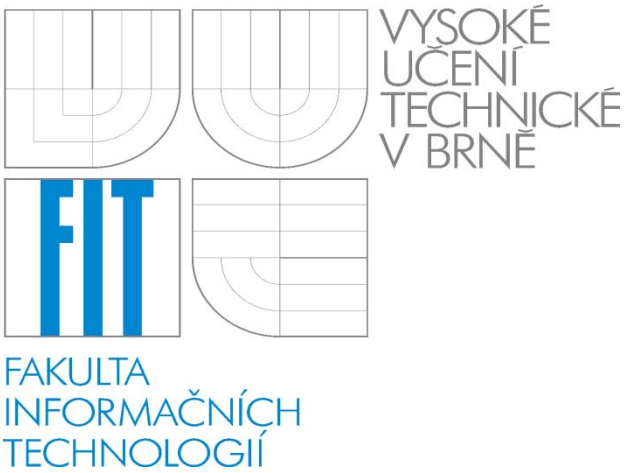


Projekt PGP

Nerealistické videoefekty

Petr Nohejl (xnohej00)

Petr Panáček (xpanac02)

28.12.2010

1 Úvod

Cílem práce bylo nastudovat metody používané v nerealistickém zobrazování videa. Hlavním úkolem pak bylo navrhnout a implementovat aplikaci realizující vybraný efekt na videosekvenci a zhodnotit možnosti metod pro zpracování videa (stabilita mezi framy, plynulost pohybů apod.). Nakonec bylo úkolem demonstrovat dosažené výsledky na vhodných vstupech a vyhodnotit.

V rámci projektu jsme implementovali aplikaci demonstrující 3 nerealistické video efekty: komiksový efekt, malířský efekt a mozaikový efekt.

2 Nerealistické videoefekty

Nerealistické renderování je oblast počítačové grafiky, která se zaměřuje na různé metody expresivního vyjádření (vykreslení) digitálního obrazu. Jedná se o opačný proces vykreslování realistické grafiky, tedy soubor technik vedoucích k výstupu, který nemá charakter fotorealistického grafického výstupu. Zpravidla se používá tehdy, kdy je potřeba imitovat různé techniky výtvarného umění nebo zvýraznit některé rysy zobrazovaného útvaru. Vstupem je videosekvence reálného obrazu, pořízená videokamerou, 3D scéna nebo jiná videosekvence. Výstupem filtru je vyrenderovaná videosekvence s daným nerealistickým efektem. [1]

Příklady některých nerealistických efektů:

- Komiks (cartoon, comic)
- Malířské techniky (painterly rendering)
- Náčrtky (sketching)
- Deformace obrazu (liquify)

Komiksový efekt

Komiksový efekt napodobuje kreslený film. Výsledkem je segmentovaný obraz s redukovaným rozsahem barev a se zvýrazněnými hranami (obr. 1).

Algoritmus pro vykreslování efektu používá metodu shlukování, konkrétně algoritmus K-means (viz kapitola 3). Při každém vykreslení snímku nejprve proběhne výpočet počtu shluků. To zajišťuje lepší stabilitu obrazu mezi jednotlivými snímky. Počet shluků se určuje z barevnosti snímku pomocí histogramu a mění se jen při výraznější změně barevnosti.

Algoritmus K-means segmentuje pixely obrazu do shluků podle 5 kritérií: červená barevná složka, zelená barevná složka, modrá barevná složka, pozice na ose x, pozice na ose y. Barva jednotlivých shluků se vypočítá z průměru barev všech pixelů daného segmentu.

Nakonec se v obraze zvýrazní hrany pomocí Cannyho algoritmu pro detekci hran (viz kapitola 3). Ten najde v obraze hrany, rozšíří je pomocí dilatace o určitou velikost a odpovídající pixely ve výsledném obraze potom ztmaví o danou hodnotu.

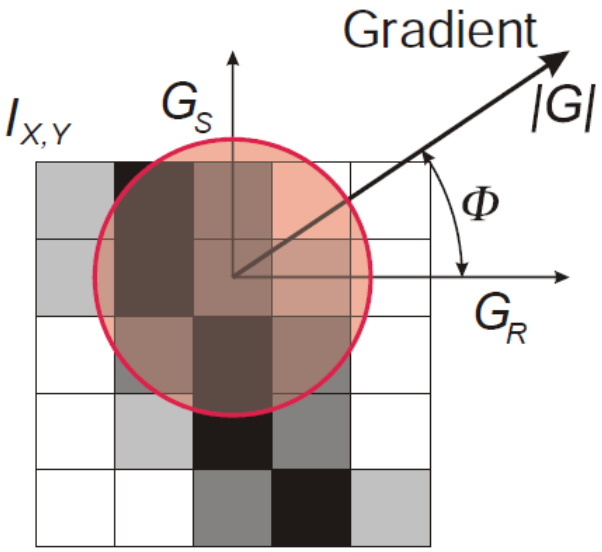
Obr. 1: Ukázka komiksového efektu.
Vlevo originální snímek, vpravo vyrenderovaný snímek.

Malířský efekt

Malířský efekt imituje obraz malovaný štětcem na plátno. Výsledkem je obraz složený z menších různobarevných oblastí, připomínající stopy štětce (obr. 4).

Algoritmus zpracovává každý snímek videa v několika průchodech. V těchto průchodech jsou do obrazu zakreslovány čáry o různých tloušťkách, přičemž každá čára představuje jeden tah štětcem (obr. 3). Při zpracování jednoho snímku je ze všeho nejdříve rozmazán původní obrázek. V každém průchodu je poté obrázek překreslen jemnějším štětcem. Kromě zmenšování tloušťky štětce je také v každém průchodu zvětšována hustota zakreslovaných čar. Jinými slovy, je zmenšován krok průchodu obrazem. Jemnější štětce jsou ovšem použity pouze v místech, kde se aktuální obraz liší od původního rozmazaného obrazu. Díky tomu je zajištěna vyšší stabilita mezi jednotlivými snímky. Překreslování tenčím štětcem tedy slouží k odstranění nepřesností v aktuálním obraze. Po použití jemnějšího štětce je obraz vždy znovu rozmazán. Rozmazání obrazu se provádí pomocí algoritmu Gaussian blur (viz kapitola 3).

Směr kreslené čáry je určen kolmicí ke gradientu obrazu. Gradient je vypočítán pomocí Sobelova filtru (viz kapitola 3) použitého ve směru osy x a y (obr. 2). Kolmice je poté získána pouhým prohozením souřadných os a změnou znaménka jedné osy. Barva čáry je vypočítána jako průměrná hodnota mezi počátečním a koncovým bodem čáry, přičemž hodnota těchto bodů je brána z původního rozmazaného obrazu. [2]



Obr. 2: Vykreslení čáry.

Obr. 3: Vykreslování čar v jednotlivých průchodech (postupně zleva doprava).

Obr. 4: Ukázka malířského efektu.

Vlevo originální snímek, vpravo vyrenderovaný snímek.

Mozaikový efekt

Mozaikový efekt je jednoduchý a často používaný. Obraz napodobuje mozaiku. Používá se pro skrytí detailů v obraze. Výsledkem je obraz složený z větších jednobarevných čtverců (obr. 5).

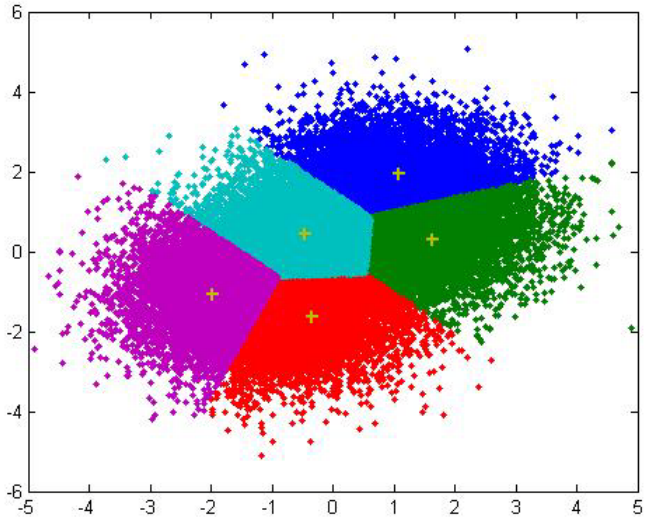
Algoritmus rozděluje obraz na čtvercové oblasti o určité velikosti. Tyto oblasti jsou potom obarveny jednou barvou, vypočtenou z průměru barev všech pixelů původní oblasti.

Obr. 5: Ukázka efektu mozaiky.
Vlevo originální snímek, vpravo vyrenderovaný snímek.

3 Algoritmy pro zpracování efektů

K-means

Algoritmus K-means slouží k segmentaci obrazu. Obraz se skládá z *K* regionů a každý pixel je klasifikován do jednoho z regionu. Každý region je reprezentován středem μ_i. Pixel je klasifikován do regionu, jehož střed je nejblíže (obr. 6).



Obr. 6: Ukázka segmentace podle K-means ve 2D (kříže znázorňují středy regionů μ_i). [3]

Algoritmus K-means:

1. Náhodně inicializuj středy regionů μ_i.
2. Klasifikuj pixely x_n do regionů.
3. Vypočti nové středy regionů.
4. Opakuj kroky 2 a 3, dokud se středy shluků mění.

Cannyho hranový detektor a Sobelův operátor

Cannyho hranový detektor je algoritmus, který nalezne množinu hran v obraze. Využívá Sobelův operátor ve směru *X* a *Y*, potlačení nemaximálních hodnot a prahování s hysterezí. Zvýraznění hran pomocí Cannyho algoritmu je vidět například v obr. 1.

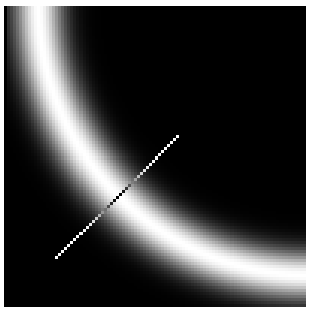
Sobelův operátor je diskrétní diferenciální operátor, který počítá aproximaci gradientu intenzity obrazu. Je založen na konvoluci obrazu pomocí filtru v horizontálním a vertikálním směru. Zpravidla používá konvoluční matice o velikosti 3x3 (vzorec 1).

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

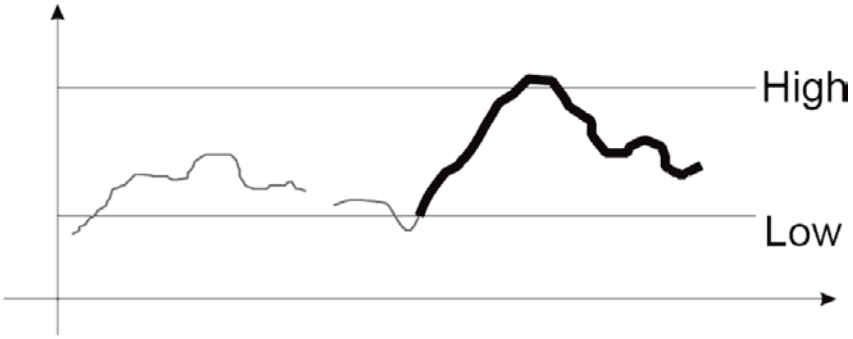
Vzorec 1: Konvoluční matice Sobelova filtru (A je zdrojový obraz, G je výsledný gradient). [4]

Potlačení nemaximálních hodnot (Non-maximum suppression) funguje tak, že ve směru gradientu se potlačí pixely, které nejsou lokálním maximem (obr. 7). Výsledkem je ztenčení tlustých hran potlačením nemaximálních hodnot v černobílém obraze.



Obr. 7: Znázornění pixelů ve směru gradientu.

Prahování s histerezí funguje tak, že silné hrany (větší než *High*) jsou ponechány. Slabé jsou ponechány, pouze pokud jsou spojeny s nějakou silnější hranou (obr. 8). Velmi slabé hrany jsou potlačeny (nižší než *Low*). Obvykle platí: *High* = 2-3 * *Low*.



Obr. 8: Prahování s histerezí.

Gaussian blur

Gaussian blur je často používaný algoritmus v počítačové grafice pro redukci šumu a detailů. Funguje na principu konvoluce pomocí Gaussovské funkce (vzorec 2). Gaussian blur je filtr s dolní propustí.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Vzorec 2: Dvou dimenzionální Gaussovská funkce.

4 Implementace aplikace

Aplikace demonstruje 3 video efekty: komiksový efekt, mozaikový efekt a malířský efekt. Vstupem programu je typ efektu a název video souboru ve formátu AVI. Výstupem jsou dvě okna, zobrazující video. První ukazuje originální obraz, druhé zobrazuje vyrenderovaný obraz, na který byl aplikován daný efekt.

Program demonstrující výše zmíněné nerealistické efekty je implementován v jazyce C/C++, v prostředí Visual Studio 2008. Používá knihovnu OpenCV. Pro úspěšné sestavení projektu je nutné mít nainstalovanou tuto knihovnu a správně definovány v nastavení projektu knihovny pro linker.

Každý efekt je implementován ve vlastní třídě (*effectComic*, *effectPainterly*, *effectMosaic*). Na začátku každého zdrojového souboru s efektem jsou definovány konstanty, ovlivňující různé vlastnosti při výpočtu efektu. Konstanty lze měnit a ovlivňovat tak některé parametry efektu. Třída *effects* se stará o základní zpracování obrazu - načtení videa ze souboru a zobrazení na výstup.

Ovládání aplikace:

- *VideoEffects.exe -h* - zobrazí nápovědu
- *VideoEffects.exe -c inputVideo* - demonstruje komiksový efekt
- *VideoEffects.exe -p inputVideo* - demonstruje malířský efekt
- *VideoEffects.exe -m inputVideo* - demonstruje mozaikový efekt

inputVideo je název vstupního video souboru ve formátu AVI.

5 Závěr

Nerealistické vykreslování obrazu může mít různé využití. Používá se například v moderních videokamerách pro „umělecké“ efekty nebo také v programech pro střihání videa. Využití může nalézt i v zábavném průmyslu, např. v počítačových hrách.

Díky projektu jsme se naučili metody a principy pro nerealistické renderování obrazu. Naučili jsme se nové algoritmy pro zpracování obrazu a prohloubili si znalosti v programování aplikací v OpenCV.

6 Zdroje

[1] Non-photorealistic rendering. In ***Wikipedia : the free encyclopedia*** [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9.1.2005, last modified on 14.10.2010 [cit. 2010-12-28]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Non-photorealistic_rendering>.

[2] HERTZMANN, Aaron; PERLIN, Ken. ***Painterly rendering : Painterly Rendering with Curved Brush Strokes of Multiple Sizes*** [online]. New York University, Media Research Laboratory : 2000 [cit. 2010-12-29]. Dostupné z WWW: <http://mrl.nyu.edu/projects/npr/painterly/>.

[3] ***Matlab Central*** [online]. 27.3.2008 [cit. 2010-12-29]. Efficient K-Means Clustering. Dostupné z WWW: <http://www.mathworks.com/matlabcentral/fileexchange/19344-efficient-k-means-clustering-using-jit>.

[4] Sobel operator. In ***Wikipedia : the free encyclopedia*** [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19.2.2004, last modified on 12.12.2010 [cit. 2010-12-29]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Sobel_operator>.