

# Kapitola 1

## 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.

### 1.1 Statistické modely tvaru

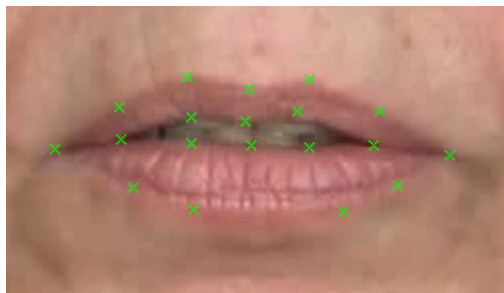
Statistické modely tvaru se využívají k reprezentaci objektů v obrázcích. Statistická analýza se aplikuje na data reprezentující určitý tvar za účelem nalezení a vytvoření jeho modelu.

#### 1.1.1 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 1.1). Pro zjištění vzájemné závislosti mezi pohybem jednotlivých bodů využijeme Analýzy hlavních komponent.

#### 1.1.2 Analýza hlavních komponent

Analýza hlavních komponent (Principal Component Analysis, dále jen PCA) slouží k dekorrelaci dat a je často využívána k redukcí 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.

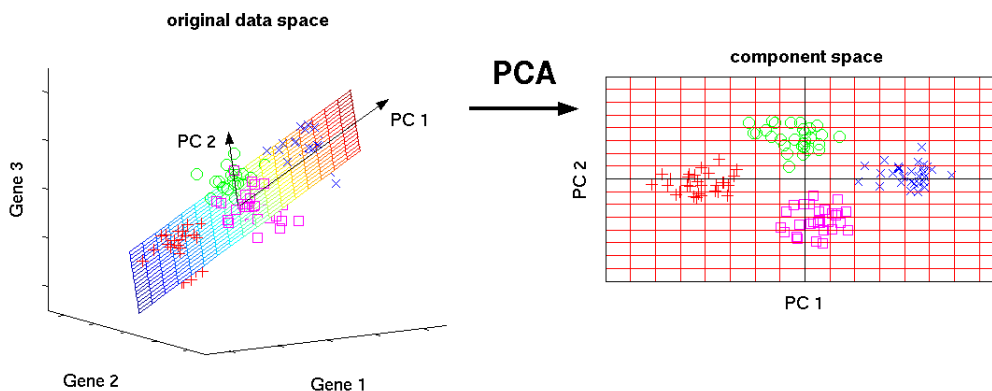


Obrázek 1.1: Významné body

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.

$$C(i, j) = (x_i - \mu_i)(x_j - \mu_j)^T \quad (1.1)$$

Prvky kovarianční matice  $C$  vypočteme podle vzorce 1.1, kde  $x_i$  a  $x_j$  jsou  $i$ -tý a  $j$ -tý vektor dat a  $\mu$  je příslušná střední hodnota. Z kovarianční matice vypočteme použitím dostupného algoritmu matici vlastních vektorů  $\Phi$  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 například 99% rozptylu dat (je zvykem zachovávat 95% či 99%). Procentuální hodnoty vlastních čísel pak sčítáme dokud nedostaneme požadovanou hodnotu a ostatní vlastní čísla a k nim náležící vektory vypustíme.



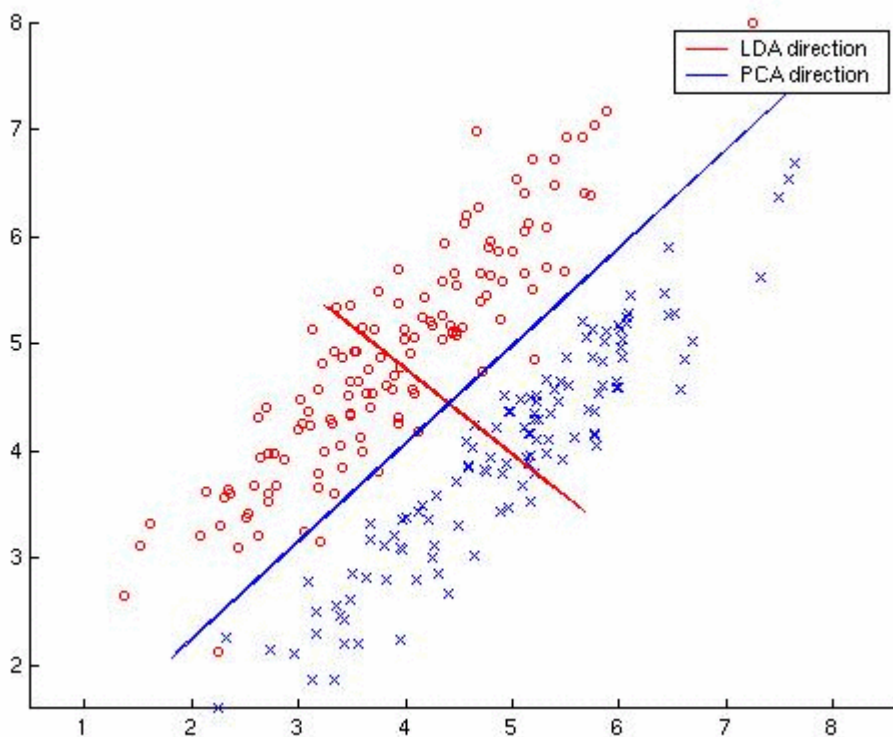
Obrázek 1.2: PCA [1]

Samotná PCA byla vymyšlena v roce 1901 anglickým matematikem Karlem Pearsonem. Nyní se používá především pro účely průzkumné analýzy dat a pro prediktivní modely. Její nespornou výhodou je její šetrnost k relativnímu měřítku souboru dat. Jedná se také o jednu z nejjednodušších vícerozměrných analýz vlastních vektorů.

### 1.1.3 Lineární diskriminační analýza

Lineární diskriminační analýza (Linear discriminant analysis, dále jen LDA) je jednou z metod mnohazměrné statistické analýzy a slouží k nalezení lineární kombinace příznaků, kterými lze charakterizovat daný soubor dat. LDA se opětovně používá k redukci dimenze dat.

LDA se, na rozdíl od PCA, která se snaží nalézt podprostor tak, aby jeho základní vektory odpovídali směrům s největší variancí v původním prostoru (ty které nejlépe popisují množinu dat), snaží nalézt takové vektory, které nejlépe diskriminují jednotlivé třídy (viz obrázek 1.3). LDA je hojně využívána při úlohách typu face recognition a to pro redukci velikosti příznakového vektoru.



Obrázek 1.3: PCA vs LDA

### 1.1.4 Lineární model tvaru

Výpočet tvaru probíhá podle vzorce 1.2.

$$x = \bar{x} + \Phi b \quad (1.2)$$

$\Phi$  je matice vlastních vektorů trénovacích dat tvaru,  $\bar{x}$  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  $b$ .

### 1.1.5 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  $\theta$  a mírou  $s$ . Pokud shrneme všechny tyto parametry do jedné funkce  $T$  dostaneme rovnici modelu ve tvaru

$$x = T_{X_t, Y_t, s, \theta}(\bar{x} + \Phi b) \quad (1.3)$$

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

$$T_{X_t, Y_t, s, \theta} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + \begin{pmatrix} s \cos \theta & s \sin \theta \\ -s \sin \theta & s \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (1.4)$$

Pokud nyní chceme najít takové parametry  $b$ , pomocí kterých vypočteme tvar  $x$  odpovídající požadovaným bodům v obraze  $Y$ , minimalizujeme vlastně výraz

$$\|Y - T_{X_t, Y_t, s, \theta}(\bar{x} + \Phi b)\|^2 \quad (1.5)$$

Tento postup se dá vyjádřit iterativním procesem:

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

$$y = T_{X_t, Y_t, s, \theta}^{-1}(Y) \quad (1.6)$$

5. Aktualizace parametrů modelu

$$b = \Phi^T(y - \bar{x}) \quad (1.7)$$

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.

## 1.2 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(1.2.1) tvarově nezávislou texturu z každého trénovacího obrazu. Na tyto textury lze aplikovat PCA(1.1.2) podobně jako u modelů tvaru. Předpokladem je, že existuje závislost mezi tvarem a příslušnou texturou a lze tedy vytvořit kombinovaný model tvaru a textury.

### 1.2.1 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.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \beta \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \gamma \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \quad (1.8)$$

V této rovnici jsou  $x_1, x_2$  a  $x_3$  vrcholy trojúhelníku a  $x$  je souřadnice bodu vyjádřená pomocí těchto vrcholů. Parametry  $\alpha, \beta$  a  $\gamma$  vypočteme z rovnic:

$$\alpha = 1 - (\beta + \gamma) \quad (1.9)$$

$$\beta = \frac{yx_3 - yx_1 - x_3y_1 - xy_3 + x_1y_3 + xy_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \quad (1.10)$$

$$\gamma = \frac{xy_2 - xy_1 - x_1y_2 - yx_2 + x_2y_1 + yx_1}{-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2} \quad (1.11)$$

$x$  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 1.8.

## 1.2.2 Lineární model textury

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

$$g = \bar{g} + \Phi_g b_g \quad (1.12)$$

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

## 1.2.3 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.

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s \Phi_s^T (x - \bar{x}) \\ \Phi_g^T (g - \bar{g}) \end{pmatrix} \quad (1.13)$$

$b$  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.  $\Phi_s$  a  $\Phi_g$  jsou vlastní vektory kovariančních matic trénovacích dat tvaru a textury.  $x$  jsou skutečné souřadnice pro daný obraz,  $\bar{x}$  je střední hodnota trénovacích dat tvaru.  $g$  je tvarově nezávislá textura příslušného obrazu a  $\bar{g}$  je střední hodnota trénovacích dat textury. Po provedení PCA lze stanovit kombinovaný model jako

$$b = \Phi_c c \quad (1.14)$$

$P_c$  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  $c$

$$x = \bar{x} + \Phi_s W_s^{-1} \Phi_{cs} c \quad (1.15)$$

$$g = \bar{g} + \Phi_g \Phi_{cg} c \quad (1.16)$$

kde

$$\Phi_c = \begin{pmatrix} \Phi_{cs} \\ \Phi_{cg} \end{pmatrix} \quad (1.17)$$

## 1.3 Active Shape Model

Active Shape Model, dále jen ASM, je statistická metoda modelů tvarů, které se iterativně deformují za účelem nalezení nejlepší 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 1.3 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  $(X_t, Y_t, s, \theta, b)$ , tak aby co nejlépe seděli na nové body  $\mathbf{X}$
3. Opakujeme proces dokud nedokverguje 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.

### 1.3.1 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 vypočítat střední hodnotu profilu  $\bar{g}$  a kovarianci  $S_g$ . Kvalita nasazení nového profilu  $g_s$ , na model profilu se pak vypočte jako Mahalanobisova vzdálenost

$$f(g_s) = (g_s - \bar{g})^T S_g^{-1} (g_s - \bar{g}) \quad (1.18)$$

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 1.1.5.

## 1.4 Active Appearance Model

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

### 1.4.1 Vyhledávání v AAM

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

$$\delta I = I_i - I_m \quad (1.19)$$

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  $\delta I$  změnit parametry modelu  $c$  můžeme dostat časově efektivní algoritmus vyhledávání.

### 1.4.2 Natrénování úprav parametrů modelu

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

$$r(p) = g_s - g_m \quad (1.20)$$

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 1.20 dostaneme

$$r(p + \delta p) = r(p) + \frac{\partial r}{\partial p} \delta p \quad (1.21)$$

Při hledání nejlepší pozice máme k dispozici residuum  $r$ . Chceme získat  $\delta p$  takové, aby minimalizovalo výraz  $|r(p + \delta p)|^2$ . Položením rovnice 1.21 rovné nule dostaneme

$$\delta p = -Rr(p) \quad kde \quad R = \left( \frac{\partial r^T}{\partial p} \frac{\partial r}{\partial p} \right)^{-1} \frac{\partial r^T}{\partial p} \quad (1.22)$$

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

$$R = -r(p)^{-1} \delta p \quad (1.23)$$



### 1.4.3 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  $\delta g_1$  a chyby  $E_1$
- Když je  $E_1 < E_0$  přijmeme  $c_1$  jako nové parametry modelu
- V jiném případě zkusíme  $k = 1.5$ ,  $k = 0.5$ ,  $k = 0.25$  atd

# Literatura

- [1] Matthias, S.. Approaches to analyse and interpret biological profile data, POTSDAM UNIVERSITY, 2006, Potsdam, Germany
- [2] Cootes, T. F. and Taylor, C. J.. Statistical Models of Appearance for Computer Vision, Manchester, 2004.
- [3] Aleix, M.. PCA versus LDA, IEEE, 1999.