

INTELIGENTNÍ „ZEŠTÍHLOVÁNÍ“ OBRÁZKŮ

Aleš Hrabě
FJFI ČVUT – AMSM

akademický rok 2011/2012

1 Úvod

Program demonstrativně implementuje tzv. seam carving (či retargeting, liquid rescaling) algoritmus. Jeho efekt spočívá v tom, že neproporcionálně zmenší obrázek tak, že důležité objekty zůstanou zachovány v přibližně původní podobě (proporcích). Nejlépe se osvědčuje na obrázcích, kde na relativně jednodušší ploše je umístěno několik výrazných objektů, nebo naopak kde složitěji členěná scéna je prostoupena nějakými plochami či pruhy.

2 Princip

Algoritmus funguje na docela jednoduchém principu. Při zmenšování obrázku v jednom směru o n pixelů jde v zásadě o to najít podle nějak definované lokální energie (zřejmě tak aby pixely důležitých objektů měly energii nejvyšší) v obrázku n cest ve směru kolmém s nejnižším celkovou energií (cenou) a ty pak odstranit. Protože důležité objekty jsou zpravidla jasně odlišné od zbytku scény a tedy mají výrazné hrany, přirozeně se nabízí vzít jako energii míru změny jasu na daném pixelu – detekované hrany. Nejlevnější cesty je možné následně vyhledat například Dijkstrovým algoritmem nebo dynamickým programováním.

3 Implementace

Program SeamCarving je zpracován v Netbeans Java IDE. Třída, která implementuje vlastní algoritmus má název **Retargeting**. Přímě při načtení obrázku vypočítá jasovou složku obrázku (Y z prostoru YUV či YCbCr) a vyhledá hrany konvolucí s běžným Sobelovým filtrem. Až při potvrzení zvoleného zmenšení uživatelem spustí výpočet cest dynamickým programováním na samostatném vlákně. Výpočet probíhá po řádcích od horního řádku obrázku, při každé iteraci se pro každý pixel do pomocné matice uloží součet hodnoty aktuálního pixelu s nejmenším ze tří spodních přilehlých. Vedle toho se druhé pomocné matice vždy uloží směr, ve kterém se nejmenší hodnota nacházela. Z těchto matic se nakonec rekonstruuje cesta tak, že se v nejspodnějším řádku matice součtů energií vyhledá nejmenší hodnota a podle matice se směry se

následně postupuje nahoru. Pixely na této cestě se z obrázku odeberou a pixely vlevo od cesty se zprůměrují s pixely vpravo kvůli vyhlazení takto vzniklé hrany. Po každém odstranění cesty se matice přepočítají (protože odstranění naruší ostatní vypočtené hodnoty) a vyhledá se nová, pokud je potřeba.

4 Ovládání

Pro spuštění programu je nutné mít nainstalované Java Runtime Environment alespoň verze 7 (sestaveno v JDK 7u2). Soubor `SeamCarving.jar` by mělo být možné spustit dvojklikem, pokud nefunguje, otevřete příkazovou řádku/terminál v dané složce a zadejte `>java -jar SeamCarving.jar`. Pokud ani to nefunguje, je ve Windows pravděpodobně potřeba ručně nastavit do proměnné prostředí `PATH` složku instalace JRE.

Program má velice jednoduché ovládání, které nepotřebuje příliš komentář. Ihned po otevření obrázku (formát jpeg, png, bmp) se vypočítá jeho jas a hrany a zobrazí v záložkách *Luma* a *Edges*. Jezdcem se následně zvolí požadovaná šířka obrázku a potvrdí tlačítkem *Retarget*. Program pak zobrazí záložku s výsledkem *Result*. Pro srovnání s běžným způsobem zmenšenou fotkou stačí přepnout do záložky *Original*, velikost okna zůstane jako v záložce *Result*. Původní poměr se vždy nastaví tlačítkem *Ratio*. Odstraněná vlákna si můžeme prohlédnout v záložce *Seams*. Výsledný obrázek je samozřejmě možné uložit.

5 Výsledky

Na obrázcích je ukázka výstupu programu a srovnání proti originálu a obrázku zmenšenému běžnou (Lanczos) metodou. Při tomto zmenšení už je na stromech vlevo trochu patrná deformace.

Algoritmus je dále možné rozšířit pro naopak zvětšování obrázku doplňováním pixelů podél cest, jejichž barva se dopočítá jako průměr okolních. Samozřejmě také není problém provádět změnu velikosti i ve směru svislém. Jako další vylepšení se nabízí vyzkoušet různé metody pro výpočet energie nebo také optimalizovat hledání vláken (cest).

Obrázek 1: Původní obrázek



Obrázek 2: Zmenšení

(a) Změnou měřítka



(b) Seam Carving

