na data reprezentující určitý tvar za účelem nalezení a vytvoření jeho modelu.

2.1.1 Významné body

Existuje více možností jak reprezentovat tvar. Pro účely statistických modelů tvaru je dobré použít významných bodů, které jsou dobře odlišitelné od okolí a vyskytují se ve všech obrazech z trénovací množiny. Příkladem takovýchto bodů je konec prstu nebo koutek úst. Na souřadnicích těchto bodů lze provádět statistickou analýzu. Tvar objektu pak dostaneme vhodným propojením bodů. Takovýto tvar je velmi hrubý a lze ho tedy doplnit i méně významnými body ležícími mezi body významnými, reprezentující například hrany objektu (Obrázek 2.1). Pro zjištění vzájemné závislosti mezi pohybem jednotlivých bodů využijeme Analýzy hlavních komponent.



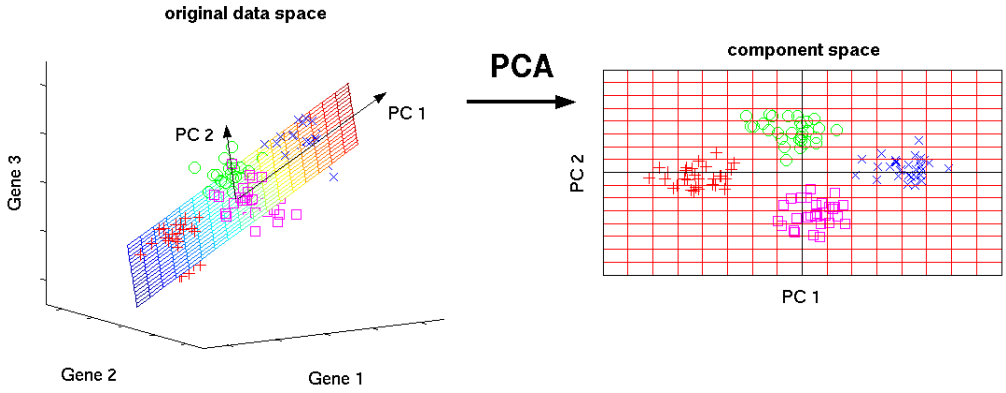
Obrázek 2.1: Významné body

2.1.2 Analýza hlavních komponent

Analýza hlavních komponent je český překlad Principal Component Analysis (dále jen PCA). Tato metoda slouží k dekorelaci data a je často využívána k redukci dimenze dat s možností ovlivnění množství ztracené informace. Jedná se v podstatě o ortogonální transformaci množiny pozorování na soubor lineárně nezávislých proměnných. Transformace je definována tak, že vektory prvků s největší variancí jsou první. PCA je možné vypočítat několika různými způsoby. Jedním z nejpoužívanějších je metoda kovariance. Před vlastním výpočtem seřadíme data jednotlivých měření do řádkových vektorů a vytvoříme matici, kde každý řádek reprezentuje konkrétní měření.

$$C(i,j) = (x_i - \mu_i)(x_j - \mu_j)^T \tag{2.1}$$

Prvky kovarianční matice C vypočteme podle vzorce 2.1, kde x_i a x_j jsou i-tý a j-tý vektor dat a μ je příslušná střední hodnota. Z kovarianční matice vypočteme použitím dostupného algoritmu matici vlastních vektorů Φ a vlastní čísla. Velikost vlastních čísel udává významnost příslušných vlastních vektorů. Pokud vydělíme vlastní čísla jejich celkovým součtem, dostaneme jejich procentuální hodnotu. Díky tomu můžeme redukovat počet vlastních vektorů takovým způsobem, kdy zachováme 99% rozptylu dat. Procentuální hodnoty vlastních čísel pak sčítáme dokud nedostaneme požadovanou hodnotu a ostatní vlastní čísla a k nim náležící vektory můžeme vypustit.



Obrázek 2.2: PCA [3]

2.1.3 Lineární model tvaru

Výpočet tvaru probíhá podle vzorce 2.2.

$$x = \overline{x} + \Phi b \tag{2.2}$$

Φ je matice vlastních vektorů trénovacích dat tvaru, $\overline{x}$ je střední hodnota trénovacích dat tvaru, b je vektor parametrů modelu a x je vypočtený tvar. Z tohoto vzorce vyplývá, že výsledný tvar může být měněn pomocí změny parametrů modelu b.

2.1.4 Nasazení modelu do nových bodů

Pozice bodů modelu v obraze není určena pouze parametry modelu b, ale také translací po osách x a y, rotací θ a mírou s. Pokud shrneme všechny tyto parametry do jedné funkce T dostaneme rovnici modelu ve tvaru

$$x = T_{X_t,Y_t,s,\theta}(\overline{x} + \Phi b) \tag{2.3}$$

Pro dva body x a y si lze transformační funkci představit takto

$$T_{X_t,Y_t,s,\theta}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} s\cos\theta & s\sin\theta \\ -s\sin\theta & s\cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.4}$$

Pokud nyní chceme najít takové parametry b, pomocí kterých vypočteme tvar x odpo-

vídající požadovaným bodům v obraze Y, minimalizujeme vlastně výraz

$$||Y - T_{X_t,Y_t,s,\Theta}(\overline{x} + \Phi b)||^2 \tag{2.5}$$

Tento postup se dá vyjádřit iterativním procesem:

1. Nastavení parametrů $b = 0$
2. Výpočet tvaru z rovnice modelu $x = \overline{x} + \Phi b$
3. Nalezení parametrů (X_t, Y_t, s, θ), které nejlépe mapují x do bodů Y
4. Promítnutí bodů Y do souřadné soustavy

$$y = T^{-1}_{X_t,Y_t,s,\theta}(Y) \tag{2.6}$$

5. Aktualizace parametrů modelu

$$b = \Phi^T(y - \overline{x}) \tag{2.7}$$

6. Pokud se parametry b výrazně liší od původních návrat na krok 2.

Zastavovací podmínka v posledním bodě se realizuje porovnáním nových hodnot a hodnot z předchozího kroku. Rozdíl se uvažuje v předem zvolené přesnosti. Pokud je v této přesnosti rozdíl nulový, algoritmus se zastaví.

2.2 Statistické modely vzhledu

K vytvoření kompletního obrazu objektu nestačí pouze informace o jeho tvaru, ale je třeba také modelovat texturu v oblasti ohraničené tvarem. V této kapitole bude vysvětleno, jak lze vytvořit statistický model reprezentující jak tvar, tak texturu objektu. Texturou je zde myšlen soubor intenzit reprezentující barvy v obraze. K vytvoření modelu jsou potřeba trénovací obrazy s vyznačenými významnými body. Z dostupných dat můžeme po výpočtu střední hodnoty tvaru vytvořit warpováním(2.2.1) tvarově nezávislou texturu z každého trénovacího obrazu. Na tyto textury lze aplikovat PCA(2.1.2) podobně jako u modelů tvaru. Předpokladem je, že existuje závislost mezi tvarem a příslušnou texturou a lze tedy vytvořit kombinovaný model tvaru a textury.

2.2.1 Warpování obrazů

Warpování obrazů označuje mapování textury jednoho objektu na jiný objekt. Dochází tedy ke změně tvaru původní textury na tvar požadovaný. Nejjednodušším způsobem warpování je triangulace souřadnic. Oba objekty nejdříve rozdělíme na stejný počet trojúhelníků. Pro každý trojúhelník cílového objektu najdeme příslušný trojúhelník výchozího objektu. Pro každý bod cílového trojúhelníku najdeme příslušný bod výchozího trojúhelníku. Tímto způsobem, kdy hledáme příslušný bod pro každý bod cílového tvaru, nevznikne situace, kdy by zůstal některý bod prázdný. Při hledání souřadnic využíváme toho, že poloha bodu v trojúhelníku se dá vyjádřit pomocí souřadnic jeho vrcholů.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \beta \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \gamma \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \tag{2.8}$$

V této rovnici jsou x_1,x_2 a x_3 vrcholy trojúhelníku a x je souřadnice bodu vyjádřená pomocí těchto vrcholů. Parametry α,β a γ vypočteme z rovnic:

$$\alpha = 1 - (\beta + \gamma) \tag{2.9}$$

$$\beta = \frac{yx_3 - yx_1 - x_3y_1 - xy_3 + x_1y_3 + xy_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \tag{2.10}$$

$$\gamma = \frac{xy_2 - xy_1 - x_1y_2 - yx_2 + x_2y_1 + yx_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \tag{2.11}$$

x a y jsou souřadnice cílového bodu a x_i a y_i jsou souřadnice vrcholů cílového trojúhelníku. Souřadnice hledaného bodu získáme dosazením vrcholů výchozího trojúhelníku do rovnice 2.8.

2.2.2 Lineární model textury

Aplikací PCA na nawarpované textury můžeme podobně jako u modelu tvaru vytvořit lineární model textury.

$$g = \overline{g} + \Phi_g b_g \tag{2.12}$$

Φ_g je matice vlastních vektorů trénovacích dat textury, $\overline{g}$ je střední hodnota trénovacích dat textury, b_g je vektor parametrů modelu textury a g je vypočtená textura.

2.2.3 Kombinovaný model

K vytvoření kombinovaného modelu musíme nejdříve zjistit závislost mezi parametry modelu tvaru a textury. Pro každý trénovací obraz provedeme zpětný výpočet příslušných parametrů. Výsledné parametry seřadíme do vektorů a provedeme na nich PCA.

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s {\Phi_s}^T (x - \overline{x}) \\ {\Phi_g}^T (g - \overline{g}) \end{pmatrix} \tag{2.13}$$

b je vektor parametrů složený z parametrů modelu textury b_g a parametrů modelu tvaru b_s. W_s je váhová matice sloužící k usměrnění parametrů modelu tvaru tak, aby je bylo možné dát do jednoho vektoru s parametry modelu textury (souřadnice vs. pixely). Jednoduchý způsob výpočtu této matice je $W_s = rI$, kde r^2 je poměr absolutní variance intenzit v texturách ku absolutní varianci souřadnic tvarů. Φ_s a Φ_g jsou vlastní vektory kovariančních matic trénovacích dat tvaru a textury. x jsou skutečné souřadnice pro daný obraz, $\overline{x}$ je střední hodnota trénovacích dat tvaru. g je tvarově nezávislá textura příslušného obrazu a $\overline{g}$ je střední hodnota trénovacích dat textury. Po provedení PCA lze stanovit kombinovaný model jako

$$b = \Phi_c c \tag{2.14}$$

P_c jsou vlastní vektory kovarianční matice parametrů obou modelů a c je vektor parametrů kombinovaného modelu. Pomocí parametrů c lze tedy kontrolovat jak tvar tak texturu vypočítaného objektu. Jelikož je model lineární, lze vyjádřit jak tvar tak texturu přímo pomocí parametrů c

$$x = \overline{x} + \Phi_s {W_s}^{-1} \Phi_{cs} c \tag{2.15}$$

$$g = \overline{g} + \Phi_g \Phi_{cg} c \tag{2.16}$$

kde

$$\Phi_c = \begin{pmatrix} \Phi_{cs} \\ \Phi_{cg} \end{pmatrix} \tag{2.17}$$

2.3 Active Shape Model

Active Shape Model, dále jen ASM, je statistická metoda modelů tvarů, které se iterativně deformují za účelem nalezení nelepší shody v pozorovaném obraze. Předpokládejme, že na počátku je dostupný hrubý odhad polohy hledaného objektu v obraze. V takovémto případě můžeme najít model, který bude možné nasadit na hledaný objekt.

Pomocí rovnice 2.3 můžeme vytvořit instanci modelu $\mathbf{X}$, definováním pozice, rotace a míry. K vylepšení přesnosti nasazení využijeme následující iterativní postup:

1. Prozkoumáme okolí každého bodu $\mathbf{X_i}$ a najdeme nejlepší možný bod $\mathbf{X'_i}$, kam by se měl bod $\mathbf{X_i}$ přesunout
2. Aktualizujeme parametry (X_t, Y_t, s, θ, b), tak aby co nejlépe seděli na nové body $\mathbf{X}$
3. Opakujeme proces dokud nedokonverguje k řešení

V praxi hledáme vhodné body k přesunu na normálách hran vypočteného tvaru procházejících významnými body. Nejlepším způsobem jak zjistit, který bod je nejvhodnější, je získat tuto informaci z množiny trénovacích dat. Jednoduše lze tento problém řešit vytvořením statistického modelu profilů v jednotlivých bodech.

2.3.1 Modelování profilů

V každém bodě navzorkujeme v obou směrech k hodnot pixelů ležících na normále hrany tvaru. Dostaneme tedy $2k+1$ vzorků z nichž vytvoříme vektor g. K potlačení vlivu osvětlení v obraze uvažujeme pouze změnu intenzity vůči bodu $k+1$. Zopakujeme tento proces pro každý bod v každém trénovacím obraze. Pro každý bod pak můžeme vypočíst střední hodnotu profilu $\overline{g}$ a kovarianci S_g. Kvalita nasazení nového profilu g_s, na model profilu se pak vypočte jako Mahalanobisova vzdálenost

$$f(g_s) = (g_s - \overline{g})^T S_g^{-1} (g_s - \overline{g}) \tag{2.18}$$

Během hledání navzorkujeme v aktuálním bodě m hodnot pixelů($m > k$). Pro každý bod $2(m-k)+1$ vypočteme kvalitu nasazení a místo kde dostaneme nejmenší hodnotu (největší shodu) zvolíme jako nový bod. Proces se opakuje, dokud nedostaneme lepší souřadnice pro každý bod tvaru. Poté aplikujeme jednu iteraci algoritmu uvedeného v sekci 2.1.4.

2.4 Active Appearance Model

Active Appearance model, dále jen AAM, vychází z ASM, ale navíc pracuje s informací o textuře objektu. V této sekci popíši algoritmus, který generuje obrázek co nejpodobnější cílovému obrázku. Předpokladem je opět rozumně zvolená startovací pozice.

2.4.1 Vyhledávání v AAM

K určení správnosti vyhledávání potřebujeme stanovit kritérium kvality nasazení modelu na skutečný tvar. Toto kritérium lze jednoduše definovat jako vektor rozdílů

$$\delta I = I_i - I_m \tag{2.19}$$

kde I_i je vektor intenzit obrazu a I_m je vektor intenzit textury vytvořené ze současných parametrů modelu. K nalezení nejlepší shody chceme minimalizovat velikost $\Delta = |\delta I|^2$ změnou parametrů modelu c. Vzhledem k velkému počtu parametrů se toto může zdát jako složitá úloha, ale lze si všimnout, že každý pokus o nasazení instance modelu do obrazu je v podstatě stejná optimalizační úloha. Natrénováním informace o tom jak se mají pro různé δI změnit parametry modelu c můžeme dostat časově efektivní algoritmus vyhledávání.

2.4.2 Natrénování úprav parametrů modelu

Rozdíl mezi námi vygenerovanou texturou a texturou nacházející se v obraze se dá vyjádřit jako

$$r(p) = g_s - g_m \tag{2.20}$$

kde p je vektor parametrů $p^T = (c^T|t^T)$. c je vektor parametrů kombinovaného modelu a t je vektor parametrů rotace, translace a míry. Taylorovým rozvojem prvního řádu rovnice 2.20 dostaneme

$$r(p + \delta p) = r(p) + \frac{\partial r}{\partial p}\delta p \tag{2.21}$$

Při hledání nejlepší pozice máme k dispozici residuum r. Chceme získat δp takové, aby

minimalizovalo výraz $|r(p+\delta p)|^2$. Položením rovnice 2.21 rovné nule dostaneme

$$\delta p = -Rr(p) \quad kde \quad R = (\frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p})^{-1} \frac{\partial r}{\partial p}^T \tag{2.22}$$

Při standartní optimalizaci by bylo nutné pokaždé přepočítat $\frac{\partial r}{\partial p}$. Nicméně předpokládáme, že trénovací data jsou normalizována a lze tedy tuto informaci natrénovat z jejich množiny. Při trénování systematicky odchylujeme jednotlivé parametry p od jejich známých optimálních hodnot a počítáme pro každou odchylku residuum r. Matici R pak vypočteme jako

$$R = -r(p)^{-1}\delta p \tag{2.23}$$

2.4.3 Řešení optimalizačního problému

Z počátečních parametrů c_0 vytvoříme na startovní pozici instanci modelu s texturou g_s

- Výpočet chybového vektoru $\delta g_0 = g_s - g_m$
- Výpočet chyby $E_0 = |\delta g_0|^2$
- Výpočet odchylky nových parametrů, $\delta c = R\delta g_0$
- Nastavení $k = 1$
- $c_1 = c_0 - k\delta c$
- Vytvoření instance modelu pomocí parametrů c_1, výpočet nového chybového vektoru δg_1 a chyby E_1
- Když je $E_1 < E_0$ přijmeme c_1 jako nové parametry modelu
- V jiném případě zkoušíme $k = 1.5,\ k = 0.5,\ k = 0.25$ atd

Kapitola 3

Popis algoritmu

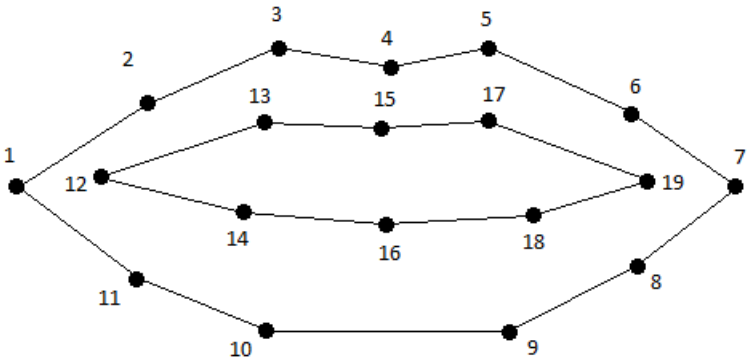
3.1 Trénovací data

Jako trénovací data jsem použil jednotlivé snímky nastříhané z videa nahraného speciálně pro tuto práci. Z každého obrazu je vybráno pouze okolí úst. Původní rozlišení snímků je 1920x1080 pixelů, rozlišení vybrané části je 385x302 pixelů. Příprava trénovacích dat zahrnovala také vhodnou volbu modelu reprezentujícího tvar rtů. Po otestování různých modelů jsem zvolil jako nejvhodnější model, který obsahuje devatenáct bodů. Model reprezentuje jak vnější tak i vnitřní tvar rtů. Jako trénovací množinu jsem vybral obrazy, jejichž počet je desetinásobek počtu bodů modelu. V tomto případě tedy tvoří trénovací množinu sto devadesát obrazů, ve kterých je přibližně stejné zastoupení všech možných tvarů rtů.

Obrázek 3.1: Příklad trénovacího obrazu

3.2 Trénování shape modelu

Při trénování modelu je nutné ručně zvolit jednotlivé body reprezentující tvar rtů. Tento proces se opakuje pro každý obrázek z trénovací množiny. Pořadí v jakém jsou body označeny jsem předem zvolil s ohledem na usnadnění označení vnitřních rtů. Jak jsem označoval jednotlivé body znázorňuje obrázek 3.2.



Obrázek 3.2: Pořadí v jakém byly označovány body

Výsledná data jsou pak reprezentována dvěma vektory x-ových a y-ových souřadnic jednotlivých bodů v obrazu. Aby měla statistická analýza smysl, je nutno data nejdříve normalizovat. Vybral jsem normalizaci, při které leží oba koutkové body modelu (1,7) na vodorovné ose a střed souřadné soustavy je ve středu této osy. Při pořizování trénovacích dat byla dodržována konstantní vzdálenost řečníka od kamery a není tedy nutné normalizovat data s ohledem na velikost. Normalizace probíhá tak, že nejdříve je střed souřadné soustavy umístěn do bodu jedna, tedy do levého koutku(bráno z pohledu na obraz). Souřadnice ostatních bodů jsou vypočteny vynásobením rotační maticí 3.1. Úhel natočení bodů se vypočte podle rovnice 3.2, kde y_2 je y-ová souřadnice pravého koutkového bodu a x_1 a x_2 jsou x-ové souřadnice levého a pravého koutkového bodu. Střed soustavy lze pak posunout odečtením poloviny vzdálenosti koutkových bodů v ose x od x-ových souřadnic všech bodů.

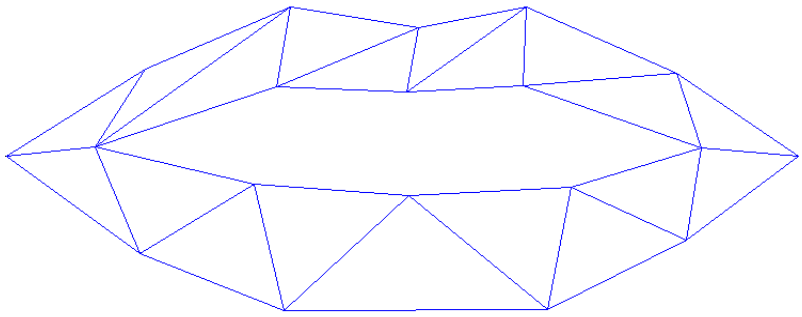
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.1}$$

$$\alpha = \tan^{-1} \frac{|y_2|}{|x_1 - x_2|} \tag{3.2}$$

Matici trénovacích dat vytvoříme spojením obou vektorů do jednoho sloupcového vektoru a připojením vektorů souřadnic z ostatních obrázků. Dostaneme matici, kde jeden řádek bude obsahovat všechny souřadnice (x nebo y) pro daný bod a každý sloupec bude náležet příslušnému obrazu. Velikost matice je v tomto případě 38x190. Na této matici poté provedeme PCA 2.1.2. Vypočteme kovarianční matici, její vlastní vektory a vlastní čísla. Z velikost vlastních čísel lze určit významnost jednotlivých vektorů pro zpětnou rekonstrukci dat. Díky těmto hodnotám můžeme volbou vhodného procentuálního počtu zredukovat množství vlastních vektorů reprezentujících daná data. Jako vhodnou hodnotu jsem zvolil 99% čímž došlo k redukci z třiceti osmy na dvacet sedm vlastních vektorů. Tato redukce je důležitá pro snížení výpočetních nároků výsledného programu. Z matice také vypočteme střední hodnotu souřadnic, kterou budeme dále potřebovat k vytváření modelu.

3.3 Trénování appearance modelu

Tvar reprezentující ústa se rozdělí na trojúhelníkové sektory vzájemným propojením jednotlivých bodů.



Obrázek 3.3: Rozdělení tvaru rtů na trojúhelníky

Určil jsem souřadnice jednotlivých pixelů ležících v každém z trojúhelníku metodou popsanou v sekci 2.2.1 a takto získanou texturu jsem nawarpoval do středního tvaru úst vypočteného z předchozího shape modelu. Tímto způsobem jsem získal pro každý obraz texturu nawarpovanou do středního tvaru a tedy se stejným počtem pixelů. Díky tomu, že tyto textury obsahují stejný počet pixelů, mohl jsem na nich dále provést PCA 2.1.2.Jelikož jsou data trojrozměrná (RGB), provedl jsem PCA pro každý kanál zvlášť. Výsledkem je model textury každého kanálu.

3.4 Kombinovaný model

Kombinovaný model vychází z předpokladu, že existuje závislost mezi tvarem úst a příslušnou texturou. Pro jeho výpočet jsem využil shape i appearance model. Informací, kterou je potřeba pro výpočet kombinovaného modelu získat, je zpětná projekce parametrů přes vlastní vektory modelů. Parametry shape modelu a appearance modelu získané zpětnou projekcí vložíme do jednoho vektoru. Pro každý obraz tedy získáme vektor parametrů obsahující informaci o obou modelech a na množině těchto parametrů opět provedeme PCA 2.1.2. Vzniká zde problém kombinace souřadnic a barvy pixelů. Jelikož parametry obou se mění různou rychlostí a v různých rozsazích, je nutné normalizovat parametry souřadnic shapu. Normalizace se dá velmi jednoduše provést vytvořením diagonální matice s hodnotami odmocniny z poměru celkové variance hodnot obou modelů. Tento postup lze také aplikovat na kombinovaný model všech tří barevných kanálů. Výpočtem kombinovaného modelu tedy dostaneme vlastní vektory, s jejichž pomocí jsme schopni generovat jednou sadou parametrů tvar i texturu.

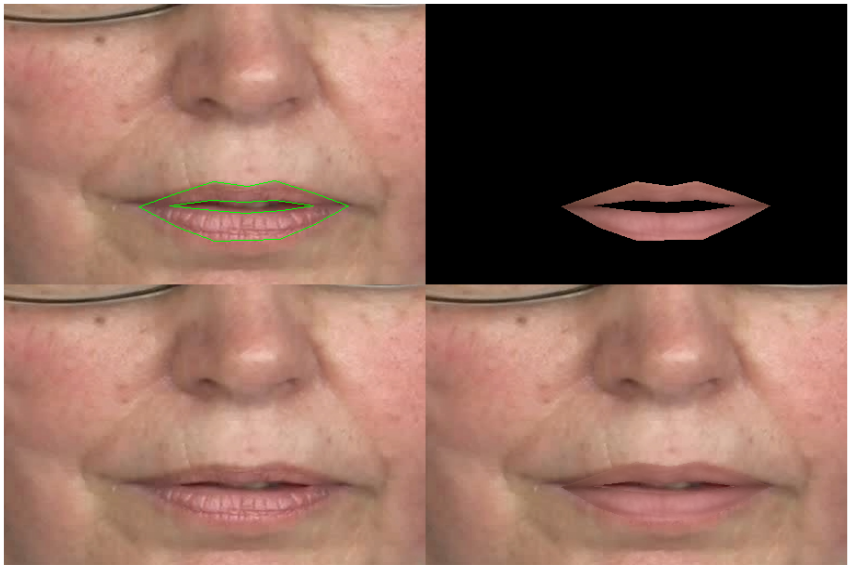
3.5 Generování instance modelu

Generování instance modelu probíhá podle rovnic 2.15 a 2.16. Sada parametrů, která navíc obsahuje posun po obou osách a rotaci se vynásobí s vlastními vektory kombinovaného modelu a vlastními vektory obou modelů. Výsledkem je sada souřadnic tvaru a textura ve středním tvaru, která se poté nawarpuje do vygenerovaných souřadnic tvaru. Před přidáním textury navíc k souřadnicím připočtu posuv po osách x a y a zohledním rotaci modelu. Výsledkem je instance kombinovaného modelu reprezentující tvar úst s příslušnou texturou.

Obrázek 3.4: Vygenerovaná instance modelu

3.6 Fitting

Pro zjednodušení budeme předpokládat, že počáteční bod vyhledávání je volen ručně. V tomto zvoleném bodě v obrazu, který by měl co nejvíce odpovídat středu úst, umístíme do obrazu vygenerovaný model. První model generujeme s nulovou hodnotou všech parametrů. Výsledný model tedy odpovídá střední hodnotě shapu a střední hodnotě textury. V místě, kde model překrývá obraz, odečteme stejným způsobem jako při trénování textury hodnoty jednotlivých pixelů. Míru chyby fittování poté určíme jako sumu rozdílu odečtené a vygenerované textury. Pokud není model shapu právě jeho střední hodnota musíme odečtenou texturu nejdříve do střední hodnoty shapu nawarpovat aby bylo možné jí s vygenerovanou texturou porovnat. K tomu, abychom věděli jak dále postupovat, potřebujeme natrénovat takovou informaci vedoucí při určité chybě k odpovídající korekci parametrů modelu, která povede ke zmenšení chyby. Pokud je tato informace dostupná, dostaneme nové hodnoty parametrů, z kterých vygenerujeme nový model. Součástí nových parametrů bude i informace o posunu po osách x a y a nová rotace modelu. Po výpočtu nového modelu opět odečteme odpovídající hodnoty překrytých pixelů a vypočteme míru chyby. Cyklus končí v případě kdy se chyba ve dvou po sobě jdoucích krocích nezmění.



Obrázek 3.5: Výstup algoritmu. Z pravého horního rohu: Nalezený tvar rtů, vygenerovaný tvar s texturou, původní obrázek, nasazení vygenerovaného tvaru do původního obrázku

3.7 Detekce rtů ve videozáznamu

V předchozí sekci jsem popsal vyhledávání v jednom obraze. Pro vyhledávání rtů ve videozáznamu jsem upravil algoritmus tak, že v prvním obraze videozáznamu je startovací bod zvolen ručně a ve všech dalších obrazech je startovací bod volen jako střední hodnota souřadnic nalezeného tvaru v předchozím obraze. Předpokládaný postup je aplikace nalezeného tvaru z předchozího obrazu na nový obraz. Zajímavým aspektem je, že výsledky hledání jsou mnohem přesnější pokud v každém obraze vyjdu ze střední hodnoty tvaru, tedy pokud jsou hodnoty všech parametrů rovny nule. Tento jev je velmi pravděpodobně způsobem nedostatečným natrénováním úprav parametrů. Jelikož jsem v době vypracování této práce neměl k dispozici stroj, který by dokázal v rozumně dlouhé době natrénovat úpravy pro všechny parametry kombinovaného modelu, nelze s jistotou určit čím je tento fenomén způsoben.

3.8 Trénování úprav parametrů

Při trénování úprav parametrů vycházím ze znalosti správných (promítnutých) parametrů modelu. Pro každý obrázek jsem provedl trénování takovým způsobem, že pro každý parametr udělám sérii změn v rámci jeho rozsahu (rozsah jsem volil podle Cootese jako polovinu směrodatné odchylky každého parametru) a sledoval jsem, jak tyto změny ovlivní výsledný model. Výsledkem je matice obsahující změny parametrů a matice obsahující odchylku modelu. Z rovnice 2.22 jsem vypočetl matici, která reprezentuje změnu parametrů modelu. Pokud tedy tuto vypočtenou matici přenásobíme vektorem chyby modelu, dostaneme požadovanou úpravu parametrů. Algoritmus trénování je velice paměťově náročný, jelikož jeden vektor s hodnotami pixelů textury obsahuje tisíce položek a počet vektorů je roven počtu obrazů vynásobeným počtem parametrů a počtem změn jednotlivých parametrů. S dostupnou výpočetní technikou bylo možno vypočítat data jen pro 66% vektorů. Navíc je diskretizace odchylek parametrů volena s velmi velkými kroky. Výsledná přesnost je tím pádem velmi ovlivněna kompletností takto vypočtených dat.

3.9 Natrénovaný model

Za použití výše popsaného algoritmu jsem vytvořil kombinovaný model vzhledu a tvaru, který je řízen pomocí 146 parametrů. Vliv parametrů na tvar a texturu modelu znázorním pomocí obrázku srovnávajících střední instanci modelu a krajní hodnoty parametrů. Jelikož je model kombinovaný, nelze parametry rozdělit na ty, které ovlivňují pouze tvar nebo pouze texturu.

Obrázek 3.6: Vliv parametrů na model je znázorněn srovnáním hodnot $-3\sqrt{\lambda}$, 0, $3\sqrt{\lambda}$, kde λ je příslušné vlastní číslo. První parametr ovlivňuje otevírání a zavírání obou rtů a současně ovlivňuje stahování a roztahování rtů do šířky. Druhý parametr řídí pouze otevírání a zavírání rtů. Další dva parametry otevírají a zavírají každý ret zvlášť. Na dalších parametrech je už patrný minimální vliv na tvar a liší se spíše texturou rtů.

Kapitola 4

Testování vlastností modelu

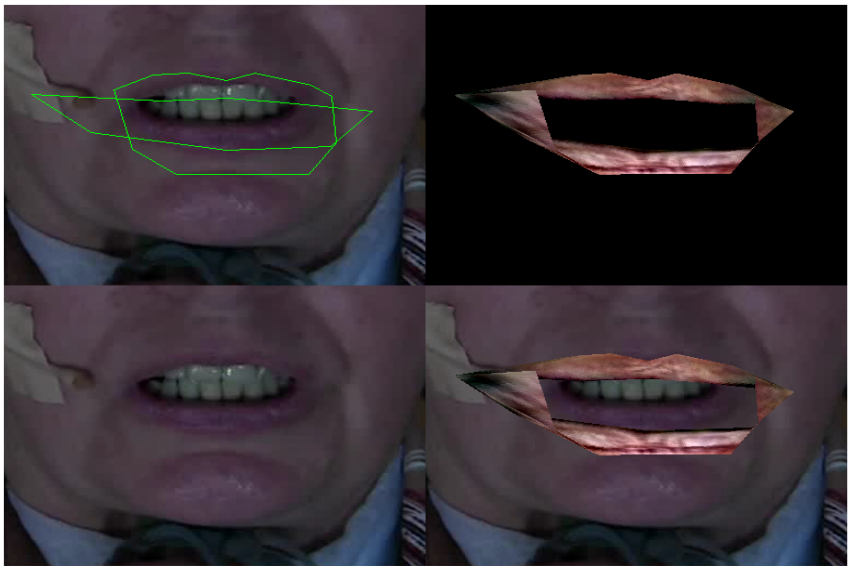
K otestování vlastností natrénovaného AAM jsem navrhnul několik experimentů. Otestuji přesnost jednotlivých bodů modelu, závislost přesnosti na množství použitých parametrů, vliv změny světelných podmínek testovacích dat, závislost přesnosti na množství trénovacích dat, změnu diskretizace při trénování úprav parametrů a úspěšnost detekce rtů ve videozáznamu.

4.1 Testování přesnosti

Pro testování přesnosti jsem připravil 100 náhodně vybraných obrázků z použitého videa. Tyto obrázky nejsou součástí trénovací množiny. Na obrázcích jsem ručně vyznačil významné body pro kontrolu přesnosti modelu. Test přesnosti probíhal porovnáním vygenerovaného tvaru s ručně označeným tvarem. Na texturu v tomto případě nebyl brán ohled. Startovací body pro nalezení objektu jsou voleny ručně. Výsledky experimentu jsou uvedeny v příloze A v tabulce A.1.

4.2 Závislost na světelných podmínkách videozáznamu

Jelikož jsem model natrénoval z jednoho videa s konkrétními světelnými podmínkami předpokládám, že úspěšnost vyhledávání objektu bude na těchto světelných podmínkách závislá. V jiném videozáznamu, který se bude lišit osvětlením nahrávané scény by mělo vyhledávání selhat. Tento test jsem provedl pomocí obrázků z jiného videa nahraného v jiné místnostni při zatažených roletách. Obraz je tedy výrazně tmavší.



Obrázek 4.1: Výstup algoritmu při jiných světelných podmínkách. Z pravého horního rohu: Nalezený tvar rtů, vygenerovaný tvar s texturou, původní obrázek, nasazení vygenerovaného tvaru do původního obrázku

Z výsledného obrázku 4.1 je jasně vidět, že algoritmus selhal. Z vygenerované textury se dá poznat, že i nejtmavší textura rtů jakou se algoritmus pokusil vygenerovat je stále mnohem světlejší než skutečné rty v obrázku. Algoritmus byl navíc zastaven po jedné iteraci, protože výsledky dalších iterací už nedávaly žádný smysl.

4.3 Změna diskretizace při trénování úprav parametrů

Původní diskretizaci pro trénování úprav parametrů jsem volil intuitivně na základě toho, že nebyl dostupný stroj, který by dokázal vypočíst kompletní matici s úpravou všech parametrů modelu. Volil jsem tedy takový přístup, kdy jsem na základě velikosti jednotlivých vlastních čísel určil jejich důležitost a tím pádem i míru diskretizace. Výsledkem je taková diskretizace, kde nejdůležitější parametr má na svém rozsahu 18 různých hodnot. Dalších pět parametrů má na rozsahu 10 hodnot a posledních 14 parametrů má na svém rozsahu hodnot 6. Tyto parametry respektují 66% rozptylu trénovacích dat. Pro otestování správnosti tohoto rozhodnutí jsem natrénoval ještě opačnou variantu, kde více hodnot připadá méně důležitým parametrům a méně hodnot těm více důležitým. Výsledky experimentu jsou uvedeny v příloze A v tabulce A.2. Z výsledků je patrné, že průměrná chyba je větší než u mnou zvolených diskretizací. Z toho vyvozuji, že mnou zvolený postup byl správný.

4.4 Změna rozsahu hodnot při trénování úprav parametrů

Tim Cootes uvádí ve svých poznámkách volbu rozsahu parametrů jako $0.5std(c_i)$, kde std je směrodatná odchylka a c_i je i-tý parametr modelu. Pro ověření správnosti této hodnoty jsem natrénoval i rozsahy $0.1std(c_i)$ a $0.2std(c_i)$. S těmito natrénovanými úpravami provedu testování přesnosti stejně jako v sekci 4.1. V rámci tohoto experimentu navíc vyzkouším natrénovat posuv po dvou pixelech namísto jednoho. Výsledky experimentu se změnou rozsahů jsou uvedeny v příloze A v tabulkách A.3 a A.4. Výsledky změny posuvu jsou v tabulce A.5. Stejně jako v předchozích experimentech se touto změnou zvětšila chyba nasazení tvaru do obrazu.

4.5 Detekce rtů ve videozáznamu

Jako poslední test jsem zvolil detekci rtů ve videozáznamu. Krátký videozáznam, který má 6s a skládá se ze 149 snímků, jsem vystřihl z nahraného videa. Hledání jsem odstartoval na první snímku ručně a dále už celý proces proběhl automaticky. Video s vykreslenými výsledky je přiloženo na DVD. U tohoto kroku jsem také provedl testování přesnosti, které probíhalo ručním označením významných bodů v každém třetím obrázku a následným vypočtením chyb. Na trénovaný model byl schopný sledovat tvar rtů v průběhu celého videozáznamu. I v případech, kdy došlo k vychýlení vygenerovaného tvaru, se algoritmus po několika snímcích dokázal opět úspěšně vrátit ke správnému tvaru rtů.

Obrázek 4.2: Ukázka z detekce rtů ve videozáznamu

Kapitola 5

Závěr

V této práci jsem vytvořil fungující algoritmus schopný sledovat tvar rtů ve videozáznamu. Model je uvažován pro individuálního řečníka a je tedy nutné ho pro každého člověka natrénovat znovu. Zpracování diplomové práce probíhalo nejdříve natočením videa, které posloužilo jako zdroj trénovacích dat. Na počátku jsem vytvořil vlastní implementaci algoritmu Active shape model a otestoval jsem její spolehlivost na zdrojovém videu. Jelikož se tato metoda projevila jako nedostatečně přesná a velmi náchylná k selhání, rozhodl jsem se Active shape model rozšířit na Active appearance model. Model tvaru je v AAM stejný jako v ASM, takže bylo možné využít již dříve natrénovaná data a pokračovat s trénováním texury rtů. Po natrénování obou modelů a vytvoření kombinovaného modelu přišel na řadu problém nasazení vygenerovaného tvaru do prohledávaného obrazu a zajištění konvergence iteračního algoritmu upravujícího tvar tak, aby co nejlépe odpovídal tvaru hledaného objektu (rtům). Při řešeního tohoto problému jsem narazil na nedostatečnou kapacitu paměti dnešní výpočetní techniky a byl jsem proto schopný natrénovat model respektující pouze 66% rozptylu dat. Nicméně i tento model se projevil jako dostačující, což dokazuje video se znázorněným trackingem přiložené na DVD. V tomto kroce jsem se musel rozhodnou jakou diskretizaci volit při trénování úprav parametrů. Vzhledem k tomu, že nebylo možné v přijatelně dlouhé době natrénovat úpravy pro všechny parametry, musel jsem zvolit diskretizaci a množství parametrů podle vlastního uvážení. Svou volbu jsem poté ověřil sadou experimentů. Výsledky těchto experimentů prokázaly, že moje rozhodnutí upřednostnit parametry s vyšším vlastním číslem a úpravy těchto parametrů trénovat s větší diskretizací na jejich rozsahu, bylo správné. Pro otestování kvality algoritmu jsem navrhl několik experimentů. Průměrná chyba nasazení u mnou zvolené diskretizace je 5.8 pixelů, což je vzhledem k rozlišení obrázků přijatelná chyba.

Funkční algoritmus, který je přiložen na DVD, byl zpracován v prostředí Matlab. Tento algoritmus není momentálně schopný zpracovávat video v reálném čase. Výpočetní náročnost jednoho snímku videa s rozlišením 1920x1080 pixelů je v rozsahu jedné až tří sekund podle množství iterací algoritmu. Při testování na trénovacích snímcích s rozlišením 385x302 je časové náročnost výpočtu maximálně jedna sekunda. Při konverzi algoritmu do jiného programovacího jazyka a následné optimalizaci algoritmu lze tedy předpokládat, že se časy potřebné pro výpočet jednoho snímku přiblíží možnosti aplikace v reálném čase.

Při pokračování této práce bych se zaměřil na schopnost algoritmu vyrovnat se s rozdílnými světelnými podmínkami videa. Lze uvažovat o třech možných řešeních tohoto problému. První možností by bylo normalizovat obrázek, ve kterém hledáme objekt tak, aby střední hodnota jasu odpovídala střední hodnotě jasu trénovacích obrázků. Další možností je zahrnout jas do trénovacích dat modelu, což by velmi pravděpodobně vedlo k vytvoření parametrů, které by ovládali právě jas textury. Třetí možností je zahrnutí jasu jako vedlejšího parametru k rotaci a translaci. Dále lze ještě zauvažovat nad možností jiné reprezentace barevných dat. V této práci byl použit barevný systém RGB, existují ale i jiné reprezentace, které oddělují barevnou informaci od jasové složky. Je tedy možné, že použitím jiné reprezentace by bylo možné dosáhnout jasově neutrálního modelu. Na úplný závěr bych dodal, že je také potřeba vyřešit otázku automatického startování vyhledávání, kde nebude nutné ručně volit startovací bod v prohledávaném obrázku. Těmto bodů bych se rád věnoval během dalšího studia.

Literatura

[1] Cootes, T. F. and Taylor, C. J.. Statistical Models of Appearance for Computer Vision, Manchester, 2004.

[2] Siew Wen Chin, Kah Phooi Seng and Li-Minn Ang. Enhanced snake model and modified H∞ for lips contour detection and tracking, International Conference on Computer Applications and Industrial Electronics, 2010

[3] Matthias, S.. Approaches to analyse and interpret biological profile data, POTSDAM UNIVERSITY, 2006, Potsdam, Germany

[4] Varcheie, P.D.Z. and Gagnon, L.. Lip tracking using adaptive fuzzy particle filter in the context of car driving simulator under low contrast near-infrared illumination, International Conference on Acoustics Speech and Signal Processing, 2010

[5] Moghaddam, M.K. and Safabakhsh, R.. TASOM-based lip tracking using the color and geometry of the face, Fourth International Conference on Machine Learning and Applications, 2005

[6] Císař, P.. Využití metod odezírání ze rtů pro podporu rozpoznávání řeči, Plzeň, 2006

Příloha A

Tabulky

V této příloze jsou uvedeny tabulky s výsledky jednotlivých experimentů. Jednotlivé experimenty jsou popsány v kapitole 4. V každé tabulce jsou popsány následující vlastnosti jednotlivých bodů reprezentující tvar rtů: $Cov(x, y)$ je kovariance mezi x-ovou a y-ovou složkou příslušného bodu. $\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$ je střední hodnota Euklidovské vzdálenosti, Δx a Δy jsou rozdíly vygenerovaných a ručně označených bodů. σ je směrodatná odchylka Euklidovských vzdáleností z předchozího řádku.

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 26.7247 & -4.2468 \\ -4.2468 & 15.9725 \end{bmatrix}$	$\begin{bmatrix} 31.6524 & -8.8149 \\ -8.8149 & 11.9812 \end{bmatrix}$	$\begin{bmatrix} 19.6586 & -0.4050 \\ -0.4050 & 10.3250 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.2482	6.4591	5.1529
σ	3.6740	3.3731	2.7391
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 9.2037 & 0.0388 \\ 0.0388 & 10.2130 \end{bmatrix}$	$\begin{bmatrix} 16.2765 & 0.8702 \\ 0.8702 & 9.0079 \end{bmatrix}$	$\begin{bmatrix} 25.1449 & 9.0317 \\ 9.0317 & 11.5270 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.2227	4.8040	5.3022
σ	2.1718	2.7168	3.0014
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 23.6348 & -1.0613 \\ -1.0613 & 10.4716 \end{bmatrix}$	$\begin{bmatrix} 45.2732 & -18.3989 \\ -18.3989 & 18.6155 \end{bmatrix}$	$\begin{bmatrix} 44.4187 & -10.5979 \\ -10.5979 & 22.8161 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.5833	6.7999	7.3421
σ	3.4109	4.3585	4.5069
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 40.6846 & 4.8949 \\ 4.8949 & 19.9328 \end{bmatrix}$	$\begin{bmatrix} 38.5825 & 9.6028 \\ 9.6028 & 14.4049 \end{bmatrix}$	$\begin{bmatrix} 52.8570 & 3.7318 \\ 3.7318 & 9.7808 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.9881	6.3501	7.1101
σ	4.5522	3.8868	5.0542
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 28.4142 & -6.1140 \\ -6.1140 & 12.2903 \end{bmatrix}$	$\begin{bmatrix} 60.3500 & 4.6811 \\ 4.6811 & 9.9636 \end{bmatrix}$	$\begin{bmatrix} 7.4621 & -1.6119 \\ -1.6119 & 11.3216 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.3829	6.9149	3.8078
σ	3.6130	4.8106	2.2773
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 8.8119 & -0.6117 \\ -0.6117 & 11.0138 \end{bmatrix}$	$\begin{bmatrix} 28.4088 & 6.9130 \\ 6.9130 & 13.1761 \end{bmatrix}$	$\begin{bmatrix} 49.5996 & 1.4293 \\ 1.4293 & 8.5990 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	3.8965	5.6496	6.8245
σ	2.1779	3.2772	3.9698
Body	19.		
Cov(x,y)	$\begin{bmatrix} 47.2485 & 0.9642 \\ 0.9642 & 6.5552 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.6547		
σ	3.8733		

Tabulka A.1: Výsledky testování přesnosti

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 31.5266 & -3.4813 \\ -3.4813 & 15.4896 \end{bmatrix}$	$\begin{bmatrix} 32.585 & -9.4762 \\ -9.4762 & 12.9068 \end{bmatrix}$	$\begin{bmatrix} 17.3889 & -2.0604 \\ -2.0604 & 16.4918 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.4493	6.4186	5.6819
σ	4.1603	3.6244	3.1847
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 8.9804 & 1.1037 \\ 1.1037 & 15.3497 \end{bmatrix}$	$\begin{bmatrix} 15.1433 & 2.4102 \\ 2.4102 & 13.5766 \end{bmatrix}$	$\begin{bmatrix} 24.9738 & 8.1129 \\ 8.1129 & 11.2404 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.8282	5.4592	5.4415
σ	2.7708	3.0262	2.9067
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 26.7266 & -0.4300 \\ -0.4300 & 12.6189 \end{bmatrix}$	$\begin{bmatrix} 43.6859 & -16.0124 \\ -16.0124 & 18.2581 \end{bmatrix}$	$\begin{bmatrix} 43.1931 & -10.4066 \\ -10.4066 & 26.9614 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.1294	6.8342	7.6614
σ	4.0825	4.4278	4.3861
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 37.7762 & 6.3666 \\ 6.3666 & 24.6923 \end{bmatrix}$	$\begin{bmatrix} 38.1391 & 9.2126 \\ 9.2126 & 15.5791 \end{bmatrix}$	$\begin{bmatrix} 56.8322 & 0.983 \\ 0.9830 & 8.9100 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.4851	6.7908	7.2813
σ	4.6637	3.933	5.4362
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 30.2038 & -10.0645 \\ -10.0645 & 19.3821 \end{bmatrix}$	$\begin{bmatrix} 55.8497 & 2.4800 \\ 2.4800 & 11.1444 \end{bmatrix}$	$\begin{bmatrix} 6.7588 & -0.7457 \\ -0.7457 & 14.6855 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.9939	6.9116	3.9676
σ	4.3205	4.4223	2.8732
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 7.9744 & -0.6540 \\ -0.6540 & 13.9774 \end{bmatrix}$	$\begin{bmatrix} 27.2996 & 6.9608 \\ 6.9608 & 16.8291 \end{bmatrix}$	$\begin{bmatrix} 48.8028 & 2.2658 \\ 2.2658 & 10.0074 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.1563	5.7715	7.3998
σ	2.5198	3.5141	4.3317
Body	19.		
Cov(x,y)	$\begin{bmatrix} 47.7038 & -0.0503 \\ -0.0503 & 7.8791 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.4623		
σ	4.4810		

Tabulka A.2: Výsledky testování přesnosti s opačnou diskretizací

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 51.2425 & -0.5312 \\ -0.5312 & 15.9131 \end{bmatrix}$	$\begin{bmatrix} 44.8806 & -1.2722 \\ -1.2722 & 18.2253 \end{bmatrix}$	$\begin{bmatrix} 19.2406 & 2.6190 \\ 2.6190 & 29.2360 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	8.0880	7.4754	6.8844
σ	5.0112	4.2634	3.3804
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 9.3244 & 2.7393 \\ 2.7393 & 33.0225 \end{bmatrix}$	$\begin{bmatrix} 20.6045 & -2.2713 \\ -2.2713 & 32.5691 \end{bmatrix}$	$\begin{bmatrix} 35.8005 & 2.9622 \\ 2.9622 & 20.2387 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.2390	7.2529	6.7859
σ	3.3870	3.74	3.6253
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 42.0264 & 1.1630 \\ 1.1630 & 17.8881 \end{bmatrix}$	$\begin{bmatrix} 71.4704 & -0.8900 \\ -0.8900 & 27.5771 \end{bmatrix}$	$\begin{bmatrix} 55.5701 & 7.6898 \\ 7.6898 & 52.0750 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.6521	8.9008	9.7859
σ	5.0010	5.4834	5.0685
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 45.3673 & -3.2964 \\ -3.2964 & 54.9962 \end{bmatrix}$	$\begin{bmatrix} 45.3673 & -3.2964 \\ -3.2964 & 54.9962 \end{bmatrix}$	$\begin{bmatrix} 45.3673 & -3.2964 \\ -3.2964 & 54.9962 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	9.1528	8.7832	9.4244
σ	5.2676	5.2401	7.7083
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 31.4434 & -1.9867 \\ -1.9867 & 43.6421 \end{bmatrix}$	$\begin{bmatrix} 96.4188 & -24.2874 \\ -24.2874 & 40.0052 \end{bmatrix}$	$\begin{bmatrix} 6.9990 & -2.6099 \\ -2.6099 & 43.8453 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.5503	9.9191	6.2175
σ	5.1175	6.2324	4.0728
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 8.7093 & 1.6357 \\ 1.6357 & 43.8417 \end{bmatrix}$	$\begin{bmatrix} 29.407 & 8.6886 \\ 8.6886 & 48.3235 \end{bmatrix}$	$\begin{bmatrix} 29.407 & 8.6886 \\ 8.6886 & 48.3235 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.0930	7.7294	9.7995
σ	4.1093	4.5391	6.0470
Body	19.		
Cov(x,y)	$\begin{bmatrix} 86.7934 & 6.2321 \\ 6.2321 & 9.341 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	8.0947		
σ	6.9762		

Tabulka A.3: Výsledky testování přesnosti s rozsahem $0.1std$

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 37.2519 & -5.3463 \\ -5.3463 & 16.0916 \end{bmatrix}$	$\begin{bmatrix} 37.7671 & -7.1654 \\ -7.1654 & 13.0275 \end{bmatrix}$	$\begin{bmatrix} 18.2101 & -1.2395 \\ -1.2395 & 19.5481 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.2850	6.8161	6.0036
σ	4.3161	3.9392	3.1238
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 9.3843 & 1.6067 \\ 1.6067 & 21.8365 \end{bmatrix}$	$\begin{bmatrix} 17.6027 & 0.7511 \\ 0.7511 & 21.0238 \end{bmatrix}$	$\begin{bmatrix} 29.1702 & 6.6442 \\ 6.6442 & 15.5031 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.2936	6.1583	6.1062
σ	2.9654	3.4197	3.1160
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 29.8096 & 1.5173 \\ 1.5173 & 17.5104 \end{bmatrix}$	$\begin{bmatrix} 48.7178 & -12.2432 \\ -12.2432 & 20.4365 \end{bmatrix}$	$\begin{bmatrix} 45.5622 & -3.4287 \\ -3.4287 & 33.6360 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.0105	7.4997	8.2155
σ	4.4593	4.5480	4.6209
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 40.8413 & 4.1985 \\ 4.1985 & 37.1912 \end{bmatrix}$	$\begin{bmatrix} 47.0316 & 2.8054 \\ 2.8054 & 24.4815 \end{bmatrix}$	$\begin{bmatrix} 64.903 & -1.7011 \\ -1.7011 & 9.3883 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	8.0333	7.4838	8.1651
σ	4.9660	4.1982	6.3600
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 29.5219 & -6.6556 \\ -6.6556 & 26.2045 \end{bmatrix}$	$\begin{bmatrix} 67.718 & -8.2184 \\ -8.2184 & 23.4954 \end{bmatrix}$	$\begin{bmatrix} 7.0337 & -1.5740 \\ -1.5740 & 24.7487 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.3659	8.1412	4.7073
σ	4.6013	4.9579	3.3627
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 8.2327 & 0.7726 \\ 0.7726 & 25.0735 \end{bmatrix}$	$\begin{bmatrix} 27.2144 & 7.6829 \\ 7.6829 & 28.9719 \end{bmatrix}$	$\begin{bmatrix} 67.1815 & 13.9134 \\ 13.9134 & 18.3308 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.8548	6.2736	8.4176
σ	3.2027	4.1868	4.9098
Body	19.		
Cov(x,y)	$\begin{bmatrix} 53.7704 & 4.5449 \\ 4.5449 & 9.5703 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.1219		
σ	5.3165		

Tabulka A.4: Výsledky testování přesnosti s rozsahem $0.2std$

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 38.1726 & -4.0311 \\ -4.0311 & 15.043 \end{bmatrix}$	$\begin{bmatrix} 35.7439 & -8.7543 \\ -8.7543 & 11.4368 \end{bmatrix}$	$\begin{bmatrix} 18.4577 & -0.8086 \\ -0.8086 & 13.8705 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.0071	6.5376	5.5618
σ	4.3273	4.1406	2.7365
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 8.7394 & 0.8188 \\ 0.8188 & 15.5585 \end{bmatrix}$	$\begin{bmatrix} 16.1701 & -0.2875 \\ -0.2875 & 15.4648 \end{bmatrix}$	$\begin{bmatrix} 27.7207 & 7.2776 \\ 7.2776 & 13.2200 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.6432	5.3807	5.7491
σ	2.3843	2.8758	3.0485
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 31.8297 & 0.2027 \\ 0.2027 & 14.6985 \end{bmatrix}$	$\begin{bmatrix} 50.5184 & -13.0162 \\ -13.0162 & 19.4965 \end{bmatrix}$	$\begin{bmatrix} 46.5398 & -4.0834 \\ -4.0834 & 29.7120 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.8723	7.3753	7.9866
σ	4.3107	4.7634	4.9203
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 41.8596 & 5.0984 \\ 5.0984 & 30.4589 \end{bmatrix}$	$\begin{bmatrix} 45.2304 & 4.9508 \\ 4.9508 & 20.4877 \end{bmatrix}$	$\begin{bmatrix} 69.6173 & -0.9239 \\ -0.9239 & 7.8228 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.8257	7.2030	8.1170
σ	4.9584	4.1146	6.2517
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 29.0789 & -5.9906 \\ -5.9906 & 19.8745 \end{bmatrix}$	$\begin{bmatrix} 62.2979 & -3.1358 \\ -3.1358 & 18.4501 \end{bmatrix}$	$\begin{bmatrix} 6.8271 & -1.8266 \\ -1.8266 & 19.9151 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	6.0132	7.7273	4.2582
σ	4.0279	4.5835	3.0310
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 7.9743 & 0.0692 \\ 0.0692 & 19.7182 \end{bmatrix}$	$\begin{bmatrix} 26.674 & 7.5989 \\ 7.5989 & 23.5094 \end{bmatrix}$	$\begin{bmatrix} 62.2994 & 9.3094 \\ 9.3094 & 14.515 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.4299	5.9345	7.9669
σ	2.8205	3.8419	4.7594
Body	19.		
Cov(x,y)	$\begin{bmatrix} 54.1715 & 4.1867 \\ 4.1867 & 9.4156 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	7.4443		
σ	4.8826		

Tabulka A.5: Výsledky testování přesnosti s posuvem o $2px$

Body	1.	2.	3.
Cov(x,y)	$\begin{bmatrix} 21.3462 & 0.0416 \\ 0.0416 & 7.2590 \end{bmatrix}$	$\begin{bmatrix} 22.1308 & -5.8446 \\ -5.8446 & 8.5864 \end{bmatrix}$	$\begin{bmatrix} 19.0657 & -2.7627 \\ -2.7627 & 7.5668 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.9692	6.2977	5.7826
σ	2.7381	3.5514	3.1289
Body	4.	5.	6.
Cov(x,y)	$\begin{bmatrix} 7.2846 & -2.9432 \\ -2.9432 & 6.5350 \end{bmatrix}$	$\begin{bmatrix} 7.1713 & -1.5200 \\ -1.5200 & 6.9362 \end{bmatrix}$	$\begin{bmatrix} 25.1139 & 5.6212 \\ 5.6212 & 8.4426 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	3.403	3.2421	5.9077
σ	2.1582	2.0024	3.5880
Body	7.	8.	9.
Cov(x,y)	$\begin{bmatrix} 26.7188 & 0.6318 \\ 0.6318 & 4.1718 \end{bmatrix}$	$\begin{bmatrix} 32.5128 & -8.3144 \\ -8.3144 & 12.4263 \end{bmatrix}$	$\begin{bmatrix} 32.5742 & -11.6878 \\ -11.6878 & 21.4067 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	4.4551	6.119	6.0904
σ	3.3862	4.438	5.0032
Body	10.	11.	12.
Cov(x,y)	$\begin{bmatrix} 60.4726 & -2.9162 \\ -2.9162 & 19.9695 \end{bmatrix}$	$\begin{bmatrix} 39.0714 & 4.1114 \\ 4.1114 & 11.925 \end{bmatrix}$	$\begin{bmatrix} 54.1822 & -1.5789 \\ -1.5789 & 3.5688 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	9.0581	8.1758	5.5253
σ	6.5895	4.7231	5.3422
Body	13.	14.	15.
Cov(x,y)	$\begin{bmatrix} 25.2583 & -8.8876 \\ -8.8876 & 16.6334 \end{bmatrix}$	$\begin{bmatrix} 104.409 & -1.0858 \\ -1.0858 & 13.717 \end{bmatrix}$	$\begin{bmatrix} 6.0036 & -2.7523 \\ -2.7523 & 14.1135 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.4242	11.8265	3.5246
σ	4.0762	6.9685	2.7776
Body	16.	17.	18.
Cov(x,y)	$\begin{bmatrix} 5.3956 & 0.0918 \\ 0.0918 & 15.5727 \end{bmatrix}$	$\begin{bmatrix} 21.3513 & 2.8232 \\ 2.8232 & 13.8784 \end{bmatrix}$	$\begin{bmatrix} 42.1739 & 0.4581 \\ 0.4581 & 8.7038 \end{bmatrix}$
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	3.3742	5.2018	8.1017
σ	3.2103	2.7943	5.4111
Body	19.		
Cov(x,y)	$\begin{bmatrix} 40.8314 & 0.1548 \\ 0.1548 & 2.7513 \end{bmatrix}$		
$\mathrm{E}\left\{\sqrt{\Delta x^2 + \Delta y^2}\right\}$	5.2823		
σ	4.1167		

Tabulka A.6: Výsledky testování přesnosti ve videosekvenci