

ISS Projekt 2013 / 14

Tomáš Mikolov, Jiří Kopecký a Honza Černocký, ÚPGM FIT VUT
December 14, 2013

1 Cíl a odevzdání

Cílem tohoto projektu je použít některé základní funkce pro digitální zpracování obrazu. Jako vstup použijte dodaný obrázek `xlogin00.bmp`, kde “xlogin00” je Váš login. Projekt je možno řešit v Matlabu, jazyce C nebo v libovolném jiném programovacím nebo skriptovacím jazyce.

Odevzdání projektu bude probíhat do informačního systému WIS jako jeden zip-soubor se jménem `xlogin00.zip`, kde “xlogin00” je Váš login. Tento bude obsahovat:

- textový soubor `reseni.txt`, ve kterém budou numerické výsledky. Soubor nebude mít žádnou hlavičku a bude obsahovat pouze řádky s výsledky, žádné prázdné řádky či poznámky. Používejte laskavě čisté ASCII kódování, ne UTF8 nebo UTF16.
- soubory `*.bmp`, kde budou výsledné obrázky. Výsledné obrázky odevzdejte ve stejném formátu, v jakém je vstupní obrázek - nekomprimované BMP 512x512 s 8-mi bitovou barevnou hloubkou (bude probíhat automatická kontrola).
- adresář `src/`, kde odevzdáváte Vaše zdrojové kódy, tedy program v C, Matlabu nebo čemkoliv jiném. Projekt je **samostatná práce**, proto budou Vaše zdrojové texty křížově korelovány a v případě silné podobnosti budou vyvozeny příslušné závěry.
- V případě, že se vypracujete **bonusový úkol**, budou ve Vašem zip souboru rovněž soubory `bonus.txt` a `bonus.bmp`. Více viz sekce 3.

Věnujte laskavě pozornost tomu, abyste projekt odevzdali přesně v popsáném formátu, v případě, že bude nutné Vaše výsledky ručně modifikovat (upravovat texták, přesouvat soubory, přejmenovávat soubory, atd atd, budeme aplikovat penalizaci -2 body.

2 Zadání

Zaostření obrazu [1 bod]

Načtete Váš vstupní obrázek (např. `xlogin00.bmp`) a proveďte jeho zaostření pomocí lineárního filtru:

$$\mathbf{H} = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ -0.5 & 5.0 & -0.5 \\ -0.5 & -0.5 & -0.5 \end{bmatrix}$$

Výsledek odevzdejte jako soubor `step1.bmp`

Otočení obrazu [2 body]

Proveďte překlopení zaostřeného obrazu kolem svislé osy.

Výsledek odevzdejte jako soubor `step2.bmp`.

Mediánový filtr [1 bod]

Nyní aplikujte na obraz mediánový filtr pro okolí 5x5 pixelů. Mediánový filtr se používá pro redukci šumu v obraze. Funguje tak, že vybere prostřední hodnotu ze seřazeného seznamu hodnot z okolí pixelu (v našem případě máme na vstupu 5x5=25 pixelů a vybereme tedy 13. hodnotu v seřazeném seznamu). Používáte-li Matlab, můžete využít vestavěnou funkci `medfilt2`.

Výsledek odevzdejte jako soubor `step3.bmp`

Rozmazání obrazu [2 body]

V tomto kroku použijte filtr:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 3 & 3 & 1 \\ 1 & 3 & 9 & 3 & 1 \\ 1 & 3 & 3 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} / 49$$

Výsledek odevzdejte jako soubor `step4.bmp`

Chyba v obraze [1 bod]

V předcházejících krocích jsme obrázek několikrát modifikovali. Spočítejte, jaké průměrné chyby vztažené na 1 pixel jsme se dopustili mezi originálním obrázkem (např. `xlogin00.bmp`) a tím, který jsme získali v předchozím kroku (`step4.bmp`). **Srovnávejte stejně otočené obrazy!** Upozornění: při výpočtu chyby budete muset nejprve převést oba obrázky z formátu `uint8` do standardního Matlabovského `double`. V laboratoři o zpracování obrazu (příklad 6) byla v tomto bodu chyba, nestačí převést až výslednou absolutní hodnotu !

Výsledek napište jako řádku

`chyba=xx.yyy`

do souboru `reseni.txt`. `xx.yyy` bude samozřejmě Vámi vypočtená chyba.

Roztažení histogramu [2 body]

Zobrazte si histogram k Vašemu obrázku. Je vidět, že nejsou využity všechny hodnoty - proveďte proto roztažení histogramu tak, aby byly využity všechny hodnoty 0-255. Návod: Nejprve nalezněte minimální a maximální hodnotu v obraze a poté lineárně přemapujte tento interval do rozsahu 0-255. Vygenerujte obrázek s přemapovanými hodnotami. I v tomto případě budete zřejmě muset nejprve převést obrázek na `double`, výsledný obrázek pak před zobrazením a uložením opět na `uint8`. Upozornění: v příkladu je zadáno **roztažení histogramu**, nikoliv **ekvalizace histogramu**, nelze tedy použít příkaz `histeq` jako v laboratoři o zpracování obrazu.

Výsledek odevzdejte jako soubor `step5.bmp`

Spočítání střední hodnoty a směrodatné odchylky [1 bod]

Spočítejte střední hodnotu a směrodatnou odchylku v obraze před a po provedení roztažení histogramu. Opět nezapomeňte na převody na `double`.

Výsledky napište jako řádky

`mean_no_hist=xx.yyy`

`std_no_hist=xx.yyy`

`mean_hist=xx.yyy`

`std_hist=xx.yyy`

do souboru `reseni.txt`. `xx.yyy` budou samozřejmě Vámi vypočtené hodnoty. **Používejte laskavě desetinné tečky, ne čárky ! K oddělení řádků laskavě použijte LF (Linux) nebo CR+LF (Windows), jiné oddělovače nás nutí ručně opravovat a Vás připraví o 2 body.**

Kvantizace obrazu [2 body]

Proveďte kvantizaci obrazu, který jste získali po roztažení histogramu. Uvažujte kvantizaci na 2 bity - ve výsledku tedy musí být zastoupeny 4 stupně šedi. I v tomto příkladu bude potřeba převod na `double`. Pokud budete využívat funkci pro kvantizaci z cvičení “Vzorkování, kvantování”, opravte si chybu ve funkci v příkladu 5 – je nutné použít zaokrouhlení, nikoli ořezání:

`round(((2^N)-1)*(double(I5)-a)/(b-a))*(b-a)/((2^N)-1) + a;`

Výsledek odevzdejte jako soubor `step6.bmp`

3 Bonusová práce – detekce špiček letadel [víno pro vítěze]

Projekt je dosti triviální a pokud jste dobře poslouchali v laboratořích, budete jej mít za hodinku hotový i s ověřením na referenčním obrázku.

Pro ty, co by si chtěli zkusit něco více, je zde následující úkol:

Detekujte v obrázku špičky všech letadel, označte je viditelně, (např. bílým křížkem) do obrázku a obrázek uložte do `bonus.bmp`. Do textového souboru `bonus.txt` popište, jak jste detekci prováděli.

K detekci špiček je možné přistupovat různě:

- vzít v úvahu, jak vypadá jejich geometrie, popsat matematicky, detekovat v obrázku (např. po předchozí detekci hran).
- natrénovat libovolný klasifikátor na několika (nebo mnoha) obrázcích s označovanými špičkami letadel, použít na Vašem obrázku (pokud se rozhodnete pro toto řešení, trénovací obrázky prosím neodevzdávejte, ale do `bonus.txt` popište, kolik jich bylo, kde jste je sehnali a jak jste zjistili, že tam nějaké špičky skutečně jsou).
- případně kombinace výše uvedených přístupů + jakékoliv další originální.

Výsledky prosím ručně neupravujte, zajímá mě skutečný výsledek Vašeho algoritmu, ne to, jak jste zruční ve Photoshopu, Paintu či Gimpu. I zajímavý algoritmus, který špičky úplně netrefí, může vyhrát.

Podobně jako u základního projektu můžete použít libovolný programovací či skriptovací jazyk a libovolné knihovny.

Hodnocení: za tuto část nejsou body. Nejoriginálnější řešení bude odměněno lahví dobrého francouzského červeného, vyhlášení proběhne na zkoušce.