

STABILIZACE VIDEO

15. května 2014

Jan Bednařík (xbedna45)
Jakub Kvita (xkvita00)

1 Úvod

Potřeba stabilizovat snímky videa vyvstává tehdy, pokud dochází k nežádoucímu pohybu kamery vůči zaznamenávané scéně. Příkladem mohou být video přenosy z dálkově ovládaných mobilních robotických sond, amatérská videa pořizovaná "z ruky" bez stativu nebo POV¹ sportovní záběry.

Úkolem softwarového stabilizátoru je tento nechtěný pohyb, který lze typicky charakterizovat jako nahodilé rychlé kmitání v rovině kolmé k optické ose či drobné rotace okolo optické osy, odhadnout a kompenzovat tak, aby byly statické objekty scény neustále promítány na stejné pozice.

Tato dokumentace popisuje návrh a implementaci softwarového stabilizátoru stavějícím na kombinaci existujících postupů, jež jsou základem mj. pro současná komerční řešení. Sekce 3 shrnuje nejčastější vady videa potřebujícího stabilizaci, sekce 4 vysvětluje principy využitých postupů, sekce 5 blíže nastiňuje implementační detaily a výsledky experimentů nakonec shrnuje sekce 6.

2 Zadání

Program i dokumentace je výsledkem práce dvojice studentů na semestrálním projektu předmětu Zpracování obrazu. Text zadání zněl následovně:

"Navrhněte a implementujte nástroj pro stabilizaci videa (např. video natočené za chůze z ruky). Využijte metody pro zpracování obrazu a videa (sledování významných bodů, výpočet optického toku, MeanShift algoritmus, apod.). Srovnajte výsledky z pohledu rychlosti, přesnosti, kvality stabilizace atd."

3 Poškození videa

Vady videosekvence vzniklé vlivem neúmyslného pohybu kamery při pořizování záznamu lze popsat jako afinní transformaci poškozeného snímku vůči snímku referenčnímu. Pro specifické běžné případy natáčení videa lze sledovat typické vady:

natáčení statického obrazu

- vada: třes

¹z angl. Point of View - záběr simulující vlastní pohled dané osoby.

- **kompence:** translace

natáčení za chůze vpřed

- **vada:** třes, otáčení
- **kompence:** translace, rotace

natáčení za chůze do strany

- **vada:** třes, otáčení, parallax
- **kompence:** translace, rotace, odhad globálního pohybu kamery

4 Návrh stabilizátorů

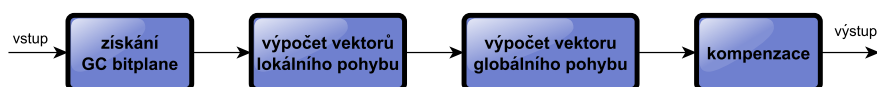
V současnosti existuje celá řada přístupů pro stabilizaci videa, jež se liší v **množství podporovaných geometrických transformací** pro kompenzaci a **rozlišení** (ve zlomcích pixelů) [1]. Výběr metody pak závisí především na míře poškození videa. Přesnější algoritmy podporující kompenzaci většího množství geometrických transformací jsou obecně náročnější na výpočetní výkon a nehodí se tak například pro real-time aplikaci v mobilních zařízeních.

Program `dis` implementuje celkem dva odlišné (parametrizovatelné) přístupy, díky čemuž bylo možné metody porovnat a rovněž uživateli nabídnout možnost výběru podle požadovaného využití. Následující sekce obě metody přibližují.

4.1 Gray-Coded Bit Plane Matching

Princip

Metoda Gray-Coded Bit Plane Matching staví na porovnávání bitových ploch po sobě jdoucích snímků videa a hledání nejpravděpodobnějšího vzájemného posunu. Pro každé dva snímky metoda prochází čtveřicí základních funkčních bloků (viz obrázek 1), jejichž funkce je popsána v následujících odstavcích.

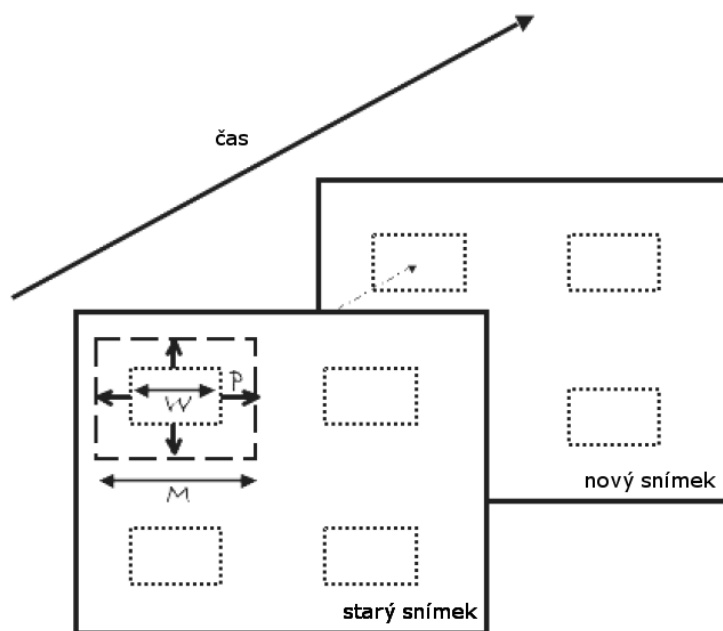


Obrázek 1: Blokové schéma stabilizátoru využívajícího metodu GC-BPM.

Bitovou plochu daného snímku lze reprezentovat jako matici jednobitových hodnot, již lze získat ze vstupního šedotónového snímku iterativní aplikací bitové operace XOR na jednotlivé bity pixelů od MSB² po LSB² (vztahy viz [1]).

Jakmile je k dispozici bitová plocha pro současný a předchozí snímek, spustí se proces hledání vektorů lokálního posunu (viz 2). Bitová plocha je rozdělena na několik shodně velkých částí (ve výchozím nastavení 4 části), kdy nad každou probíhá nezávisle výpočet jednoho vektoru lokálního posunu. Vyhledávací okno W v předchozím snímku se iterativně posouvá v rámci rozsahu p a porovnává se s s fixně umístěným vyhledávacím oknem W' stejné velikosti v aktuálním snímku pomocí vztahu:

²MSB - Most Significant Bit, LSB - Least Significant Bit



Obrázek 2: Dvě po sobě jdoucí bitové plochy. Zdroj [1]

$$\varepsilon(m, n) = \sum_w W \oplus W', \quad (1)$$

kde m, n značí aktuální posun vyhledávacího okna W a \oplus značí binární operaci **xor**. Úkolem je minimalizovat chybovou funkci $\varepsilon(m, n)$ a získat tak vektor lokálního posunutí.

Vektor globálního posunutí se následně získá jako medián nově spočítaných vektorů lokálního posunu a předchozího vektoru globálního posunu. Aktuální snímek je následně transformován pomocí matice translace (obsahující nový vektor globálního posunutí).

Kompenzace

Metoda podporuje kompenzaci **translace**.

Výhody

Protože algoritmus staví převážně na bitových operacích **xor**, je snadno implementovatelný přímo v hardwaru, dobře paralelizovatelný (hledání všech vektorů lokálního posunutí probíhá zcela nezávisle) a je tedy potenciálně velmi rychlý, díky čemuž by mohl být používán například v mobilních zařízeních, fotoaparátech apod.

Nevýhody

Při pouhém softwarovém řešení na běžném CPU a bez paralelizace se žel ukázal jako pomalejší, než jiné robustnější algoritmy (viz kapitolu 6). Protože kompenzuje pouze rotace, hodí se pouze na stabilizaci statických záběrů či videa natáčeného při chůzi vpřed.

4.2 KLT tracker a SVD

Princip

Druhou implementovanou metodu lze z hlediska výsledků považovat za robustnější. Jádrem celého algoritmu je hledání a sledování významných bodů v obraze a následný odhad afinní transformace skládající se výhradně z translace a rotace.

Opět se pracuje pouze s dvojicí po sobě jdoucích snímků s tím, že jsou porovnávány pozice vyhledaných významných bodů - příznaků. Za významné body jsou považovány rohy v obraze, jež jsou získávány pomocí standardního postupu good features to track [3]. Ty jsou následně mezi snímky sledovány pomocí pyramidové implementace metody Lukas-Kanade pro výpočet řídkého pole optického toku. Obrázek 3 demonstruje kompenzaci transformace rotovaného a posunutého snímku vůči referenčnímu snímku včetně zobrazení významných bodů.



Obrázek 3: Výsledek stabilizace dvou po sobě jdoucích snímků. V horním řádku jsou vyobrazeny snímky před stabilizací, kdy je druhý snímek oproti prvnímu posunutý a rotovaný. Dolní řádek ukazuje vyhledané příznaky a druhý snímek po stabilizaci.

Vyhledávání významných bodů je výpočetně poměrně náročné. V závislosti na povaze videa se v průběhu sekvence snímků sledované body ztrácejí (např. kamera se pohybuje kupředu, mění se okolní prostředí) a je nutné body opět vyhledat. K problému přistupujeme tak, že je sledování významných bodů inicializováno pouze tehdy, když počet aktuálně sledovaných bodů klesne pod $1/4$ (hodnota byla stanovena experimentálně) původně vyhledaných bodů, díky čemuž lze výpočet urychlit.

Odhad transformace mezi odpovídajícími si body je potom získán za pomoci algoritmu singular value decomposition, díky němuž lze získat samostatnou transformační matici pro translaci i rotaci.

Kompenzace

Metoda podporuje kompenzaci **translace** a **rotace**. V současné implementaci je podporován také odhad dlouhodobého pohybu kamery, díky čemuž je algoritmus dobře použitelný i pro natáčení videa při chůzi do strany (či natáčení panoramatických pohledů).

Výhody

Pro většinu běžných amatérských videí je algoritmus poměrně robustní a lze pomocí něj video s výhodou stabilizovat. Současná implementace nicméně nepodporuje ořezání výsledného obrazu.

Nevýhody

Algoritmus neběží v reálném čase a stále není natolik robustní, aby dovedl perfektně kompenzovat pohyb ve videu, v němž dochází k velice rychlým pohybům. Problémy mu činí také pohyb ve scéně a fenomén paralaxy.

4.3 Simulace dlouhodobého pohybu kamery

Ze své podstaty se všechny používané metody chovají tak, že se snaží pozici aktuálního snímku ve výsledku skrze všechny předchozí použité transformace kompenzovat vůči prvotnímu referenčnímu snímku. To nevadí v případě statického záznamu či natáčení videa při chůzi vpřed. Při pořizování panoramatických pohledů či záznamu videa při chůzi do strany se však již nebudou zobrazovat ty snímky, které se posunou za hranici původního snímku.

Tento problém lze řešit simulací žádoucího pohybu kamery, který tedy nebude ve výsledku kompenzován, nýbrž bude tolerován. Aby výsledné video působilo plynulým dojmem, je vhodné odhadovaný pohyb kamery specificky omezit. Současná state-of-the-art implementace stabilizátoru, již využívá online editor videí webové služby YouTube.com, staví na třech druzích pohybu: statické pozici kamery a lineárním a parabolickým pohybu [2].

Náš stabilizátor se tímto přístupem inspiroje, nicméně jej zjednodušuje a využívá simulaci lineárního pohybu kamery. Toto vylepšení v současné implementaci podporuje pouze stabilizátor *KLT tracker a SVD* (viz sekci 4.2)

5 Programová realizace

Program `dis` je implementován jako konzolová aplikace, jež přijímá vstupní video a produkuje výstupní stabilizované video. Výběr stabilizační metody včetně její parametrizace a výstupního formátu videa lze ovlivnit pomocí parametrů shrnutých v sekci A.

Jako implementační jazyk byl s ohledem na rychlost běhu zvolen **C++** společně s knihovnou **OpenCV**. Při vývoji jsme narazili na problémy s kódováním videa, výsledkem čehož je zaručená podpora pouze pro video kódované kodekem **Xvid**.

5.1 Třídy

Třída Stabilizer: Soubor `Stabilizer.h` nese abstraktní bázovou třídu pro oba implementované stabilizátory.

Třída StabilizerGCBPM: Soubory `StabilizerGCBPM.cpp` a `StabilizerGCBPM.h` nesou implementaci stabilizátoru *Gray-Coded Bit Plane Matching*.

Třída StabilizerSVD: Soubory `StabilizerSVD.cpp` a `StabilizerSVD.h` implementují stabilizátor *KLT tracker* a *SVD*. Pro výpočet významných bodů je využita OpenCV funkce `goodFeaturesToTrack`, pro výpočet optického toku funkce `calcOpticalFlowPyrLK` a pro výpočet SVD funkce `SVD`.

5.2 Výstupní video

Výstupní video může nabývat tří různých forem podle uživatelské volby (viz sekci [A](#)).

RAW: Výstupní snímky mají stejnou velikost jako vstupní. Pokud se vstupní dostane vlivem stabilizace částečně mimo toto okno, prázdná místa jsou nahrazena černou barvou.



Obrázek 4: Výstupní snímek typu `raw`.

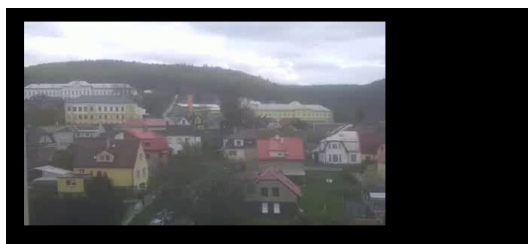
INTERSECTION: Vstupní snímky jsou stabilizací ořezány tak, aby na výstupu nedocházelo ke zobrazování černých okrajů. Rozměry snímků výstupního videa se tak dopočítají až za běhu podle míry translace ve videu.



Obrázek 5: Výstupní snímek typu `intersection`.

UNION: Velikost výstupních snímků se přizpůsobí tak, aby se zobrazili všechny stabilizované snímky nezávisle na použité transformaci. Snímky výsledného videa jsou tak typicky výrazně větší, než snímky vstupní.

Nutno zmínit, že tyto volby při současné implementaci podporuje pouze stabilizátor `GC-BPM`.



Obrázek 6: Výstupní snímek typu **union**.

6 Experimenty

Úspěšnost stabilizace závisí ve velké míře na povaze vstupního videa. Sada 20 testovacích videí byla proto rozdělena do skupin podle toho, zda kamera zachycuje statickou scénu, pohybuje se vpřed (při běhu či chůzi), nebo zda byl pořizován panoramatický záběr. Všechny případy jsou dále kombinovány s úmyslným zahrnutím otřesů a rotací. Výsledky reprezentované slovním popisem subjektivních dojmů z výstupních stabilizovaných videí jsou shrnuty v tabulce 1.

Tabulka 1: Výsledky experimentů

		GC-BPM	KLT a SVD
statický záběr	posun	Dobře kompenzuje nižší i vyšší třes, zanechává pouze malé záchvěvy, ale problém nastává u snímků s velkými jednotvárnými oblastmi, jako je např. obloha.	I nejvyšší třes kompenzuje výborně, objekty zůstávají zcela na fixních pozicích.
	rotace	/	Výborná kompenzace. Při dlouho trvajících rotacích (neustále se otáčející kamera) se kumuluje chyba a rotace není kompenzována dostatečně.
chůze vpřed	posun	Podobně jako u statického záběru kompenzuje pohyb dobře, ale přestává fungovat u snímků s jednotvárnými plochami.	Kompenzuje poměrně dobře, ale u některých záběrů se po čase vlivem kumulace objeví nežádoucí rotace.
	rotace	/	Stejný závěr jako pro posun.
běh	posun	Stejné chování jako u chůze, pouze se ve výsledku více projevuje nežádoucí záchvěvy.	Nežádoucí rotace se objevuje dříve a ve vyšší míře, než u chůze.
	rotace	/	Stejný závěr jako u posunu.
panorama	posun	Při pomalém pohybu kompenzuje dobře, při vyšších rychlostech snímek 'mizí' za okraj obrazovky.	Výborná kompenzace.
	rotace	/	Výborná kompenzace.
FPS		2,23	9,53

7 Závěr

V dokumentaci byl nastíněn princip a implementační detaily vlastního softwarového řešení digitálního stabilizátoru obrazu. Výsledný program, jež zahrnuje dvě nezávislé implementace dvojice odlišných stabilizačních metod, byl otestován na sadě testovacích videí, jež odhalily slabé i silné stránky obou metod. Metoda KLT a SVD se jeví jako robustnější ve srovnání s metodou GC-BPM, neboť dovede kompenzovat kromě posunů i rotaci a díky sledování významných bodů dosahuje lepších výsledků také při dlouhodobém pohybu kamer (vpřed, do strany). Při dané implementaci je rovněž druhá metoda více než čtyřnásobně rychlejší.

Bohužel ani jedna z metod nedosahuje dostatečné robustnosti například pro komerční nasazení softwaru, neboť v mnohých případech selhává. Jak možné rozšíření potenciálně vedoucí k lepším výsledkům se nabízí sledování pohybujících se objektů ve scéně tak, aby neovlivňovaly výsledek stabilizace, filtrování nalezených významných bodů tak, aby byly použity pouze relevantní z nich (například pomocí metody RANSAC) nebo propracovanější odhad a aproximace dlouhodobého pohybu kamery.

Reference

- [1] Brooks, A. C.: Real-Time Digital Image Stabilization. *Image Processing Computer Project Final Paper*, 2003.
- [2] Grundmann, M.; Kwatra, V.; Essa, I.: Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [3] Shi, J.; Tomasi, C.: Good Features to Track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, s. 593 – 600.

A Spuštění programu

SYNOPSIS

```
dis -i jmeno -o jmeno [volby]
```

VOLBY

```
-i jmeno
    jmeno vstupního videa

-o jmeno
    jmeno výstupního videa

-s stabilizator
    nastaví metodu stabilizace. Dostupné metody jsou:

    gcbpm (výchozí)
    svd

-m metoda
    nastaví typ výstupního videa. Dostupné metody jsou:

    raw (výchozí)
        velikost snímku vstupního a výstupního videa je stejná
    intersection
        velikost snímku výstupního videa odpovídá průniku všech snímku
    union
        zobrazeny jsou všechny snímky, tudíž je velikost snímku
        výstupního videa větší

-l
    pokud byla zvolena metoda 'union' (viz volbu -m), každý snímek
    je ponechán ve výstupním videu, tedy nové snímky jsou zobrazeny
    na staré.
```

VOLBY relevantní pro stabilizator 'gcbpm'

```
-b bit
    nastaví n-ty bit pro použití při výpočtu bitových ploch. Povolene
    hodnoty: 0-8 (výchozí hodnota: 5).

-N velikost
    nastaví velikost okna pro porovnání (čtvercové okno velikost
    x velikost) v pixelech. Vyšší hodnota zapriciní vyšší kvalitu
    stabilizace ale delší běh. Pokud je nastavena hodnota příliš
    vysoká, bude automaticky dopočítána nejvyšší možná vyšší.

-f offset
    nastaví offset (v x i y souřadnici), v jehož rámci se bude
    pohybovat okno pro vyhledávání. Vyšší hodnota zapriciní vyšší
    kvalitu stabilizace ale delší běh. Pokud je nastavena hodnota
```

prilis vysoka, bude automaticky dopocitana nejvyssi mozna vyssi.

-x podoblasti

nastavi pocet podoblasti v ose x pro hledani vektoru lokalniho posunu (vychozi hodnota: 2).

-y podoblasti

nastavi pocet podoblasti v ose y pro hledani vektoru lokalniho posunu (vychozi hodnota: 2).

-d utlum

nastavi utlumeni predchoziho vektoru globalniho posunu. Vyssi hodnota zajisti lepsi stabilizaci, ale vysledny snimek s vyssi pravdepodobnosti kvuli akumulaci chyb 'opusti' okno. Povolene hodnoty (float): 0-1 (vychozi hodnota: 0,95)

VOLBY relevantni pro stabilizator 'svd'

-c body

nastavi pocet vyznamnych bodu pro sledovani.