

Živé panorama

Lenka Jalůvková*



Abstrakt

Tento článek je zaměřen na tvorbu živých panoramat. Vstupem je video ze streamujících kamer, výstupem je taktéž stream – panorama neustále se překreslující podle aktuální polohy kamery. Výsledné řešení je založeno na algoritmech SIFT a SURF pro hledání klíčových bodů a deskriptorů ve snímcích, dále na algoritmu RANSAC pro výpočet homografie sloužící k následnému plynulému spojování snímků a splynutí přechodů. Překreslování snímků zleva doprava i zprava doleva vyžadovalo váhování přechodů z obou stran vkládaného snímku.

Tvorba panoramat je závislá na zvolené obrazové projekci, v našem případě byla použita planární a cylindrická projekce. Funkčnost navrhnutého řešení byla otestována na řadě testovacích nahrávek.

Klíčová slova: Spojování snímků — SIFT — SURF — RANSAC — Planární projekce — Cylindrická projekce

*xjaluv02@stud.fit.vutbr.cz, Fakulta informačních technologií VUT v Brně

1. Úvod

Panorama je vizuální reprezentace prostředí, které je viděno z jedné pozice v prostoru [1]. Označení panorama se používá při širokoúhlé reprezentaci obrazu ve fotografii, kresbě, filmu či videu.

V této práci se zabýváme problémem tzv. živých panoramat, a to jak po teoretické, tak i implementační stránce. Vstupem je video ze streamujících kamer, výstupem je opět stream – panorama neustále se překreslující podle aktuální polohy kamery.

Tvorba panoramat je již poměrně probádané téma, ať už po teoretické ([2], [3], [4]), tak i implementační stránce ([5], [6], [7], [8]). Všechna tato řešení však realizují pouze spojení snímků do statických panoramat, takže žádné z nich není přímo aplikovatelné na tvorbu živých panoramat.

Řešení navržené v tomto článku využívá algoritmy SIFT [9], SURF a RANSAC [10]. Navržené řešení je implementováno v prostředí *MATLAB R2013a* s využitím *Image Processing*, *Computer Vision System* a

dalších standardních Toolboxů. Řešení bylo otestováno na řadě testovacích videí.

Zvolené algoritmy jsou blíže popsány v druhé sekci článku, poslední dvě sekce jsou zaměřeny na experimentální výsledky a jejich shrnutí.

2. Tvorba živých panoramat

Tvorba panoramatických fotografií nemusí být tak jednoduchá, jak se na první pohled může zdát. Prvním krokem je extrakce snímků z videa, dále převod jednotlivých video snímků do vhodné projekční plochy, jejíž výběr je závislý od rotačního úhlu kamery v nasnímávaném videu. Dalším krokem je nalezení klíčových bodů a deskriptorů ve snímcích pomocí algoritmů SIFT či SURF, následně výpočet homografie pro nalezení vzájemné pozice klíčových bodů algoritmem RANSAC. Následně plynulé spojení snímků a vyhlazení přechodů a nakonec překreslování panoramatu podle pohybu kamery zleva doprava.

2.1 Extrakce snímků

Nejprve jsou z videa vyextrahovány jednotlivé snímky. Z videí se cyklicky vybírají snímky s určitým krokem, podle video frekvence. Tyto snímky se ukládají a dále zpracovávají dle níže popsanych kroků. Uživatel si zvolí, zda snímky chce ponechat v originální velikosti nebo je zkomprimuje.

2.2 Projekční plocha

V našem přístupu byly využity dva typy projekčních ploch, a to planární a cylindrická. Planární projekce je vhodná při panoramatech architektury, protože zachovává rovnost čar (nijak je nedeformuje). Tuto projekci lze však použít pouze v případě, že úhel snímání je malý (do 120°), nad tento úhel je výsledné panorama nepřijatelné, protože roztažení obrazu způsobuje ztrátu ostrosti [11] a vzniká tzv. motýlkový efekt [7], kdy se panorama stále více rozšiřuje, což je nežádoucí. Příklad vidíme na obrázku 1.



Obrázek 1. Planární panorama složené ze tří snímků s viditelným zkreslením a motýlkovým efektem.

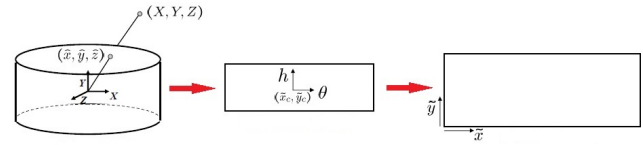
Tento problém lze částečně řešit vhodným zvolením středového snímku, jak vidíme na obrázku 2, ale výsledek ani přesto není optimální. Při spojení většího počtu snímků je panorama silně prohnuto.



Obrázek 2. Planární panorama se zvoleným středovým snímkem, redukuje se motýlkový efekt a roztažení obrazu.

V našem případě je tedy vhodnější projekce cylindrická, která je použitelná pro úhel snímání až do 360° . Je to standardní typ používaný právě u panoramatických kamer. Cylindrická projekce zachovává přesnější relativní velikosti objektů než planární projekce, to je však provedeno na úkor vykreslování rovnoběžných čar, které se v důsledku jeví jako zakřivené.

Tvorba cylindrického snímku je schematicky znázorněna na obrázku 3. Příklad cylindrického panoramatu je na obrázku 4.



Obrázek 3. Schematické znázornění převodu snímku do cylindrických souřadnic.



Obrázek 4. Cylindrické panorama ze tří snímků.

Pro tento převod je nejprve potřeba namapovat 3D body (X, Y, Z) na válec podle vzorce:

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Z^2}}(X, Y, Z). \quad (1)$$

Body na povrchu válce jsou parametrizovány úhlem θ a vahou h s 3D cylindrickými souřadnicemi odpovídajícími (θ, h) danými vztahem:

$$(\sin\theta, h, \cos\theta) = (\hat{x}, \hat{y}, \hat{z}). \quad (2)$$

Nakonec je nutný převod do cylindrických souřadnic snímku:

$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c), \quad (3)$$

kde f značí ohniskovou vzdálenost v pixelech.

V implementaci je počítáno s fixní ohniskovou vzdáleností, kterou je potřeba před spuštěním nastavit. Lze ji vyčíst z EXIF informací snímků. Pokud u snímků není uvedena, je potřeba ji odhadnout nebo spočítat podle vztahu [12]:

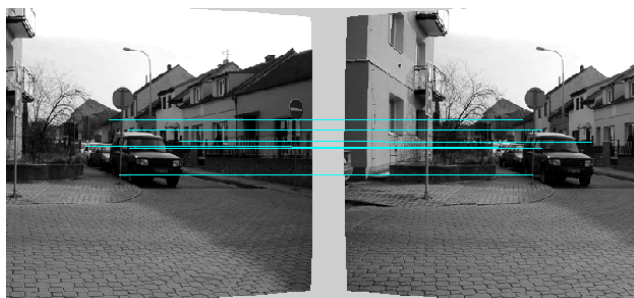
$$f_{\text{pixel}} = \frac{f_{\text{mm}}}{\text{Sensor_width}_{\text{mm}}} * I_{\text{width}_{\text{pixel}}}, \quad (4)$$

kde f_{mm} je ohnisková v milimetrech, $\text{Sensor_width}_{\text{mm}}$ je šířka senzoru kamery či fotoaparátu, kterým byl snímek pořízen a $I_{\text{width}_{\text{pixel}}}$ je šířka snímku v pixelech. Z videa by bylo možno ohniskovou vzdálenost dynamicky počítat, avšak takový výpočet je již nad rámec této práce.

2.3 Klíčové body a deskriptory

Aby bylo možné spojit jednotlivé snímky do jednoho panoramatu, je potřeba nalézt klíčové body a deskriptory mezi jednotlivými dvojicemi snímků. K tomuto slouží algoritmy SIFT (*Scale Invariant Feature Transform*) a SURF (*Speeded Up Robust Features*). Pro správné nalezení klíčových bodů, deskriptorů a jejich vzájemné polohy je doporučené překrytí jednotlivých snímků alespoň 25% [13].

SIFT byl poprvé publikován Davidem Lowem v r. 1999. Mezi hlavní výhody tohoto algoritmu patří nezávislost na měřítku, změně snímáního úhlu, velká odolnost vůči šumu a změně osvětlení [9]. David Lowe rozděluje SIFT do čtyř částí, na detekci extrémů, zpřesnění lokalizace významných bodů, rozhodnutí o orientaci bodů a tvorbu deskriptorů. Dvojici snímků a odpovídající detekci SIFT klíčových bodů a deskriptorů vidíme na obrázku 5. Klíčových bodů mezi dvojicí snímků bývá většinou nalezeno mnohonásobně více než vidíme na obrázku 5, na něm je jich pro přehlednost zobrazeno jen šest.



Obrázek 5. Shody mezi dvojicí snímků.

Novější obdoba algoritmu SIFT je metoda SURF, která byla popsána r. 2006 Herbertem Bayem. Hlavní výhodou tohoto algoritmu je rychlost, které autoři dosáhli především použitím integrálního obrazu [14].

2.4 Homografie

Po nalezení klíčových bodů a deskriptorů, které popisují jednotlivé snímky, je potřeba nalézt jejich vzájemnou polohu. K tomuto účelu slouží algoritmus RANSAC (*RANdom Sample Consensus*), který na základě podobnostních metod a pomocí výpočtu matice homografie nalezne nejpravděpodobnější vzájemnou souvislost mezi významnými body. Matice homografie tuto souvislost popisuje.

RANSAC, jak již vyplývá z názvu vybírá pro výpočet matematického modelu náhodně n hodnot z celkového počtu zadaných, proto je vhodnější mít k dispozici co nejvíce přesně stanovených hodnot pro nalezení co nejpresnějšího modelu. V souvislosti s hledáním tohoto modelu jsou definovány pojmy *inliers* a *outliers*, kdy *inliers* jsou body odpovídající matematickému modelu, *outliers* nikoli.

2.5 Spojování snímků

Finálním krokem k sestavení statického panoramatu je samotné spojování snímků a vyhlazení přechodů mezi nimi.

V implementaci při spojování bereme první čtvrtinu z prvního snímku, zbylou část ze snímku druhého. Experimentálně tento postup poskytoval nejlepší výsledky. Okraje připojovaného snímku váhujeme, aby byl spoj plynulejší. Důležitou roli hraje váhovaná šířka okraje, jak vidíme na obrázcích 6 a 7, kde v prvním příkladu jsou přechody mnohem více zřetelné než v příkladě druhém.



Obrázek 6. Váhování 5pix okraje, přechody mezi spojovanými snímky jsou zřetelnější.



Obrázek 7. Váhování 50pix okraje, přechod již není tak zřetelný.

Při váhování přechodů bylo potřeba váhovat i pravý okraj snímku, který při překreslování zprava doleva vytvářel v panoramatu výrazné artefakty.

3. Experimentální výsledky

Pro testovací účely byla natočena sada videí různých scén a časových rozsahů, jak je shrnuto v tabulce 1.

Tabulka 1. Testovací sady videí

Název	Čas [s]	f [pix]
klidná ulice	76	677
koleje	72	488
křižovatka	45	488
nádvoří FIT	333	488
Palackého vrch 1	111	488
Palackého vrch 2	80	488
park	529	488
Štramberská Trúba	276	488
Moravská brána	375	488

Testovací sady byly natočeny digitální zrcadlovkou Nikon D5100 (velikost snímače $23.6 \times 15.6\text{mm}$) s objektivem NIKKOR 18-105mm F3.5-5.6 ED a s využitím

video stativu s otočnou hlavou. Všechna videa mají rozlišení 1280×720 s frekvencí 25 snímků za sekundu. Při zpracování by však snímky měly zůstat přiměřené velikosti, např. 640×360 , pro větší snímky je delší. U takto velkých snímků trvá přidání dalšího snímku do panoramatu přibližně 14 sekund.

Z každé testovací sady bylo vytvořeno panorama, poté AVI video za použití vestavěné funkce v prostředí *MATLAB*. Některá z vytvořených panoramat vidíme na následujících obrázcích 8, 9, 10, 11 a 12. Ze všech testovacích sad vznikla kvalitní živá panoramata.

V některých případech však překreslování není úplně dokonalé a v aktuálním běhu kamery se může stát, že na scéně zůstane část překreslovaného pohybujícího se objektu, jak vidíme na obrázku 13. V následujícím běhu se však tato scéna opět překreslí. Vznik takovýchto artefaktů se dá odstranit vhodným nastavením velikosti kroku.

4. Závěr

Ačkoli tvorba živých panoramat není úplně jednoduchá, ukázali jsme, že je řešitelná. Využili jsme metody založené na známých poznatcích pro tvorbu statických panoramat ([2], [3], [6]), především na algoritmech SIFT [9], SURF a RANSAC [10].

Naše řešení poskytuje inovativní možnost využití uživatelských multimediálních dat (např. prezentace obcí, informace ze sjezdovek atd.). Jak ukázaly experimentální výsledky, metoda poskytuje kvalitní výsledky, avšak v některých ohledech má ještě prostor pro zlepšení. Například v automatickém hledání ohniskové vzdálenosti, zvýšení rychlosti výpočtu či tvorba uživatelského rozhraní.

Literatura

- [1] Yalin Xiong and Ken Turkowski. Registration, calibration and blending in creating high quality panoramas. In *Fourth IEEE Workshop on Applications of Computer Vision*, 1998.
- [2] Richard Szeliski. *Computer vision : Algorithms and Applications*. Texts in computer science. Springer, New York, London, 2010. <http://szeliski.org/Book/>.
- [3] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, August 2007. <http://dx.doi.org/10.1007/s11263-006-0002-3>.
- [4] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. <http://dx.doi.org/10.1145/258734.258861>.
- [5] Autopano. online, 2015. <http://www.kolor.com/autopano/>.
- [6] Varghese Mathew. Panoramic image stitching. online, August 2008. <http://pages.cs.wisc.edu/~vmathew/>.
- [7] Vytvoreni panoramatickych obrazu pomoci funkce photomerge. online. <https://helpx.adobe.com/cz/photoshop/using/create-panoramic-images-photomerge.html>.
- [8] Katie Dektar. Cylindrical panoramas. online, 2010. <http://graphics.stanford.edu/courses/cs178/applets/projection.html>.
- [9] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, Washington, DC, USA, 1999. IEEE Computer Society. <http://dl.acm.org/citation.cfm?id=850924.851523>.
- [10] Konstantinos G. Derpanis. Overview of the ransac algorithm. online, May 2010. http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf.
- [11] Autopano giga, panorama editor, projections. online, 2004 – 2014. http://www.kolor.com/wiki-en/action/view/Autopano_Giga_-_Panorama_Editor_-_Projections.
- [12] Noah Snavely. Estimating the focal length of a photo from exif tags. online, January 2008. <http://phototour.cs.washington.edu/focal.html>.
- [13] Robert Thomas Technologies. Landscape photography. blogpost (english). <http://photoblogstop.com/photography/landscape-photography>.
- [14] LucVan Gool Herbert Bay, Tinne Tuytelaars. Surf: Speeded up robust features. article, 2006. <http://www.vision.ee.ethz.ch/surf/eccv06.pdf>.



Obrázek 8. Panorama "klidná ulice".



Obrázek 9. Panorama "křižovatka".



Obrázek 10. Panorama "Palackého vrch 1".



Obrázek 11. Panorama "Palackého vrch 2".



Obrázek 12. Panorama Štramberk Trúba.



Obrázek 13. Artefakt při překreslování scény.