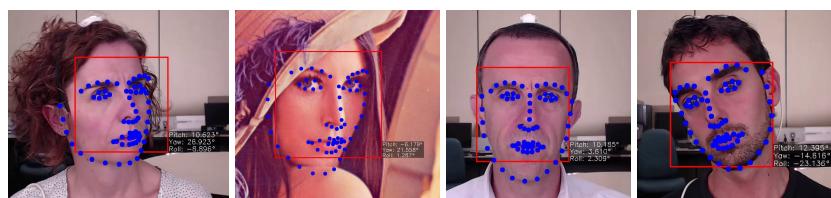


Určení azimutu natočení hlavy v záznamu bezpečnostní kamerou

Ondřej Blucha*



Abstrakt

Cílem této práce je vytvořit program, který bude schopen ze vstupního videa určit úhly natočení hlavy vůči kameře. Samotné určení natočení hlavy se skládá ze tří samostatných částí, a to z detekce obličeje, určení charakteristických bodů v obličeji a nakonec určení samotného natočení hlavy na základě detekovaných charakteristických bodů. Detekce obličeje je založená na algoritmu Viola–Jones. Charakteristické body jsou detekovány pomocí metody Active Appearance Model. Výsledné natočení hlavy je určeno na základě antropometrických vlastností hlavy.

Klíčová slova: Detekce obličejů — Charakteristické body v obličeji — Určení úhlu natočení hlavy

Přiložené materiály: [Demonstrační video](#)

*xbluch00@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

V současné době máme bezpečnostní kamery již téměř na každém rohu. Také s detekcí osob a obličejů se můžeme na dnešních kamerách běžně setkat. Ovšem určením úhlu natočení hlavy vůči kameře zatím běžnou součástí zabezpečovacích zatím není. Přínos práce může být například při strojovém rozpoznávání osob.

1.1 Existující řešení

Existující řešení lze rozdělit na dvě skupiny podle toho zda pracují s daty ze standartní kamery nebo pracují s hloubkovými daty. Algoritmus, který pracuje se standartními daty, je obsažen například v knihovně OpenFace [1]. S hloubkovými daty pracuje algoritmus *Real Time Head Pose Estimation with Random Regression Forests* [2]. Tato práce je zaměřená pouze na práci s daty ze standartní kamery (RGB data).

2. Návrh

Určení natočení hlavy lze rozdělit na tři nezávislé části. V první řadě je potřeba detektovat obličeje, dále je potřeba v každém obličeji detektovat charakteristické body a nakonec se na základě detekovaných bodů určí výsledné natočení hlavy pro všechny tři osy.

2.1 Detekce obličejů

Problém detekce obličejů je již znám a bylo již navrženo mnoho různých algoritmů. Algoritmy určené k detekci obličejů (respektive čehokoliv jiného) lze rozdělit podle toho, jestli používají ručně vytvořený klasifikátor nebo vyučovaný klasifikátor. Z algoritmů používajících vyučovaný klasifikátor se dnes používají zejména detektor Viola-Jones [3] a detektor založený na histogramech orientovaných gradientů (HOG) [4], a to především proto, že jsou schopny detektovat s velmi vysokou úspěšností a dokážou pracovat v reálném čase, což je pro detekci z videa nezbytné.

2.2 Detekce charakteristických bodů v obličeji

K zachycení rysů v obličeji je vhodné použít metody fungující na principu aktivní šablony. Tyto metody slouží k tomu, aby na obličeji (popřípadě jakýkoliv jiný objekt) přesně namapovaly šablonu skládající se z bodů, které si v různých obličejích významově odpovídají. Může se jednat o koutek levého a pravého oka, špičku nosu atd.

Rozlišujeme metodu ASM (*Active Shape Model*) [5] a AAM (*Active Appearance Model*) [5]. Šablonu pro metodu ASM se skládá z bodů a hran mezi nimi, kdežto šablonu pro metodu AAM se skládá z bodů a hran, ale i z informace o textuře uvnitř šablony.

K tomu, aby bylo možné šablonu použít, je potřeba ji nejdříve vytvořit, a to ze sady trénovacích obrázků s anotovanými body. K vytvoření šablony pro metodu ASM se používá metoda *Statistical Shape Models* [5] a k vytvoření šablony pro metodu AAM se používá metoda *Statistical Models of Appearance* [5].

Při detekci charakteristických bodů v obličeji je velmi důležitá vhodná volba charakteristických bodů. Vhodně zvolené body by se měly nacházet na všech obličejích a měly by se na každém obličeji nacházet na podobném místě. Také by se měly ideálně nacházet v ostrých rozích, v místech střetu dvou hran nebo snadně biologicky lokalizovatelných bodech, jako například střed oka, špička nosu atd. Takovýchozí bodů bývá většinou málo, proto se ještě přidávají body, které se nacházejí na hranách mezi dvěma již zvolenými body.

Ukázkový obličej s detekovanými 68 charakteristickými body je zobrazen na obrázku 1.

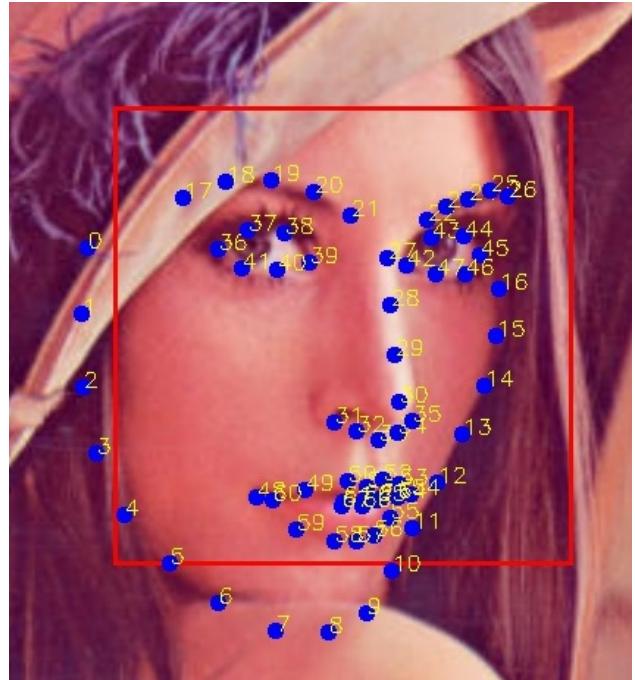
2.3 Určení natočení hlavy

Určení natočení hlavy vůči kameře je potřeba provádět pro všechny tři osy zvlášť. Nezávisle na sobě se provede detekce natočení hlavy v ose x (anglicky *pitch*), v ose y (anglicky *yaw*) a v ose z (anglicky *roll*). Model hlavy se znázorněním jednotlivých os lze vidět na obrázku 2.

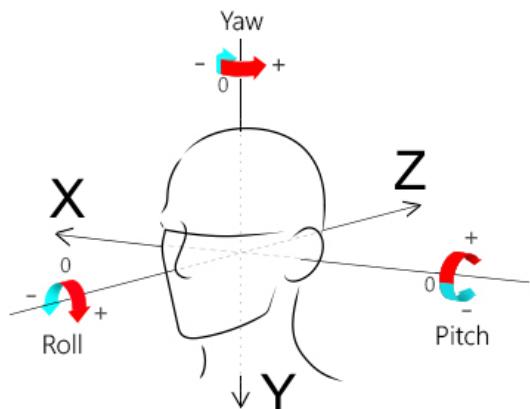
Samotné určení natočení hlavy bude vycházet z již předem detekovaných charakteristických bodů, které jsou vidět na obrázku 1. Všechny následující výpočty budou předpokládat takto očíslované body.

2.3.1 Určení natočení hlavy v ose z (roll)

K detekci natočení hlavy v ose z je nevhodnější použít dva body na obličeji nacházející se ve stejné výšce, pokud je natočení hlavy v ose z nulové. Jako nevhodnější se zdají být body levého a pravého vnějšího koutku oka (body 36 a 45 z obrázku 1). Jako další vhodné body by mohly být zvoleny levý a pravý koutek úst, ovšem oči jsou nejméně ovlivněny mimikou obličeje a je u nich malá pravděpodobnost chybné detekce.



Obrázek 1. Obličej s detekovanými 68 charakteristickými body



Obrázek 2. Model hlavy s třemi vyznačenými osami

Pokud budou použity body 36 a 45, pak výsledný úhel natočení γ se spočte pomocí arkus tangens takto:

$$\gamma = \tan^{-1} \frac{y_{45} - y_{36}}{x_{46} - x_{36}} \quad (1)$$

kde x_{45} je x -ová souřadnice bodu 45 a y_{45} je y -ová souřadnice bodu 45.

Pokud budeme předpokládat počátek souřadnicového systému obrázku (tedy bod o souřadnicích $[0, 0]$) vlevo nahoře, jak je tomu v OpenCV, pak pokud vyjde úhel γ kladný, tak je obličej natočený po směru hodinových ručiček z pohledu pozorovatele, a pokud vyjde záporný, tak je natočený proti směru hodinových ručiček z pohledu pozorovatele.

2.3.2 Určení natočení hlavy v ose y (yaw)

Další v pořadí je detekce natočení hlavy v ose y. Jako body vhodné k této detekci lze vybrat bod ležící pod

špičkou nosu (bod 30) a dva body u ležící na obrysу obličeje ve stejné výšce (body 2 a 14). Nejprve je potřeba si spočítat jaký tvar má průměrná hlava. Pro výpočet průměrného tvaru hlavy byly použity tabulka z knihy *Human engineering design data* [6]. Při využití těchto dat jsem vždy použil hodnotu mediánu pro dané měření (v tabulce je to hodnota 50. percentilu). A jelikož jsou v tabulce zvlášť hodnoty pro muže a ženy, tak výslednou hodnotu určím jako průměr mužského a ženského mediánu.

Dle tabulky 6 je mediánová vzdálenost mezi ušima 13,9 cm. Dle tabulky 12 je mediánová vzdálenost mezi nosem a nejvzdálenějším místem vzadu na hlavě 19,55 cm. Pokud tedy budeme považovat, že polovina této vzdálenosti je střed hlavy, pak od středu hlavy k nosu to je 9,775 cm. Ve skutečnosti nás nezajímají přesné rozměry hlavy, ale poměr vzdálenosti mezi ušima ke vzdálenosti od středu hlavy ke špičce nosu. Tento poměr si zaznamenáme jako konstantu k a vypočítáme ji takto:

$$k = \frac{9,55}{13,9} = 0,703 \quad (2)$$

K určení natočení hlavy v ose y je potřeba zjistit jak moc se bod nacházející se pod špičkou nosu vychyluje od středu hlavy. Nejprve je potřeba přesunout bod nacházející se pod špičkou nosu na přímku, která je tvořena dvěma body na obrysu obličeje. K tomu je potřeba si spočítat přímku mezi dvěma body na obrysu obličeje (body 2 a 14). Přímka bude spočítaná ve směrnicovém tvaru ($y = kx + q$). Směrnice přímky se spočítá:

$$k_1 = \frac{y_{14} - y_2}{x_{14} - x_2} \quad (3)$$

Pokud známe směrnici přímky a alespoň jeden bod na přímce, pak lze již jednoduše spočítat koeficient q . Pro výpočet koeficientu q_1 použijeme např. bod 2, pak rovnice bude vypadat takto:

$$q_1 = y_2 - k_1 x_2 \quad (4)$$

Následně je potřeba spočítat rovnici přímky, která je kolmá k předchozí přímce a zároveň prochází bodem nacházející se pod špičkou nosu (bod 30). Směrnici nové přímky spočítáme jako převrácenou hodnotu směrnice první přímky s opačným znaménkem takto:

$$k_2 = \frac{-1}{k_1} \quad (5)$$

Koeficient q_2 spočteme obdobně jako v předchozím případě:

$$q_2 = y_{30} - k_2 x_{30} \quad (6)$$

Potom je na řadě spočítat průsečík obou přímek. X -ová souřadnice nově vzniklého bodu se spočte takto:

$$x_p = \frac{q_2 - q_1}{k_1 - k_2} \quad (7)$$

Y -ová souřadnice se pak můžeme spočítat z první nebo druhé přímky, pro první přímku bude vypadat rovnice takto:

$$y_p = k_1 x_p + q_1 \quad (8)$$

V tuto chvíli již máme všechny tři body na jedné přímce. Nyní je na řadě zjistit jak moc se bod ležící pod špičkou nosu blíží k levému nebo pravému kraji obličeje a na základě znalostí o antropometrii hlavy pak určit pod jakým úhlem je obličej v ose y natočen.

Nejprve si podle zjištěné šířky obličeje odvodíme pomocí konstanty k vzdálenost l od středu hlavy ke špičce nosu:

$$l = \sqrt{(x_{14} - x_2)^2 + (y_{14} - y_2)^2} * k \quad (9)$$

Dále je potřeba si pročítat střed hlavy s . Ten spočítáme jako střed mezi levým a pravým okrajem obličeje:

$$x_s = \frac{x_{14} + x_2}{2} \quad (10)$$

$$y_s = k_1 x_s + q_1 \quad (11)$$

Pak již lze vypočítat výsledný úhel β natočení hlavy v ose y . K výpočtu využijeme faktu, že se hlava otáčí kolem svého středu a že vzdálenost špičky nosu od středu je pořád konstantní, proto špička nosu opisuje kružnici kolem středu hlavy. S těmito faktami pak již můžeme jednoduše vypočítat výsledný úhel β pomocí arkus sinus takto:

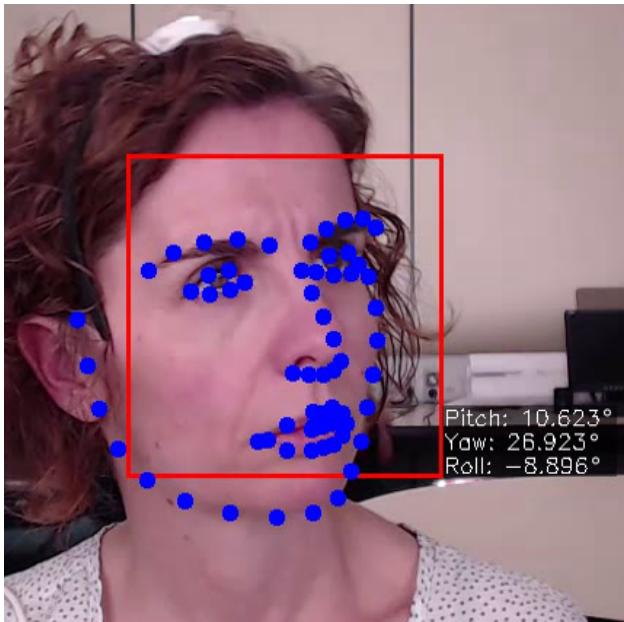
$$\beta = \sin^{-1} \frac{\sqrt{(x_p - x_s)^2 + (y_p - y_s)^2}}{l} \quad (12)$$

Pokud vyjde úhel β kladný, pak se hlava otáčí doprava z pohledu pozorovatele, a pokud vyjde záporný, tak se otáčí doleva z pohledu pozorovatele.

Místo bodu ležícím pod špičkou nosu se dá v podstatě použít jakýkoliv jiný bod ležící ve středu obličeje. Celý výpočet by byl v podstatě stejný jen by se lišila konstanta k .

3. Implementace

K implementaci tohoto projektu byl použit programovací jazyk C++ a knihovny OpenCV [7] a Dlib [8]. V současnosti se jedná o konzolovou aplikaci. Jako



Obrázek 3. Ukázkový výstup z aplikace

vstup lze zadat obrázek, video nebo výstup z webkamery. Výstup je buď zobrazen přímo na obrazovku, nebo je uložen do odpovídajícího souboru.

V následující části této kapitoly bude popsána implementace jednotlivých částí. Jako první bude popsána detekce obličejů. Problém detekce obličejů je již poměrně známý. Kvalitní detektory obličejů jsou již implementovány například v knihovně OpenCV. Tento detektor dosahuje velmi vysoké úspěšnosti a je proto vhodný pro použití v tomto programu. OpenCV používá k detekci algoritmus Viola-Jones. Detekovaný obličej je ve výstupním snímku ohraničen červeným obdélníkem.

K detekci charakteristických bodů v obličeji byla použita knihovna Dlib, která již obsahuje natrénovanou šablonu s 68 charakteristickými body. Detekce charakteristických bodů probíhá nad každým detekovaným obličejem. Charakteristické body jsou ve výstupním snímku zobrazeny jako modré kruhy a navíc lze ke každému charakteristickému bodu zobrazit i jeho číslo ze šablony.

Samotný výpočet úhlu natočení hlavy se provádí nezávisle pro každou osu. Výpočet vychází z detekovaných charakteristických bodů z předchozí části. Tato část je implementována přesně podle podkapitoly Určení natočení hlavy 2.3. Úhly jsou uváděny ve stupních.

Výstupní snímek tedy může obsahovat pro každý detekovaný obličej obdélník ohraničující obličej, charakteristické body a spočítané úhly natočení hlavy. Ukázka takového obrázku je na obrázku 3.

4. Testování

Testování výsledků bylo prováděno tak, že se měřila průměrná chyba ve všech třech osách vůči videům s referenčními výsledky. Jako referenční videa byla použita videa z datasetu od Gi4E [9]. Tento dataset obsahuje 120 videí, které jsou tvořeny 10 osobami, kde každá osoba natočila 12 videí. Každé video trvá 10 s a je tvoreno 300 snímků. Videá byly pořízeny běžnou webkamerou v rozlišení 1280x720 pixelů. Přesná hodnota úhlů je určena pomocí senzorů umístěných na hlavě. Ukázková fotografie z datasetu je na obrázku 3, další fotografie z tohoto datasetu jsou zobrazeny v hlavičce tohoto dokumentu.

Průměrná chyba pro každý úhel se spočítá jako průměrná hodnota absolutní hodnoty z rozdílu výsledku mého programu s referenčním výsledkem:

$$c = \frac{1}{N} \sum_{k=1}^N |m_k - r_k| \quad (13)$$

kde c je výsledná chyba, m_k je výsledný úhel z mého programu pro k -tý vzorek, r_k je referenční hodnota úhlu pro k -tý vzorek a N je počet vzorků.

Výsledná průměrná chyba mého programu je pro $roll$ $0,83^\circ$, pro yaw $2,1^\circ$ a pro $pitch$ $6,56^\circ$.

Pro všechny úhly byla také spočtena směrodatná odchylka, dle vzorce:

$$\sigma = \sqrt{\frac{1}{N} \sum_{k=1}^N (m_k - \bar{c})^2} \quad (14)$$

kde σ je výsledná směrodatná odchylka, m_k je výsledný úhel z mého programu pro k -tý vzorek, \bar{c} je průměrná chyba pro daný úhel a N je počet vzorků.

Směrodatná odchylka pro $roll$ vyhází $0,65^\circ$, pro yaw vychází $1,97^\circ$ a pro $pitch$ vychází $5,77^\circ$.

Na stránkách datasetu je tabulka, kde lze porovnat výsledky svoje výsledky s výsledky ostatních, kteří také testovali na tomto datasetu. Jsou zde vidět průměrné chyby pro 4 různé implementace. Pro $roll$ je v tabulce nejnižší hodnota průměrné chyby $1,12^\circ$ a průměrná hodnota průměrné chyby je $1,44^\circ$, pro yaw je nejnižší chyba $2,3^\circ$ a průměrná hodnota chyby je $3,13^\circ$ a pro $pitch$ je nejnižší chyba $4,04^\circ$ a průměrná hodnota chyby je $6,1^\circ$.

Z daných výsledků je patrné, že v porovnání se 4 ostatními detektory natočení, dosahuje můj detektor nejnižší průměrné chyby pro $roll$ a yaw a je mírně podprůměrný při výpočtu $pitch$.

5. Závěr

V rámci této práce jsem vytvořil program, který dokáže ve vstupním videu detektovat obličeji a následně určit

natočení hlavy vůči kameře. Program byl implementován s využitím knihoven OpenCV a Dlib.

Pro testování jsem využil dataset od Gi4E. Dataset obsahoval 120 videí o 300 snímcích. Testováním jsem zjistil, že můj algoritmus dosahuje lepších výsledků než ostatní algoritmy použité na stejně testovacím datasetu.

Tato práce může mít do budoucna přínos při zdokonalování metod pro rozpoznávání a identifikaci osob. Získané výsledky mohou mít pozitivní vliv na zlepšení využitelnosti dat z CCTV systémů, například pro strojové rozpoznávání osob.

Poděkování

Rád bych poděkoval panu Ing. Tomáši Goldmannovi za odbornou pomoc, cenné rady a věnovaný čas při řešení této práce.

Literatura

- [1] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: an open source facial behavior analysis toolkit. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.
- [2] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR)*, pages 617–624, June 2011.
- [3] Viola Paul and Jones Michael. Robust real-time face detection. *International Journal of Computer Vision*, 2004.
- [4] Dalal Navneet and Triggs Bill. Histograms of oriented gradients for human detection. *INRIA Rhone-Alps*, 2005.
- [5] T. F. Cootes and J. C. Taylor. Statistical models of appearance for computer vision. 2004.
- [6] Poston Alan. *Human engineering design data digest*. Departmen of defence, 2000.
- [7] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.
- [8] King Davis, E. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [9] Ariz Mikel, Bengoechea José J., Villanueva Arantxa, and Cabeza Rafael. A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. *Computer Vision and Image Understanding*, 148:201–210, July 2016.