

Cvičení z předmětu Biometrie

Úloha 3: Rozpoznávání duhovky

E. Bakštein

23. listopadu 2011

Úvod

Cílem úlohy je vyzkoušet v praxi metody, používané pro rozpoznávání duhovky. Úloha je složena následujícími částmi:

1. Implementace kódu pro segmentaci oblasti duhovky pomocí Houghovy transformace
2. Přiřazení zadaných obrázků oka k irisCodes v databázi

Řešení bude probíhat formou implementace požadovaných funkcí do existujícího Iris Toolboxu, který je ke stažení na stránkách cvičení.

Předpokládá se samostatná práce na úloze.

Úloha 4, Duhovka - extrakce příznaků a porovnávání irisCode (10 bodů)

Termín odevzdání: 14. 12. 2011, 9:15

1 Segmentace oblasti duhovky pomocí Houghovy transformace

K úspěšnému zakódování informace, obsažené ve struktuře duhovky, je nezbytné nalézt a oddělit (segmentovat) oblast duhovky od okolí (bělma, víčka atd.). Nejjednodušším způsobem je dostatečně přesná aproximace oblasti duhovky jako mezikružím. K nalezení oblasti duhovky pak můžeme použít Houghovu transformaci (HT) pro kruh (Viz přednáška 9). Při této metodě se nejprve pomocí hranového detektoru naleznou v obrázku hrany a následně se každý nalezený hranový bod projektuje do prostoru parametrů. Projekce pak probíhá přičtením pevně dané konstanty na všechna místa prostoru parametrů, kterými prochází pomyslná kružnice se středem ve zvoleném hranovém bodě a s daným poloměrem. Za předpokladu známého poloměru se v bodě středu hledaného kruhu v prostoru parametrů vyskytne maximum. Postup pro neznámý poloměr je obdobný: provede se projekce pro hodnoty poloměru ve zvoleném rozsahu a následně se hledá maximální hodnota přes celý prostor parametrů.

Úkol:

1. Implementujte funkci

```
[circlePupil, circleIris] = findIrisAnnulus(uint8 iris_image),
```

založenou na Houghově transformaci. *circlePupil* a *circleIris* jsou vektory parametrů nalezených kružnic [x, y, poloměr] pro zornici a duhovku. Rozsahy parametrů HT pro prohledávání vhodně zvolte tak, aby nebyla nutná jejich úprava pro jednotlivé Vám přidělené obrázky. duhovky.

2. Funkci využijte spolu se zbytkem Iris Toolboxu pro zakódování zadaných obrázků duhovky.

Pozn.: Pro detekci hran použijte funkci *edge(I, 'canny')*. K vizualizaci nalezených hran můžete využít funkci *circlecoords*, obsaženou v Iris Toolboxu

Tip.: Pro zrychlení výpočtu je možné po detekci vnějšího okraje duhovky oříznout pro další hledání obrázků na oblast duhovky. Tím se výrazně zmenší prostor parametrů.

2 Nalezení zadaných obrázků v databázi

Tato část úlohy simuluje proces identifikace na základě duhovky tak, jak probíhá v reálném prostředí. Neznámé osobě jsou nejprve sejmuty fotografie duhovky (zde: zadané obrázky), které jsou zakódovány shodným způsobem, jako zbytek databáze. Následně se v databázi vyhledává uložený vzorek, který je právě sejmutému vzorku nejpodobnější. Tímto způsobem je v databázi nalezen nejpravděpodobnější subjekt.

K porovnání dvou irisCodes se používá Hammingovy vzdálenosti (*Hamming distance - HD*), normované počtem překrývajících se použitelných bitů v maskách obou Iris-kódů:

$$HD = \frac{\| (codeA \otimes codeB) \cap maskA \cap maskB \|}{\| maskA \cap maskB \|}, \quad (1)$$

kde \otimes je logický *XOR* operátor, \cap logický *AND* a $\|$ *norm* operátor - počet jedničkových bitů v daném řetězci. *codeA* a *codeB* jsou porovnávány iris kódy a *maskA* a *maskB* příslušné šumové masky (zde 1 je užitečný signál, 0 šum. *Pozor: v Iris toolboxu je konvence opačná!*)

Aby se předešlo nežádoucímu vlivu rotace duhovky (naklopení hlavy) na výsledek porovnání, používá se vzájemný bitový posun iris kódů vůči sobě ve zvoleném rozsahu a výpočet Hammingovy vzdálenosti pro každý posun. Jako výsledek porovnání se pak bere nejlepší shoda - tedy nejmenší dosažená vzdálenost.

Úkol:

1. Implementujte funkci

```
HD = irisHammingDistance(codeA,codeB, maskA, maskB)
```

pro porovnání dvou iris kódů na základě vzorce (??). Implementujte vzájemný bitový posun iris kódů (rozsah volte ± 15 bitů).

2. Projděte databázi uložených iris kódů (*database.mat*) a najděte takovou osobu, od které Vám přidělený obrázek duhovky nejpravděpodobněji pochází. Spočtené hodnoty Hammingových vzdáleností vynesete do histogramu (funkce *hist*). K načtení a procházení databáze můžete pro usnadnění použít připravený skript *walkDatabase.m*

Bonus:

- Do výpočtu vzdálenosti iris kódů zahrňte vzorec pro statistickou korekci vlivu počtu efektivních porovnávaných bitů (viz přednáška 9, slide 36).

Stručný přehled Iris Toolboxu

- Představu o funkci toolboxu si uděláte spuštěním *iris_demo.m*. Při spuštění dema se zobrazí hodnoty předem předpočítané a uložené v cache.
- Hlavní funkcí, zodpovědnou za extrakci příznaků, je *createiristemplate.m*
- Pro nastavení parametrů a přidání adresářů do PATH matlabu slouží *iris_init.m*
- Kostry funkcí, které máte za úkol naimplementovat, najdete v adresáři *ToDo*