

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ČASOSBĚRNÉ VIDEO

BAKALÁŘSKÁ PRÁCE

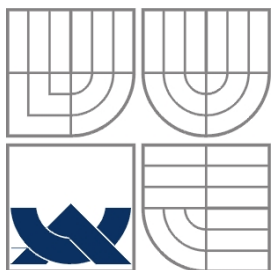
BACHELOR'S THESIS

AUTOR PRÁCE

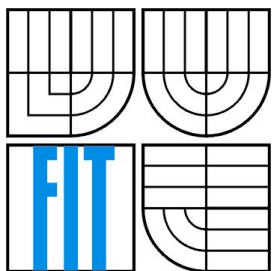
AUTHOR

Matěj Kubiš

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ČASOSBĚRNÉ VIDEO

TIMELAPSE VIDEO

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Matěj Kubiš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Adam Herout, Ph.D.

BRNO 2010

Abstrakt

Tato práce zahrnuje tvorbu grafického uživatelského rozhraní pro správu sady fotografií, její navigaci a přehrávání a následný export do videoformátu. Studuje problematiku časosběrného videa a řeší implementaci algoritmů zabývajících se odstraňováním nežádoucích jevů vznikajících při jeho tvorbě. Konkrétně zpracovává interpolaci kolísavého jasu v sekvenci a stabilizaci snímků při fotografování bez stativu.

Abstract

This thesis includes the creation of a graphic user interface for the management of a set of photographs, its navigation, playback and subsequent export into videoformat. It studies the problems of timelapse video and deals with the implementation of algorithms concerning the removal of undesirable effects rising during the process of its creation. To be specific, it processes the interpolation of fluctuating levels of brightness in the sequence and the stabilization of image sequences acquired without a tripod.

Klíčová slova

časosběrné video, fotografie, zpracování obrazu, počítačové vidění, OpenCV, EmguCV, SURF, uživatelské rozhraní

Keywords

timelapse video, photography, image processing, computer vision, OpenCV, EmguCV, SURF, user interface

Citace

Matěj Kubiš: Časosběrné video, bakalářská práce, Brno, FIT VUT v Brně, 2010

Časosběrné video

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Adama Herouta, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Matěj Kubiš
17. května 2010

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Adamu Heroutovi, Ph.D. za jeho vedení a rady při vývoji aplikace a tvorbě této práce.

© Matěj Kubiš, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
1.1 Co je to časosběrné video.....	2
1.2 Problémy časosběrného videa.....	2
1.3 Cíl práce.....	3
2 Analýza.....	4
2.1 Cílová platforma.....	4
2.2 Funkce aplikace.....	4
2.3 Použité technologie.....	5
3 Návrh.....	9
3.1 Návrh grafického rozhraní.....	9
3.2 Návrh interpolace jasu.....	10
3.3 Návrh stabilizace.....	12
4 Implementace.....	16
4.1 Grafické rozhraní.....	16
4.2 Interpolace jasu.....	17
4.3 Stabilizace.....	20
4.4 Filtrace vadných shod.....	21
5 Zhodnocení výsledků.....	26
5.1 Funkční prvky rozhraní.....	26
5.2 Použitelnost interpolace jasu.....	27
5.3 Spolehlivost stabilizace.....	28
6 Závěr.....	30
Literatura.....	31
Seznam příloh.....	33

1 Úvod

Bakalářská práce zpracovává problematiku tvorby časosběrného videa takovým způsobem, aby byla dostupná běžným uživatelům. Zaměřuje se na správu fotografií, export do videosouboru a především se soustředí na metody a algoritmy odstraňující ze sekvencí rušivé vlivy, které jsou pro tvorbu časosběrného videa typické. Konkrétně se jedná o úpravu jasu při nežádoucích světelných změnách scény a stabilizaci sekvencí, které byly pořízeny bez pomoci stativu.

Vzhledem k faktu, že k tématu tvorby časosběrného videa prakticky neexistuje odborná literatura, vycházím já, jako autor práce, ze své vlastní několikaleté zkušenosti v této oblasti.

Práce je svoji strukturou rozdělena na analýzu, kde čtenář nalezne popis problematiky a podrobně stanovené cíle, návrh – směr, kterým se vývoj ubíral, implementaci, která popisuje postupy použité při vývoji a konečně zhodnocení výsledků, kam může čtenář nahlédnout, zajímá-li ho praktický přínos práce.

1.1 Co je to časosběrné video

Časosběrné video je sekvence snímků (nejčastěji fotografií), u které jsou intervaly pořízení mezi jednotlivými snímky řádově větší, než-li tomu je u klasické videokamery. Takové video je ve většině případů pořízeno klasickým fotoaparátem a zachycuje dlouhodobý děj způsobem, díky kterému je možné pozorovat například růst květu rostliny, pohyb mraků, stavbu budovy apod. v několika málo sekundách.

1.2 Problémy časosběrného videa

Při fotografování za účelem pořízení datového materiálu k tvorbě časosběrného videa, stejně jako u každé činnosti zahrnující záznam obrazu, dochází k mnoha nežádoucím vlivům. Několik takových vlivů, se kterými se fotograf potýká před, při i po samotném pořízení materiálu, si uvedeme v následujících odstavcích.

Při periodickém fotografování jedné scény se občas, především tedy v exteriérech, stává, že se do zorného pole fotoaparátu dostane nežádoucí předmět, případně ho z větší části zakryje. V takovém případě je zapotřebí předmět odstranit, jinak dochází k narušení plynulého dění ve scéně.

Je zapotřebí si uvědomit, že při dlouhodobějším fotografování, jímž sběr dat za účelem tvorby časosběrného videa bezpochyby je, se ve scéně projevují vlivy, které ovlivňují světelné podmínky a z širšího hlediska znehodnocují videa subjektivními, přesto však nepříjemnými efekty. Jedná se především o „blikání“ výsledného videa způsobené krátkodobými změnami osvětlení mezi

jednotlivými snímky. Příčinou tohoto jevu může být například měnící se světelná intenzita slunce v závislosti na zakrytí mraky, zářivka napájená střídavým proudem nebo fotoaparát použitý při fotografování (v tomto případě pravděpodobně kompaktní), který je nastaven na částečnou, případně úplnou automatickou konfiguraci nastavení clony/závěrky/citlivosti.

Při fotografování se ve většině případů používá stativ nebo jiný způsob upevnění fotoaparátu. A to jak z důvodu časové náročnosti, tak z vlastní podstaty záznamu, kdy sebemenší pohyb fotoaparátu je nežádoucí a může podstatným způsobem negativně ovlivnit výsledek. Ne vždy však má fotograf stativ k dispozici a někdy je okolními podmínkami nucen pořídit záznam bez jakékoliv opory. To se samozřejmě ve výsledku projeví „skákáním“ obrazu a celkový dojem z takového videa je značně degradován.

1.3 Cíl práce

Vzhledem k volnosti zpracování tématu práce jsem se rozhodl, že cílem bude vytvořit aplikační software pro správu, manipulaci a editaci fotografií. A to hned z několika důvodů:

- Nedostatek volně dostupných nástrojů pro tyto konkrétní účely
- Potenciální užitečnost projektu i mimo studijní účely v rámci vývoje
- Snadné navázání dalšími moduly (např. v diplomové práci)

Program bude disponovat intuitivním grafickým rozhraním a bude automaticky (s možností interaktivního zásahu uživatele) upravovat fotografie v rámci odstraňování nežádoucích vlivů zmíněných v předchozí podkapitole.

2 Analýza

Tato kapitola pojednává o zvolených cílech práce a nastudování problémů s nimi spojenými. Popisuje počáteční směr vývoje a zdůvodňuje klíčová rozhodnutí.

2.1 Cílová platforma

S přihlédnutím k faktu, že mnoho výrobců fotoaparátů (Canon, Nikon, Olympus aj.) vyvíjejí software výhradně pro platformy MS Windows a Mac OS, bude projekt s přihlédnutím k tomuto faktu vyvíjen pro operační systém Windows. Jedním důvodem je většinový podíl jeho uživatelů a druhým, neméně závažným, je skutečnost, že nedisponují prostředky pro vývoj na platformě Mac OS. Program je vyvíjen konkrétně pod operačním systémem:

- Microsoft Windows 7 Professional 32-bit

Díky integrovanému rozhraní Microsoft .NET Framework v operačních systémech Windows Vista a novějších, tím pro uživatele zajištěním kompatibility bez nutnosti instalace dalších balíků, jsem toto rozhraní zvolil pro vývoj aplikace. .NET Framework, konkrétně zvolený programovací jazyk C#, poskytuje dostatečné prostředky pro tvorbu grafického uživatelského rozhraní.

2.2 Funkce aplikace

Z hlediska uživatelského rozhraní musí být aplikace přehledná a ovládání jednoduché a intuitivní. Jazykem ovládacích prvků bude angličtina. Aplikace se tímto krokem pokusí prosadit možnost oslovit větší množství uživatelů. Mezi funkce bude patřit:

- načtení fotografií z paměťového média (pevného disku počítače, paměťové karty)
- zobrazení náhledu
- přehrávání sekvence fotografií
- ukládání/otevírání projektů pro rychlejší opětovnou práci
- export sady fotografií do videoformátu
- použití algoritmů pro odstraňování nežádoucích prvků v sekvenci

V podkapitole 1.2 lze nalézt informaci o třech problémech při pořizování fotografií za účelem tvorby časosběrného videa. Z těchto tří bude aplikace disponovat algoritmy pro odstranění posledních

dvou (nežádoucí změna světelných podmínek a roztřesení obrazu). Algoritmus pro odstranění objektu ze sekvence by byl kontraproduktivní s principy tvorby časosběrného videa a v případě zakrytí neakceptovatelně velké části scény nežádoucím objektem lze fotografii ze sekvence jednoduše vyřadit.

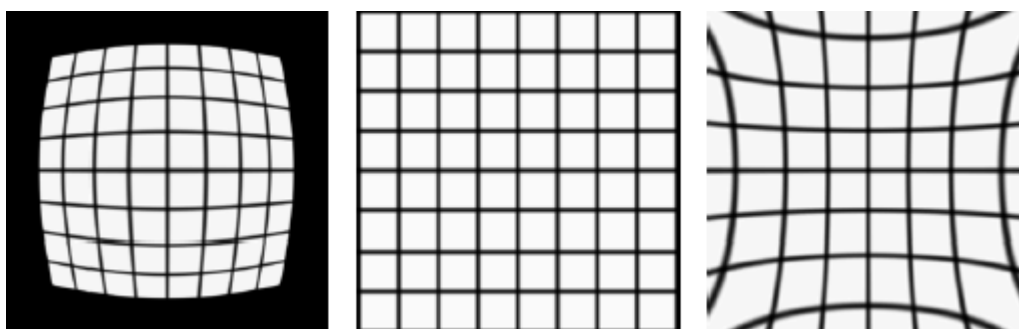
2.3 Použité technologie

Pro pořizování testovacích dat byl použit digitální fotoaparát Canon 450D s objektivem Canon EF 17-40mm f/4 L. V oblasti programového vybavení použitého pro vývoj je, kromě programovacího jazyku uvedeného v předchozí podkapitole, pro pokročilejší manipulaci s fotografiemi zapotřebí využít některou z volně dostupných knihoven pro zpracování obrazu a základními prvky počítačového vidění.

2.3.1 Digitální záznam obrazu

Pro naše účely si uvedeme několik principů a problémů, které jsou pro fotografování typické a přímo se týkají, či velice úzce souvisí se záznamem obrazu v digitální podobě za účelem tvorby časosběrného videa.

Prakticky všechna zařízení nižší třídy produkují v určité míře geometrické pokřivení obrazu způsobené zakřivením čočky. Ačkoliv mohou být vlastnosti takovýchto fotografií žádoucí v uměleckém směru, pro mnoho účelů, pro nás tedy především pro účel tvorby časosběrného videa, je třeba s těmito jevy počítat a eliminovat je (viz obr. č. 1). Fotoaparáty kompenzují tato zakřivení vlastní kalibrací obrazu v závislosti na podmínkách a vlastnostech čočky (ohnisková vzdálenost, aktuální hodnota přiblížení...), ne vždy je však kalibrace optimální a často je proto geometrie zaznamenaného obrazu zkreslená [1].

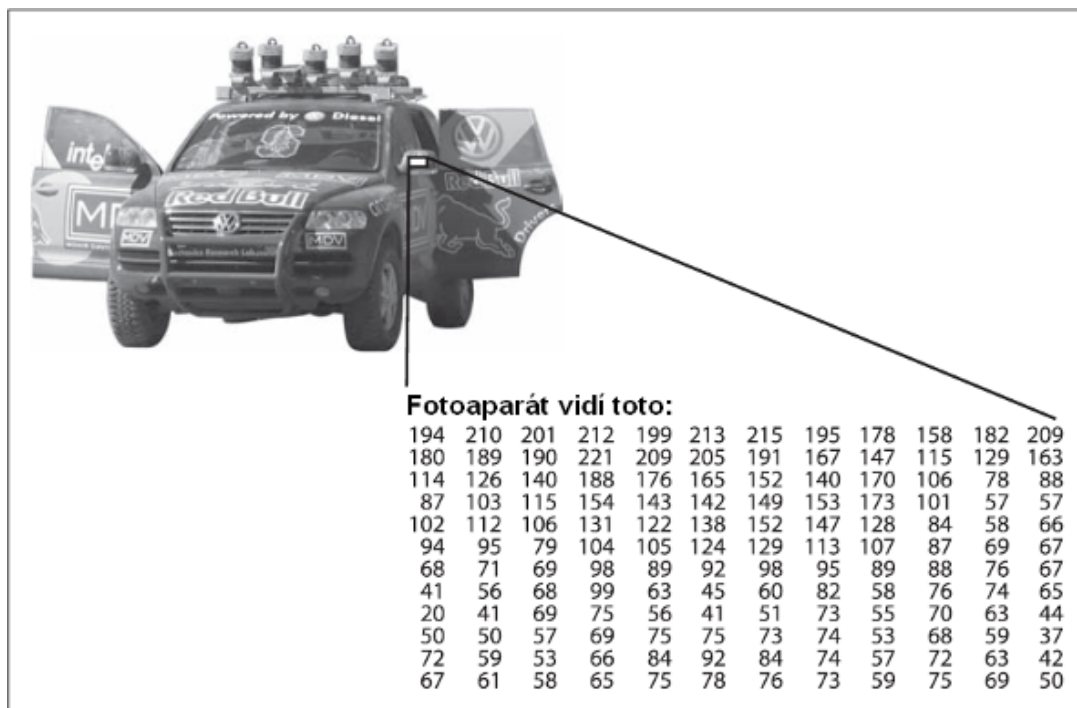


Obrázek 1: Uprostřed vidíme originální obraz; vlevo – sudovité zkreslení (barrel); vpravo poduškovité zkreslení (pincushion)

Na rozdíl od lidského mozku, kde „vnímání podnětů je ovlivněno subjektivními emočními vlastnostmi a motivy očekávání jedince“ [2], je vnímání barev obrazu fotoaparátem (potažmo jakýmkoliv zařízením pro záznam obrazu) objektivní a nezkrácené (ovšem pochopitelně v závislosti na kvalitě zpracování zařízení). V našem případě se jedná tedy především o nedokonalosti v oblasti vyvážení bílé barvy a celkové světlosti obrazu. Světelné podmínky scény a variabilita reprezentace takových podmínek jednotlivých zařízení má podstatný vliv na výsledný dojem výstupního videa.

Pro nás důležitým aspektem při srovnávání snímků různých fotoaparátů je součástka pro snímání světla. Dvěma dnes používanými senzory jsou CCD, který je výrobně starší a CMOS, který se jako světelný senzor používá teprve několik let. Liší se především ve způsobu záznamu informace o jednotlivých barevných složkách, výrobní cenou, produkovaným teplem, spotřebou elektrické energie, poměrem šumu v obraze, ale pro nás podstatnou informací je jejich rozdílná světelná citlivost, která se může projevit při porovnání dvou fotografií pořízených různým světelným senzorem za určitých podmínek.

O samotný převod signálu ze senzoru se stará AD převodník, který v závislosti na počtu kvantovacích úrovní určuje barevnou hloubku fotografie. Číslicový formát fotografie (viz obr. č. 2) umožňuje její následné zobrazení a použití v číslicových systémech, jakými je například osobní počítač. Digitální data jsou poté uložena na paměťovém médiu fotoaparátu v „syrovém“ formátu RAW a nebo přímo exportována do bezztrátového či ztrátového formátu, s kterými pracuje i tato



Obrázek 2: Pro počítač je boční zrcátko pouze matice čísel

aplikace. Proto i samotný formát, případně úroveň samotné komprese znatelně ovlivní výslednou fotografii [3].

Zatímco v případě krátkodobých sekvencí (řádově v hodinách) je běžné, že fotograf pořizuje snímky použitím jednoho zařízení, při sběru dat v rámci delšího časového intervalu (dny – tání sněhu; měsíce – stavba budovy; roky – stárnutí člověka) se běžně stává, že je vystřídáno několik různých zařízení (výměna fotoaparátu, na sběru se podílí více fotografů). Proto je důležité si uvědomit, že různé fotoaparáty se liší svými vlastnostmi způsobu záznamu a že fotografie totožné scény za stejných podmínek, pořízeny různými fotoaparáty, se mohou lišit rozdíly způsobenými právě těmito vlastnostmi. Je proto pochopitelné, že sekvence zaznamenané pomocí více zařízení, které by v případě jednoho zařízení produkovaly kvalitní video, působí rušivým dojmem a konečná kvalita výsledného videa je nižší.

2.3.2 Počítačové vidění

Na rozdíl od lidského mozku, kde identifikaci a interpretaci zrakového vjemu zajišťuje korová zraková oblast [4], je obrazová informace, zaznamenaná obrazovým záznamovým zařízením (fotoaparátem, videokamerou), v digitální podobě reprezentována pouze velkým množstvím čísel. Způsobem reprezentace a informacemi získanými z těchto dat se zabývá právě oblast počítačového vidění.

„Počítačové vidění je transformace dat z fotoaparátu nebo videokamery buďto v rozhodnutí nebo novou reprezentaci.“ [5]. Rozhodnutím může být například tvrzení, že objekt x se v obraze vyskytuje/nevyskytuje, případně na jakých souřadnicích. Pod novou reprezentací si lze představit například převod fotografie do stupňů šedi, nebo změnu jejího tvaru. Právě změnou velikosti a tvaru, obecně transformací fotografie se část této práce zabývá.

2.3.3 OpenCV

Open source projekt OpenCV vznikl v roce 1999 v laboratořích společnosti Intel za účelem vytvořit nejen základní, ale také optimalizovanou infrastrukturu pro vývoj v oblasti počítačového vidění. Projekt, díky kterému měla dlouhodobě vzrůst poptávka po výpočetním výkonu počítačů, se brzy, především díky masově se rozšiřujícím vícejádrovým procesorům, těšil široké členské komunitě, která později zcela převzala jeho vývoj.

Čistě pro zajímavost, v poslední době se v oboru počítačového vidění z hlediska výpočetního výkonu prosazují spíše grafické karty, viz [6].

V praxi se jedná o knihovnu napsanou v jazyce C a C++, fungující pod operačními systémy Linux, Windows a Mac OS X. Knihovna poskytuje více jak 500 optimalizovaných funkcí v oblasti počítačového vidění a podporuje využití vícejádrových procesorů.

2.3.4 EmguCV

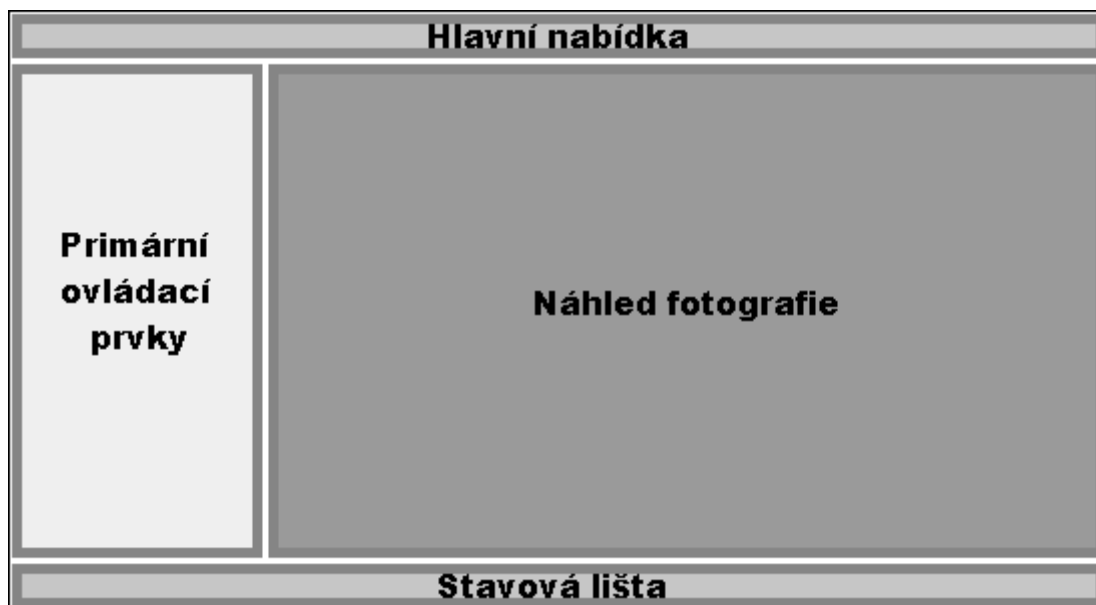
Jedná se o wrapper umožňující objektově orientované použití knihovny OpenCV ve frameworku .NET. Spolu s možností volání nativních funkcí knihovny OpenCV umožňuje EmguCV použití některých sofistikovaných vyšších funkcí a navíc poskytuje vlastní datovou strukturu (objekt) pro základní reprezentaci dat v počítačovém vidění – obrázek a matici – umožňující plné využití řízeného kódu frameworku .NET a s ním souvisejícího Garbage collectoru [7].

3 Návrh

Po důkladné analýze všech prvků aplikace a volbě vhodných prostředků k realizaci se tato kapitola pokusí objasnit konkrétní způsoby, metody a algoritmy, pomocí kterých bude aplikace realizována a pomocí kterých se pokusí dosáhnout všech stanovených cílů.

3.1 Návrh grafického rozhraní

Nejvýraznějším prvkem grafického uživatelského rozhraní aplikace pro práci s fotografiemi by měl být panel s náhledem pro snadnou identifikaci, orientaci a navigaci v rámci sady jakéhokoliv rozsahu. Této myšlence byl přizpůsoben celkový návrh rozložení prvků rozhraní. S narůstajícím trendem širokoúhlých zobrazovacích displejů roste jak poměr mezi jejich vlastníky a vlastníky klasických zařízení se stranami obrazovky v poměru 4:3, tak poměr mezi samotnými stranami zařízení. Z toho důvodu jsou hlavní ovládací prvky umístěny vertikálně – pro většinu uživatelů bude omezující rozměr obrazovky její výška. Krom relativně malého prostoru, který zabírá panel hlavní nabídky na horní straně okna a stavová lišta v dolní části, je celý vertikální prostor okna aplikace k dispozici panelu náhledu fotografie (viz obr. č. 3).



Obrázek 3: Rozložení prvků grafického uživatelského rozhraní aplikace

Z důvodu datového objemu (dnes běžná 12Mpx fotografie v bitmapovém formátu potřebuje více než 35MB paměti) aplikace při načtení fotografií uloží do paměti pouze náhled uživatelem

specifikované velikosti spolu s informací o umístění původní fotografie. Proces načítání a zmenšování většího množství fotografií může být časově náročný, proto aplikace umožní uložit celou načtenou sadu (náhledy snímků a informace o jejich attributech a lokaci) jako projekt, který lze později načíst ve zlomku původního času.

Vzhledem k povaze aplikace se pro řešení problému navigace mezi jednotlivými snímky nabízí použití posuvného prvku – pro uživatele je důležité jak pozorovat změny mezi jednotlivými snímky, tak průběh celého videa. Tímto způsobem má možnost plynule ovládat průběh, rychlost i směr změny zobrazovaného náhledu. Spolu s tímto základním prvkem nabízí program uživateli přehrávání celé sekvence (či její konkrétní části) konstantní rychlostí definovanou hodnotou snímků za vteřinu. Tímto způsobem je možné shlédnout celou sekvenci a zvolit nejvhodnější rychlost před tím, než provede uživatel export fotografií do videoformátu (což bývá velice časově náročný proces).

Spolu s exportem fotografií do formátu videa, který knihovna OpenCV podporuje s použitím všech dostupných kodeků v počítači uživatele, aplikace umožní uložení samotných modifikovaných fotografií na paměťové medium pro další zpracování (bez ztráty kvality umožní použít data například v profesionálním video editoru, kde je možné přidat zvukovou stopu, titulky, efekty apod.).

Samozřejmostí je implementace všech časově náročnějších operací v samostatném vlákně s indikací poměru hotové části.

3.2 Návrh interpolace jasu

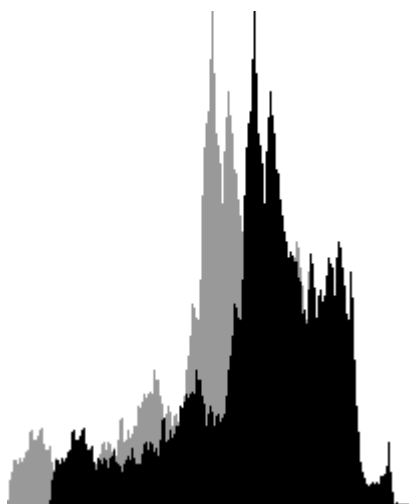
V sadě fotografií dochází ke změně světelných podmínek jak v žádoucích (fotografování východu/západu Slunce), tak nežádoucích případech (zmíněných v kapitole 1.2). Při tvorbě algoritmu pro odstranění nežádoucích případů je proto zapotřebí postupovat způsobem, kdy nehrozí narušení dlouhodobějších světelných změn, které fotograf odstranit nechce, ba jsou dokonce důvodem tvorby samotného videa. Z tohoto důvodu nelze pro úpravu fotografií použít střední hodnotu jasu získanou ze všech fotografií v sekvenci (viz obr. č. 6), ale postupovat po úsecích v závislosti na relativní pozici fotografie. Ideální řešení nabízí klouzající okno (*sliding window*). Použitím klouzajícího okna je eliminováno nežádoucí odstranění dynamických světelných přechodů a zachována schopnost interpolovat hodnotu jasu v rámci předem definované (případně upravené uživatelem) velikosti okna.

Fotografie je tvořena mnoha pixely, jejichž vzájemné barevné poměry tvoří celkový obraz. Nelze proto pracovat s jednotlivými pixely (změnil by se poměr barev ve fotografii), ale je nutno pracovat s intenzitou snímku jako celku (o důsledcích tohoto problému viz kapitola 5.2).

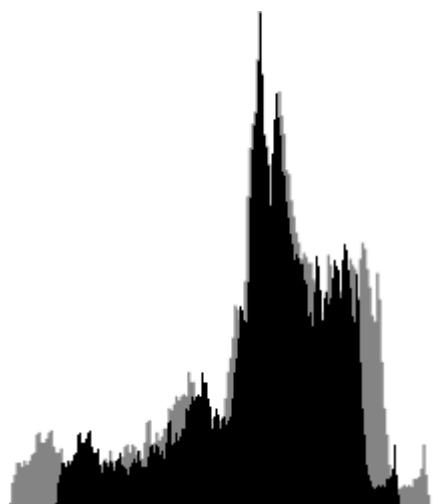
Z podstaty principu digitálního videozáznamu (konkrétně časosběrného videa) vychází předpoklad, že spolu sousedící snímky jsou z hlediska poměru hodnot barev obrazu (jednotlivých barevných kanálů) velice podobné. Změna jasu obrazu se obecně provádí přičtením hodnoty ke všem

pixelům všech kanálů obrazu (zachování barevného poměru). Prakticky se jedná o posunutí histogramu jasu (viz obr. č. 4).

Další možností úpravy světelných podmínek je nikoliv posun histogramu, ale jeho roztažení (viz obr. č. 5). V tomto případě se namísto jasu obrazu upravuje kontrast. Vzhledem k individuálním



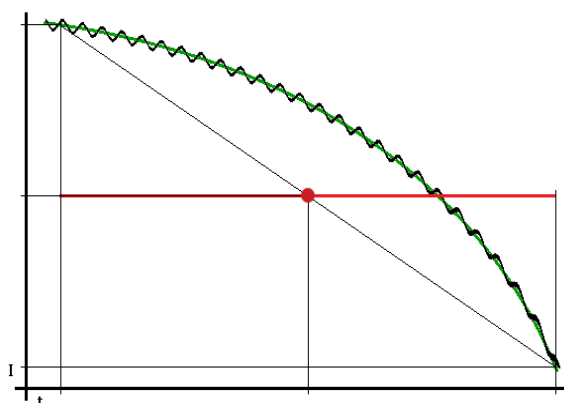
*Obrázek 4: Histogram jasu:
posunutí*



*Obrázek 5: Histogram kontrastu:
roztážení*

světelným vlastnostem scény může tato metoda přinést leckdy uspokojivější výsledky – difference úrovně světlosti sousedních snímků nebývá velká.

Jako třetí možnost pro interpolaci jasu bude mít uživatel možnost použít metodu s využitím principu *alpha blend*.. Jedná se o „prolnutí“ jednotlivých snímků s využitím transparency. Alpha



Obrázek 6: Ilustrační graf intenzity sekvence fotografií západu Slunce: černá představuje zaznamenanou intenzitu snímků; zelená hledanou interpolační křivku; červená interpolační přímkou při použití střední hodnoty (aritmetického průměru)

blending není obvykle používán pro tento účel a proto lze metodu s úspěchem použít pouze za specifických podmínek:

- Pohyb objektů v sekvenci je dostatečně malý/pomalý (dynamičnost scény)
- Klouzající okno je dostatečně malé (v závislosti na dynamičnosti snímků)

V případě nedodržení podmínek v dostatečné míře lze očekávat (vzhledem k povaze metody), že výsledná sekvence bude působit rozmazaně, případně se budou v obraze objevovat „duchové“.

3.3 Návrh stabilizace

Problematika stabilizace obrazu je značně netriviální a oblast počítačového vidění se jí zabývá již řadu let. Díky pokroku v oblasti výpočetního výkonu počítačů a kvalitě záznamu videokamer a jejich cenové dostupnosti se stabilizace obrazu dostala na úroveň, která dovoluje současným zařízením stabilizaci v reálném čase.

Stabilizace sekvence snímků spočívá v principu transformace jednoho snímku co možná nejpřesněji do snímku druhého. Prakticky se jedná o snahu překrývání odpovídajících částí obrazu. Celý proces se dá rozdělit do několika částí, kterým se práce věnuje v následujících podkapitolách:

1. Detekce význačných bodů (*features*) v obraze A , který je označen jako původní a který bude cílem transformace následujícího snímku B
2. Nalezení klíčových bodů v obraze B
3. Přiřazení význačných bodů obrazu A k odpovídajícím bodům obrazu B
4. Eliminace vadných shod
5. Vytvoření transformační matice z rozdílů pozic jednotlivých dvojic bodů obrazu A a B
6. Transformace obrazu B do pozice obrazu A s pomocí transformační matice

Proběhnou-li všechny zmiňované body úspěšně, lze očekávat uspokojivý výsledek v míře kvality implementace jednotlivých částí. V případě neúspěchu libovolného dílčího bodu procesu stabilizace dojde k degradaci výsledku, případně je proces ukončen (konkrétní snímek přeskočen), nelze-li z důvodu neúspěchu pokračovat k dalšímu bodu.

3.3.1 Detekce význačných bodů

Význačné body slouží pro jednoznačnou identifikaci konkrétního bodu v obraze. Spolehlivost unikátnosti, invariance (neměnnosti) vůči transformaci obrazu a časová náročnost nalezení takovýchto bodů se liší v jednotlivých algoritmech použitých pro jejich vyhledávání.

Přehledný rozcestník používaných algoritmů pro detekci nejen bodů, ale i význačných oblastí lze najít na wikipedii [8]. Pro naše účely jsou algoritmy vyhledávající hrany (edges) a rýhy (ridges) nevyhovující, uvědomíme-li si, že vzhledem k obecnosti fotografované scény nelze jejich výskyt v obraze zaručit. Zaměříme se tedy na použití detekce jednotlivých bodů (i v tomto případě se však pracuje s malou oblastí několika okolních bodů, které jsou použity pro popis a identifikaci). Z porovnávacího testu (především přesnosti) [9] je patrné, že za nejefektivnější algoritmy lze považovat metodu *SIFT* (*Scale-invariant feature transform*) a metodu *GLOH* (*Gradient Location and Orientation Histogram*). V roce 2006 byl však publikován detektor bodů *SURF* (*Speeded Up Robust Features*) ([10] – revidované vydání z roku 2008) založený na algoritmu SIFT, který vyžaduje podstatně méně výpočetního výkonu ke zpracování za cenu minimální ztráty přesnosti. [11]

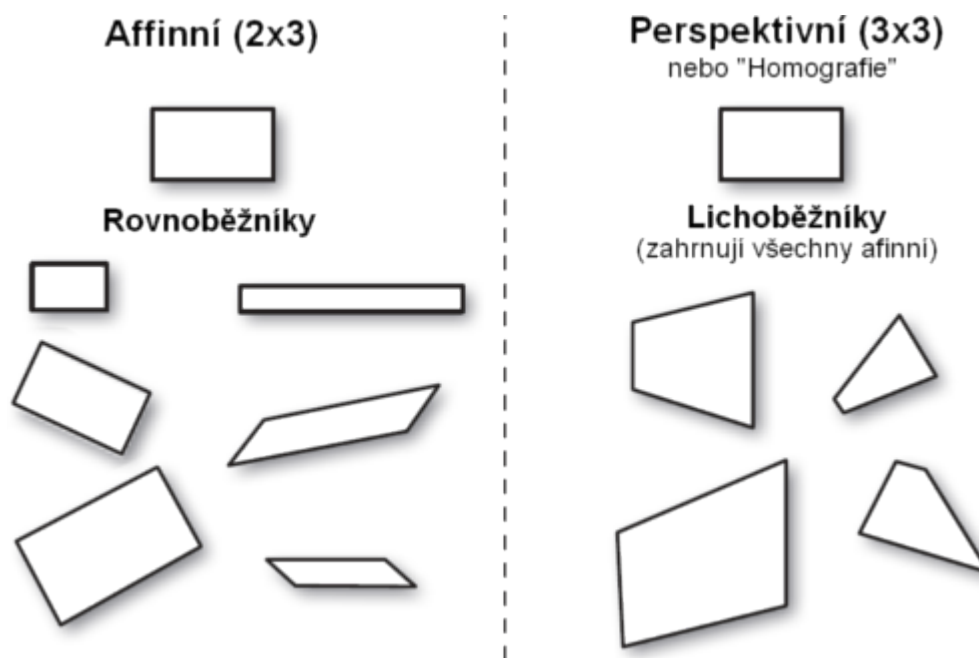
Algoritmus SURF se od svého vzniku dočkal značné popularity a mnoha implementací (seznam lze nalézt na wikipedii – [12]). Je implementován i v knihovně OpenCV potažmo EmguCV použité v této práci.



Obrázek 7: Demonstrace nalezených význačných bodů v praxi

3.3.2 Transformace obrazu

Máme-li ve dvou snímcích spárované význačné body, můžeme přistoupit k transformaci manipulovaného snímku za účelem „překrytí“ původního – řídicího snímku.



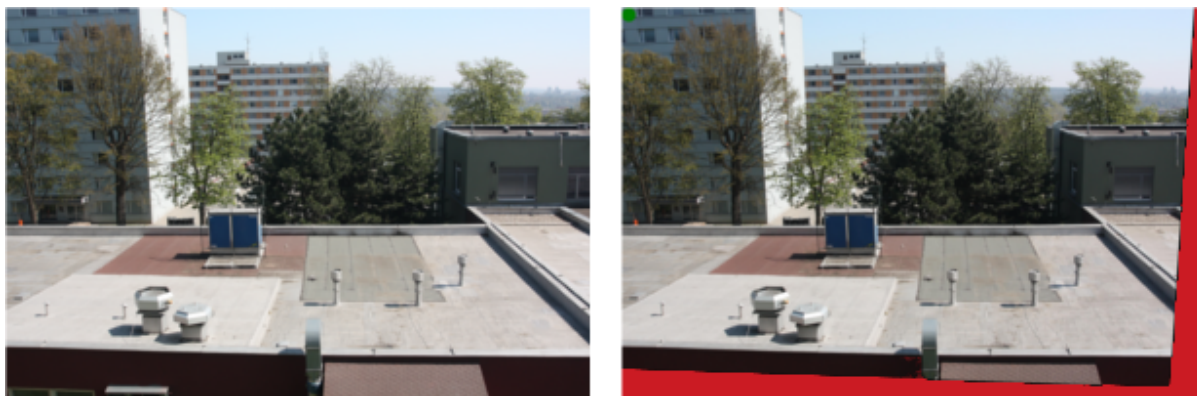
Obrázek 8: Afinní a perspektivní transformace

Cílem takové transformace může být rovnoběžník, který pokrývá změnu velikosti (v našem případě přiblížení – zoom), otočení, posunutí, zkosení, či libovolné kombinace těchto operací. Takováto transformace se nazývá afinní a je reprezentována maticí 2×3 čísel.

Druhou možností je perspektivní transformace (tzv. homografie), která je reprezentována maticí 3×3 čísel a jejím výsledkem je lichoběžník. [5]

Každá z transformací má svoje výhody a nevýhody. Afinní transformace sice neumožňuje transformaci, která je zapotřebí při změně úhlu pohledu na scénu (ke které při fotografování téže scény běžně dochází), ale ze stejného důvodu neumožňuje extrémní chybovost transformace do nežádoucího tvaru obrazu při nedostatečné přesnosti matice. Oproti tomu s pomocí perspektivní matice lze provést přesnou shodu snímků za všech okolností (především tedy při změně úhlu pohledu), avšak s tím rizikem, že při transformaci z nedostatečně přesné matice může dojít ke zkreslení takového rozsahu, kdy se výsledný snímek díky své nedostatečné kvalitě stává nepoužitelným.

Po transformaci obvykle zůstává kolem transformovaného (stabilizovaného snímku) okraj, ke kterému nemá jednotlivý snímek dostatečné množství informací k vyplnění. Řešení je několik. Nejtriviálnějším způsobem je snímek ponechat ve stejné velikosti a oblast bez obrazové informace vyplnit jednou barevnou hodnotou (barvou splývající s pozadím, případně černou a nebo naopak kontrastní barvou pro demonstrační účely) – viz obr. č. 9. Další metodou je oříznutí fotografie do nejmenšího (případně explicitně definovaného) obdélníku – tzv. *trimming*. Metoda zabývající



Obrázek 9: Typický příklad ztráty informace při stabilizaci obrazu

se pokusem o doplnění chybějící informace je tzv. *mozaikování* – doplnění informace z okolních snímků [13]. Opticky nepříjemnou vlastností této metody mohou být opakující se vkládané „dlaždicové“ části. Pokročilejší metody se snaží nedostatky mozaikování eliminovat [14].

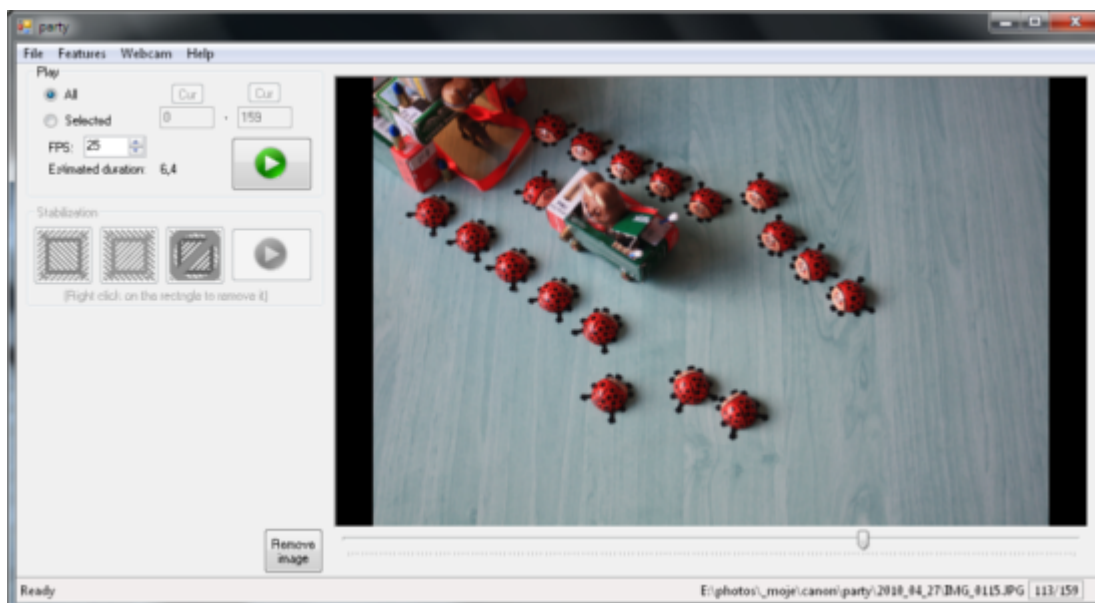
Vzhledem k povaze použití výstupních dat aplikace při stabilizaci (následná editace v profesionálních video editorech) se touto problematikou projekt nezabývá.

4 Implementace

Implementace jako celek i jednotlivých dílčích částí se snažila držet předlohy návrhu (viz předchozí kapitola). Došlo-li oproti navrhovanému postupu k odlišnostem, jejich změny a důvody, proč k nim došlo, jsou následně popsány a vysvětleny.

4.1 Grafické rozhraní

Výsledná podoba grafického uživatelského rozhraní (rozložení komponent v prostoru okna) odpovídá nákresu v kapitole 3.1. Část okna po levé straně je zaplněna ovládacími prvky o statické velikosti, zbytek prostoru (bez panelu hlavní nabídky v horní a stavové lišty v dolní části) je k dispozici náhledu fotografie, který se dynamicky mění v závislosti na velikosti okna aplikace. Velikost náhledu fotografie se přizpůsobuje velikosti panelu náhledu při zachování poměru stran a v případě nesouhlasné velikosti je zbylý prostor vyplněn nerušivou černou barvou (viz obr. č. 10).



Obrázek 10: Demonstrace vzhledu uživatelského rozhraní

Dvě hlavní oblasti ovládacích prvků zahrnují přehrávání všech (nebo vybraného intervalu) fotografií sekvence specifikovanou rychlostí a nástroje pro definici zájmové oblasti pro detekci význačných bodů (více v kapitole 4.3 o stabilizaci).

Na stavové liště nalezne uživatel všechny důležité informace o stavu aplikace. Konkrétně se jedná po pravé straně o informace k zobrazené fotografii – pozice v sekvenci a absolutní cesta

k umístění na paměťovém médiu a po levé straně o informaci týkající se aktuální činnosti programu. Ta je užitečná především při časově náročných operacích (načítání fotografií, stabilizace obrazu...), kdy stavová lišta navíc zobrazuje dosavadní progres operace (poměr hotové a zbývajících částí).

Načítání fotografií do aplikace je časově náročný proces z několika důvodů:

- Fotografie pořízené dnešními fotoaparáty běžně přesahují velikost 10Mpx
- Množství použitých fotografií při tvorbě časosběrného videa běžně dosahuje několika set až tisíc snímků
- Dochází k převodu každého snímku do datového typu knihovny EmguCV
- Navíc je snímek zmenšen na velikost náhledu

Díky možnosti (jež se zároveň stává předpokladem) uložení načtené sekvence do projektového souboru se považuje načítání fotografií za jednorázovou operaci, proto je k změně velikosti na náhled použita nejpomalejší avšak nejkvalitnější metoda pro změnu velikosti – bikubická interpolace. Samotný projekt se fyzicky skládá ze dvou souborů:

1. XML soubor se základními atributy fotografií (velikost originální fotografie a náhledu; počet snímků v sekvenci a jejich odpovídající umístění na paměťovém médiu)
2. Binární soubor serializovaných snímků náhledu v bitmapovém formátu

Je-li binární soubor pro uživatele příliš velký (v závislosti na velikosti snímku náhledu – pro ilustraci soubor o velikosti 675 snímků při rozlišení 300x200px dosahuje objemu dat kolem 80MB), lze binární soubor bez obav smazat a program při následujícím otevření projektu snímky náhledu vytvoří znovu.

Export do video kontejneru `.avi` probíhá pomocí implicitních funkcí knihovny OpenCV a lze pro něj použít veškeré v operačním systému nainstalované kodeky.

4.2 Interpolace jasu

Výsledek interpolace může uživatel při prohlížení snímků v aplikaci pozorovat v reálném čase. Panel uživatelského rozhraní pro zobrazení náhledu se v takovém případě přizpůsobí rozdělením vertikálně (při práci s vertikálními fotografiemi) či horizontálně (v případě horizontálně orientovaných fotografií) na dvě stejné části, kde v horní resp. levé části je zobrazen původní snímek a v dolní resp. pravé části může uživatel pozorovat snímek modifikovaný zvoleným algoritmem pro interpolaci jasu.

Tento přístup umožňuje uživateli porovnávat výsledek a původní snímek bez nutnosti exportu a zvolit nejvhodnější metodu.

Z toho důvodu vyžaduje algoritmus dvou průchodový přístup. V prvním průchodu sadou fotografií se pro každou matici snímku (každý barevný kanál – obvykle tři v případě barevné fotografie) spočítá hodnota odpovídající aritmetickému průměru hodnot jednotlivých pixelů, ze které se počítá výsledná intenzita (jas) fotografie.

Metod pro výpočet hodnoty jasu z obrazu je několik. Pro tento účel budou otestovány dvě a počítá bude metoda, jenž bude dosahovat uspokojivějších výsledků. Pro výpočet jasu (barevné intenzity) pixelu o indexu p lze použít:

1. Prostý aritmetický průměr jednotlivých hodnot pixelů barevných kanálů:

$$I_p = \frac{1}{3}(R_p + G_p + B_p) \quad (1)$$

2. Výpočet intenzity podle empirického vzorce (2) váženého aritmetického průměru založeného na citlivosti lidského oka na jednotlivé barevné složky [15].

$$I_p = 0.299 * R_p + 0.587 * G_p + 0.114 * B_p \quad (2)$$

V programu byla aplikována metoda druhá. Hodnocení úspěšnosti těchto metod je však dosti subjektivní, proto bylo rozhodnutí převzato ze zdroje.

Intenzita I celého snímku j je tedy průměrem všech (m) jednotlivých pixelů ve snímku i :

$$I_j = \frac{1}{n} \sum_{i=(j-\frac{n}{2})}^{j+\frac{n}{2}-1} I_i \quad (3)$$

Pro každou fotografii j v sekvenci je spočítán aritmetický průměr intenzit snímků jeho klouzavého okna velikosti n (včetně něj samotného):

$$I_i = \frac{1}{m} \sum_{p=1}^m I_p \quad (4)$$

Výsledná průměrná hodnota intenzity (jedna hodnota typu *double* pro každý snímek) je uložena do paměti – připravena pro druhý průchod. Samotná operace úpravy jasu probíhá až při prohlížení jednotlivých snímků uživatelem – díky tomu, že většina výpočtů se provedla na počátku a výsledek je k dispozici v paměti, lze upravené snímky vytvářet velkou rychlostí dynamicky bez nutnosti celé snímky ukládat. Způsob úpravy hodnoty jasu (jednotlivé barevné složky) každého snímku lze provést dvěma způsoby:

1. Aditivní

V druhém průchodu dochází k přičtení rozdílu uloženého průměru klouzavého okna a hodnoty aktuálního snímku ke každému pixelu xy kanálu k :

$$I_{xyk} = I_{xyk} + (I_j - I_i) \quad (5)$$

2. Multiplikativní

V druhém průchodu dochází k vynásobení každého pixelu xy kanálu k podílem hodnoty uloženého průměru klouzavého okna a aktuálního snímku:

$$I_{xyk} = I_{xyk} * \left(\frac{I_j}{I_i}\right) \quad (6)$$

Z hlediska zachování co možná nejvyššího stupně identičnosti lze považovat multiplikativní metodu za korektnější – v případě výskytu hodnoty dokonale černé barvy ve fotografii (hodnota 0) zůstává tato hodnota i nadále dokonale černá na rozdíl od aditivní metody, kdy se hodnota posouvá do stupňů šedi. Názorně lze důsledek pozorovat na histogramu, kdy na rozdíl od aditivní metody dochází k posunu histogramu, při multiplikativní metodě dochází k jeho roztažení.

Interpolaci jasu v rámci snímků klouzajícího okna lze metodou alpha blend (prostá verze – všechny hodnoty mají stejnou váhu) dosáhnout vypočtením jednotlivých průměrů všech odpovídajících pixelů snímků okna a tím vytvořit snímek nový, který lze nazvat „průměrným snímkem okna“:

$$I_{xyk} = \frac{1}{n} \sum_{i=(x-\frac{n}{2})}^{x+\frac{n}{2}-1} I_{xyk} \quad (7)$$

4.3 Stabilizace

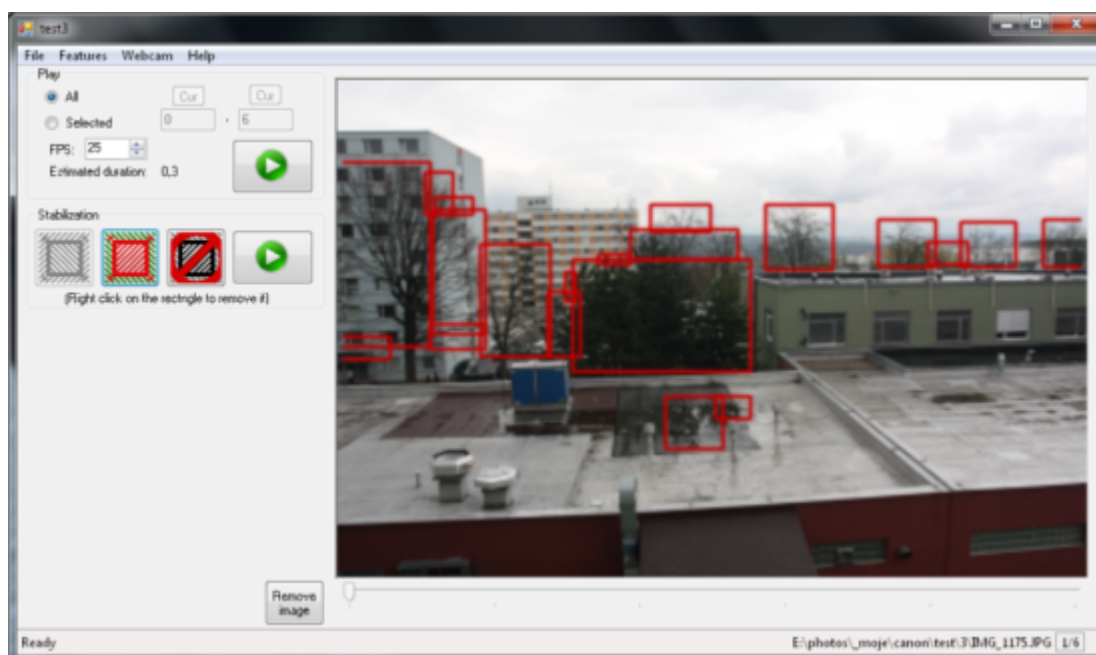
Jak bylo řečeno v kapitole 3.3, stabilizaci lze rozdělit do několika kroků, kterým je zapotřebí se věnovat individuálně. Vzhledem k implementaci algoritmu SURF včetně:

- Vyhledání význačných bodů
- Přiřazení odpovídajících bodů dvou snímků
- Odstranění vadných shod bodů

zůstává, vedle úpravy zdrojového kódu samotné knihovny, velmi málo prostoru pro úpravu procesu stabilizace. Proto bude pozornost zaměřena především na

- Konkretizaci oblasti pro vyhledávání význačných bodů
- Dodatečné odstraňování vadných shod

Při fotografování se často stává, že se na fotografii vyskytuje objekt, který mění svoji relativní polohu (resp. polohu jeho význačných bodů) v obraze. Typickým příkladem jsou například lidé nebo stromy a jejich listy pohybující se ve větru. V takovém případě dojde k nesprávné shodě dvou bodů v obraze (přesto, že význačné body jsou shodné) a díky tomu k nesprávnému výpočtu transformační matice, což vede ke zkreslení výsledného obrazu. Ke stejnému jevu dochází v případě změny



Obrázek 11: Demonstrace vyloučení oblasti vyhledávání v aplikaci

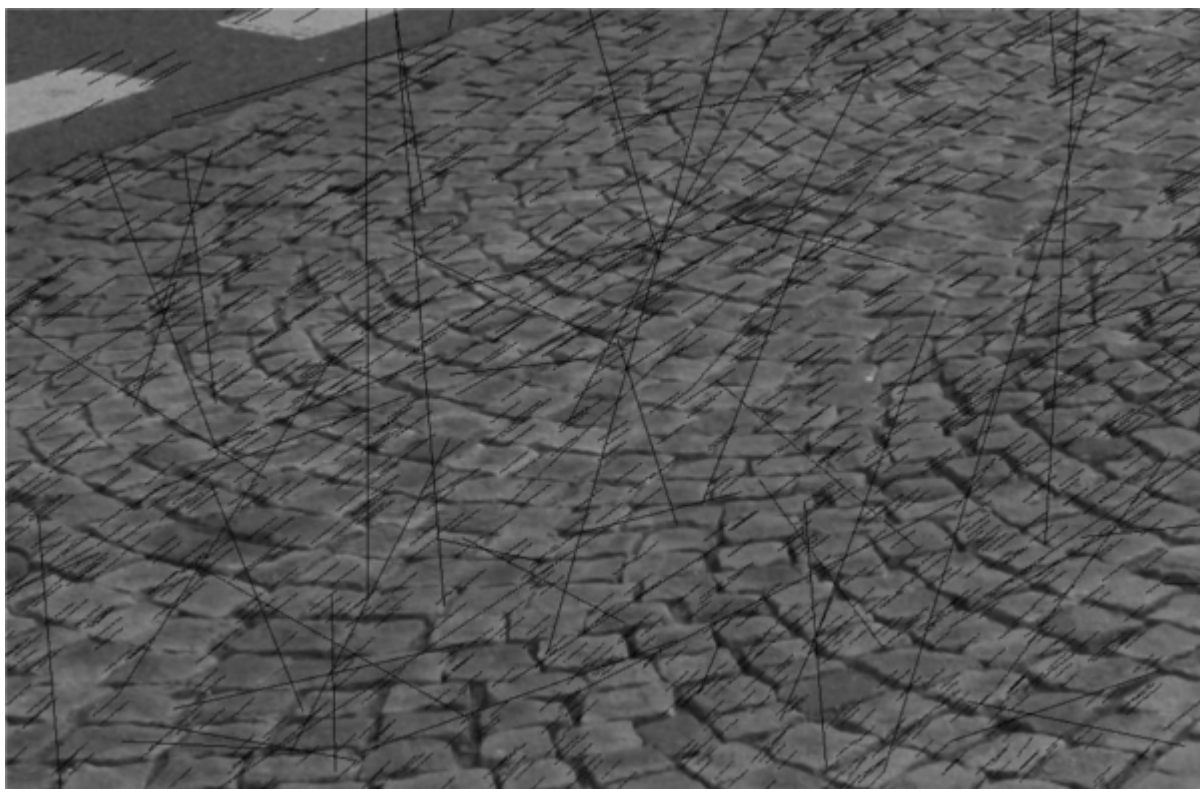
perspektivy v obraze (nejčastěji pohybem fotografa během fotografování), ke které dochází především vyskytují-li se předměty v blízkosti objektivu.

Předejít takové situaci lze pouze omezením části prohledávané algoritmem SURF na takové oblasti, ve kterých k pohybu nedochází. Za tímto účelem disponuje aplikace funkcí, která umožňuje uživateli vybrat části snímku, které z prohledávání vyloučit (*exclude*) nebo naopak vybrat pouze oblast, ve které vyhledání proběhne (*include*). Na obrázku č. 11 můžeme vidět praktickou ukázkou nástroje pro zmenšování oblasti prohledávané algoritmem. Shodným způsobem funguje i inverzní způsob pro výběr prohledávaných míst. Zvolené oblasti lze odstranit buďto všechny příslušným tlačítkem, nebo jednotlivě pravým kliknutím myši do prostoru konkrétního obdélníku.

Vzhledem k identifikaci prohledávané oblasti na principu masky, lze v případě potřeby přidat i jiné obrazce, případně nástroj štětce apod.

4.4 Filtrace vadných shod

Knihovna EmguCV ve své implementaci algoritmu SURF poskytuje metodu pro odstranění vadných shod na základě nesouhlasné velikosti a orientace. Tato metoda je však pro stabilizaci obrazu nedostačující – zanechá i po zpracování mnoho shod mezi původním a stabilizovaným snímkem

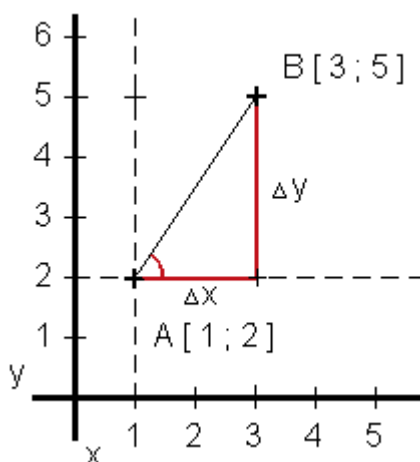


Obrázek 12: Na obrázku lze pozorovat velké množství nesprávných shod projevujících se neshodnou orientací a velikostí vůči shodám správným

neshodně spárované. Bylo proto zapotřebí vyvinout algoritmus, který odstraní takové množství vadných shod, aby transformační matice získaná ze zbylých párů klíčových bodů vedla k transformaci obrazu v uspokojivé přesnosti.

Pokud v jedné souřadné soustavě zaznamenáme odpovídající body snímku předlohy a stabilizovaného snímku, tak jejich spojnicí získáme vektor posunutí daného bodu mezi jednotlivými snímky. Vadná shoda se při vykreslení projeví jako přímka, která svojí velikostí a/nebo orientací neodpovídá okolním přímkám (viz obr. č. 12).

Základní myšlenka algoritmu se proto zaměří na odstranění shod, jejichž směr spojnice se výrazně liší od majoritní většiny ostatních spojnic. Pro takové porovnání je jeden bod z páru shody posunut ve směru jedné z os (řekněme, že se jedná o osu y) o takovou hodnotu, aby směrnice spojnic správných shod všech afinních transformací (například při rotaci či změně velikosti v případě jednotlivých snímků časosběrné sekvence, v poměru velikosti fotografie k rozdílu směru spojnic o relevantní délce, k výrazným výchyilkám nedochází) byly s určitou mírou tolerance rovnoběžné a zároveň se vadné shody projeví výraznou odchylkou směrnice (viz obr. č. 14 vlevo).



*Obrázek 13: Připomenutí
goniometrické funkce tangens*

Spočítáme směrnici pro každou spojnicí bodů všech párů shod – vyjádříme arcus tangens:

$$s = \arctan\left(\frac{\Delta y}{\Delta x}\right) \quad (8)$$

Všechny vypočtené hodnoty směrníc seřadíme podle velikosti a vybereme hodnotu reprezentující směrnici majoritní většiny spojnic klíčových bodů. Empiricky bylo zjištěno, že velice

přesně o této hodnotě vypovídá medián seznamu (správné shody se od směrnice mediánu obvykle liší o velmi malou hodnotu – pouze o několik málo tisícín).

Nyní již stačí vyřadit ty shody, jejichž směrnice spojnice jejich bodů se od mediánu liší o větší hodnotu, než je uživatelem předem definovaná tolerance (implicitně 0,01). Změna tolerance obvykle výrazně ovlivňuje výsledek transformační matice a tudíž i vzhled výstupního snímku.

Jediný případ vadné shody, který metoda nedetekuje, je situace, kdy směrnice takového páru shody spadá do tolerance i přes to, že velikost spojnice se značně liší od většiny. Pro eliminaci takových bodů je nutné metodu provést znovu s tím rozdílem, že tentokrát provedeme posun v ose kolmé na osu předchozí (tedy v ose x).

Úspěch implementovaného algoritmu lze pozorovat z následující tabulky.

	Shod celkem nalezeno	Po implicitní	Po explicitní
Odstraněno shod	-	6037	3025
Počet shod po aplikaci	13740	7703	4678
Z toho vadných shod	9114	3077	52
Správné shody v %	33,70%	60,10%	98,90%

Tabulka 1: Přehled úspěšnosti algoritmu knihovny EmguCV (implicitní) a následné aplikace implementovaného algoritmu na základě směrnice spojnic (explicitní).

Z obrázku č. 14 je patrné, že algoritmus odstranil téměř všechny relevantní nesouhlasné shody. Výsledkem je transformační matice, jejíž projekcí získáme přesnější výsledek (bílý rovnoběžník vyznačuje oblast původní fotografie) – změna je patrná v dolní části fotografie v místě kabelu.

V případě neuspokojivého výsledku výstupních snímků lze analýzou snímků shod (o jejichž vytvoření lze rozhodnout před procesem stabilizace) odhalit příčinu. Zpravidla se jedná o

1. Nedostatečné množství shod (celkem – tedy i správných)

V takovém případě lze zvýšit toleranci rozdílu směrnice (především jedná-li se o fotografii menších rozměrů, kde posun v jednotlivých osách nedosahuje velkých hodnot, může se stát, že některé správné shody mohou být vyřazeny na základě nízké tolerance)

2. Výskyt výrazného množství nesprávných shod

Je zapotřebí najít uspokojivou hranici tolerance, při které jsou vadné shody odstraněny a správné zachovány.



Obrázek 14: Vlevo vidíme výchozí spojnice bodů po aplikaci pouze implicitního algoritmu knihovny EmguCV, z obrázku je patrné velké množství vadných shod; vpravo vidíme výsledek po aplikaci navrženého algoritmu na základě směrnice



Obrázek 15: Na detailu můžeme vidět mnoho vadných shod, které nebyly algoritmem odstraněny. Za dostatečného množství správných shod však výsledek téměř neovlivní.

Empiricky bylo zjištěno, že u fotografií dnešních velikostí (10Mpx a více) vzniká přesnější matice ze snímků, kde nižší tolerance způsobila odstranění správných shod (pochopitelně spolu s nesprávnými) než u snímků, kde vyšší tolerance vedla k zachování shod nesprávných.

Problémem pro algoritmus zůstávají spojnice nesouhlasného směru, které mají v poměru k velikosti snímku zanedbatelnou velikost (obr. č. 15). Díky svojí malé velikosti však tyto nesprávné shody výslednou transformační matici téměř neovlivní.

5 Zhodnocení výsledků

Ačkoliv si lze výsledky prohlédnout na přiloženém DVD, v této kapitole se čtenář dozví o částech projektu, kterým se jednotlivé kapitoly věnují jednotlivě, jejich dosažených výsledcích, úspěších a nedostacích. O způsobech, kterými by se na aplikaci dalo pokračovat v budoucnu, ať už ve volném čase či v rámci diplomové práce. O jaké prvky lze aplikaci rozšířit a jaké funkce přidat v budoucnu a jakým směrem by se vývoj ubíral.

5.1 Funkční prvky rozhraní

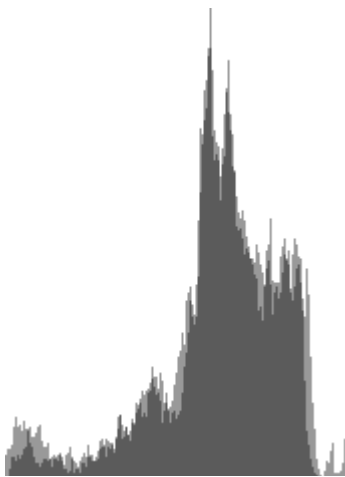
Z hlediska účelnosti a jednoduchosti splnila aplikace požadavky návrhu (viz obr. č. 3). Program neobsahuje zanořené ovládací prvky a umístění panelu nástrojů umožňuje rychlé a intuitivní ovládání. Důležité informace o činnosti aplikace a aktuálnímu snímku náhledu lze nalézt na stavové liště. Jednorázové operace a aplikování algoritmů probíhá přes hlavní nabídku s případným dialogovým oknem, zatímco z uživatelského hlediska komplexnější a opakované operace jsou prováděny přímo na panelu nástrojů. Velikost náhledu fotografie se přizpůsobuje svojí orientaci a aktuální velikosti okna aplikace a díky bikubické interpolaci použité k změně velikosti zůstává i při nižším rozlišení a větší velikosti okna náhled účelný a použitelný. S pomocí exportu videa lze sekvenci snímků uložit v libovolné velikosti a kodeku, avšak pouze do kontejneru `.avi`. Aplikace umožňuje export přímo z datové struktury náhledu fotografie (která se nachází v operační paměti), v tom případě proběhne export videa ve zlomku času potřebného k exportu ze souborů. Během testování nedocházelo k pádům nebo neošetřeným výjimkám ani nebyly zaznamenány žádné závažnější problémy.

Mezi nedostatky grafického uživatelského rozhraní patří bezpochyby nedostatečný prostor pro betatesting, což se v určitých situacích může projevit v drobných chybách aplikace jako jsou neošetřené stavy komponent apod. Testování daty většího objemu (800snímků) s náhledem o velikosti 300x200px probíhalo na počítačích s kapacitou operační paměti alespoň 3GB, není proto známo, jakým způsobem se aplikace chová v případě použití velkého objemu dat při dostupnosti nižší kapacity. V takovém případě má však uživatel možnost dále snižovat velikost náhledu načítaných fotografií a tím omezit jediný dynamický zdroj spotřeby paměti.

Pro nasazení programu je zapotřebí provést některé kosmetické a funkční úpravy. Jedná se především o tvorbu nápovědy formou externího souboru, případně tooltipů jednotlivých tlačítek, potvrzovací dialogy operací, ikona a název aplikace, nástroj pro zmenšení velikosti a oříznutí exportovaných fotografií (potřebné především po stabilizaci fotografií) a v neposlední řadě je zapotřebí provést důkladný betatesting.

5.2 Použitelnost interpolace jasu

Použitelnost algoritmu pro interpolaci jasu na základě posouvání či roztahování histogramu je nejslabší částí programu. Vzhledem k empiricky zjištěným skutečnostem se přístup implementovaného algoritmu jeví jako naivní. Přístup k problému na základě globální změny na úrovni konstantní (lineární - v případě multiplikativní metody na základě hodnoty barvy) změny hodnot jednotlivých pixelů zohledňuje pouze výslednou hodnotu intenzity světla odrážejícího se od povrchu objektů ve scéně. S čím však nepočítá jsou chemické, fyzikální a geometrické vlastnosti povrchů jednotlivých objektů, které výrazným způsobem mění poměr odraženého/pohlceného světla, čímž dochází nejen ke změně barevných vlastností, ale i ke změně intenzity na základě sekundárních světelných zdrojů (viz obr. č. 16).



Obrázek 16: V praxi se změna osvětlení scény projeví na histogramu komplexnějším způsobem

Algoritmus proto dokáže do určité míry odstranit změnu intenzity způsobenou záznamem fotoaparátu. Nedokáže však za běžných okolností odstranit změnu způsobenou změnou světelných podmínek scény.

Určitou alternativu poskytuje metoda interpolace založená na principu alpha blend, kdy jsou pro zvolené snímky spočítány průměrné hodnoty odpovídajících pixelů a výsledný snímek tvoří „prolnutí“ na základě průhlednosti. Tuto metodu lze použít pouze v sekvencích s nepatrnými změnami rozložení scény v rámci sousedících snímků (např. tající kostka ledu, plynulý rozkvět rostliny apod.). Za těchto podmínek produkuje metoda překvapivě dobré výsledky. V sekvencích zachycujících např. dopravní tok, stavbu budovy apod. vytváří alpha blend nepříjemné rozmazání či „duchy“ objektů výrazně měnících polohu mezi následujícími snímky.

5.3 Spolehlivost stabilizace

Stabilizace obrazu není zcela automatická, což vyžaduje od uživatele určitou míru interakce, ale ve výsledku manuální korekce a volba vhodných parametrů vede k mnohem kvalitnějším výstupním sekvencím než by tomu bylo při použití čistě automatizovaného procesu. Lze proto za specifických (při tvorbě časosběrného videa snadno dosažitelných) podmínek očekávat sekvence, které výrazným způsobem (ne-li ne zcela) stabilizují obraz a zvýší tak celkový dojem a kvalitu výstupního videa.

Pomocí nástroje pro interaktivní definici prohledávané oblasti může uživatel odstranit ze základního snímku (snímek s polohou scény, které se stabilizované snímky snaží co možná nejvíce přiblížit) oblasti, ve kterých by došlo k nalezení význačných bodů, které nemají vůči zbytku scény neměnnou polohu (pohybující se listy stromu apod.). Navíc má možnost ovlivnit počet nalezených bodů změnou parametrů funkce knihovny OpenCV – změnou prahu přípustnosti a kvalitu jejich popisu použitím rozšířeného deskriptoru bodu.

Počet shod a poměr správných vůči nesouhlasným shodám může uživatel změnit hodnotou tolerance rozdílu směrnice implementovaného algoritmu.

Pokud uživatel při stabilizaci zvolil možnost tvorby snímků se záznamy o shodách, má v případě neuspokojivého výsledku k dispozici materiál, z kterého lze vyvodit informaci o tom, která část byla pro stabilizaci nevhodně nakonfigurována.

Pro spolehlivost procesu stabilizace je nutno dodržet několik (z principu algoritmu vcelku pochopitelných) podmínek při fotografování:

1. Ve scéně se musí vyskytovat pevná místa, z kterých lze extrahovat význačné body. Čím větší je takový poměr takové plochy v obraze, tím více význačných bodů algoritmus najde a tím kvalitnější bude zpravidla výstup. Stabilizace proto nikdy nebude fungovat např. při fotografování pouze oblohy nebo tekoucí vody.
2. Ve scéně nesmí dojít ke změně perspektivy. V závislosti na vzdálenosti nejbližších předmětů od objektivu je nutné, aby se fotograf pokusil fotografovat vždy ze stejného místa. Mnohdy stačí pouze malý krok stranou a díky změně perspektivy scény může dojít k malému počtu shod a nebo posun vzájemných význačných bodů ve snímku zkreslí celou výslednou transformaci (viz obr. č. 17). Je proto nutné aby si fotograf například vyznačil přesné místo, z kterého bude pořizovat fotografie a tuto pozici udržoval co nejpřesněji po celou dobu sběru materiálu pro časosběrné video.
3. Kvalita fotografií musí být dostačující (například extrémní rozostření snímku má vliv na počet a kvalitu nalezených bodů).



Obrázek 17: Demonstrace výsledku změny (cca jeden krok stranou) perspektivy při výskytu blízkých předmětů – v tomto případě kostek silnice a obrubníku. Na detailu je patrné "posunutí" obrubníku vůči silnici, které způsobilo zkreslení celé transformace.

Při dodržování těchto zásad při tvorbě časosběrné sekvence je šance na vysokou kvalitu výstupního videa mnohem vyšší.

Kvalita výsledku by se dala dále zvýšit odstraněním zbylých vadných shod, či například použitím Kalmanova filtru.

6 Závěr

Výsledkem této práce je program, který nejen umožňuje načítat a manipulovat s fotografiemi, pracovat s nimi v rámci projektu a dovoluje jejich přehrávání a vytváření video sekvencí z poskytnutých fotografií, ale slouží především jako platforma pro algoritmy zabývající se zpracováním a úpravou těchto fotografií za účelem eliminace nežádoucích vlastností z pohledu celé sekvence.

Interpolaci změny jasu na základě změny osvětlení scény, zpracovávanou jako první algoritmus, nelze vzhledem k vlastnostem povrchu objektů ve scéně řešit prostou transformací histogramu. Proto lze tuto metodu uplatnit pouze v některých případech a nebo aplikovat metodu založenou na principu alpha blend, která dovoluje vyrovnat hodnotu jasu v sekvencích, kde nedochází k rychlým změnám polohy objektů.

Oproti tomu stabilizace obrazu poskytuje při dodržování několika málo zásad (vycházejících z principu algoritmu pro hledání a porovnávání význačných bodů) při fotografování velice uspokojivé výsledky. V případech, kdy se ve scéně vyskytují pohybující objekty, které mění svou relativní polohu vůči statickým objektům scény (rameno jeřábu na staveništi, větve stromu ve větru apod.) a způsobují tím degradaci přesnosti transformační matice, má uživatel možnost pomoci uživatelského rozhraní a nástrojů k tomu určených takové oblasti označit a zabránit použití bodů z těchto nežádoucích míst. Pro zvýšení přesnosti transformace snímků byl implementován algoritmus na vyřazování vadných shod, na jejichž poměru vůči správným shodám závisí výsledná kvalita stabilizace. Běžně množství zastoupení správných shod přesahuje hodnotu 95%, kde zbylé množství je zastoupeno vadnými shodami o malé velikosti, jejichž vliv na výslednou transformaci je obvykle zanedbatelný. Spolu s vyznačením použitých oblastí dosahuje stabilizace velice kvalitních výstupů.

Už zpočátku byl program navržen pro integraci a snadné připojení dalších algoritmů. Oblast časosběrného videa poskytuje mnoho prostoru pro zpracování obrazu, potažmo počítačové vidění (například stabilizace obrazu na základě detekce obličeje v obraze při dlouhodobém fotografování za účelem zachycení růstu vousů, stárnutí apod.). Implementace takových algoritmů spolu se zdokonalováním funkčnosti uživatelského rozhraní nabízí mnoho možností při volbě směru dalšího vývoje.

Literatura

- [1] FARID, Hany; POPESCU, Alin C. Blind Removal of Lens Distortion. *Optics InfoBase : Journal of the Optical Society of America A* [online]. 2001, vol. 18, issue 9, [cit. 2010-05-13]. Dostupný z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.9227&rep=rep1&type=pdf>>.
- [2] PLHÁKOVÁ, Alena. *Učebnice obecné psychologie*. Vyd. 1. . Praha : ACADEMIA, 2007. 472 s. ISBN 978-80-200-1499-3.
- [3] ZMEŠKAL, Oldřich; BUCHNÍČEK, Miroslav; SEDLÁK, Ondřej. Principy digitálního záznamu obrazu. [online]. 2001, [cit. 2010-05-15]. Dostupný z WWW: <http://www.fch.vutbr.cz/lectures/imagesci/download/stud02_hrakra01.pdf>.
- [4] ATKINSON, Rita L., et al. *Psychologie*. 2., aktualiz. vyd. Praha : PORTÁL, 2003. 751 s. ISBN 80-7178-640-3.
- [5] BRADSKI, Gary; KAEHLER, Adrian. *Learning OpenCV*. Sebastopol : O'REILLY, 2008. 555 s. ISBN 978-0-596-51613-0.
- [6] FANG, James; MANN, Steve. Using graphics devices in reverse : GPU-based image processing and computer vision [online]. [cit. 2010-05-07]. Dostupný z WWW: <http://www.uweb.ucsb.edu/~yichuwang/ecv/paper/using_graphics_device_in_reverse.pdf>.
- [7] *Emgu CV : OpenCV in .NET (C#, VB, C++ and more) (en)* [online]. 2008 [cit. 2010-05-13]. Dostupné z WWW: <<http://www.emgu.com>>.
- [8] Feature detection (computer vision) In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 16 December 2004, 10 February 2010 [cit. 2010-05-15]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Feature_detection_\(computer_vision\)](http://en.wikipedia.org/wiki/Feature_detection_(computer_vision))>.
- [9] MIKOLAJCZYK, Krystian; SCHMID, Cordelia. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2005, vol. 27, no. 10, [cit. 2010-05-09]. Dostupný z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.5507&rep=rep1&type=pdf>>.
- [10] BAY, Herbert, et al. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* [online]. 2008, vol. 110, issue 3, [cit. 2010-05-09]. Dostupný z WWW: <<http://linkinghub.elsevier.com/retrieve/pii/S1077314207001555>>. ISSN 1077-3142.
- [11] BAUER, Johannes; SÜNDERHAUF, Niko; PROTZEL, Peter. Comparing several implementations of two recently published feature detectors. In *International Conference on Intelligent and Autonomous Systems*. Toulouse : 2007 [cit. 2010-05-09]. Dostupné z WWW: <<http://www.tu-chemnitz.de/etit/proaut/rsrsrc/iav07-surf.pdf>>.
- [12] SURF In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 20 July 2007, 27 April 2010 [cit. 2010-05-15]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/SURF>>.

- [13] LITVIN, Andrew; KONRAD, Janusz; KARL, William C. Probabilistic video stabilization using Kalman filtering and mosaicking. In *Conference on Electronic Imaging*. Santa Clara : A, 2003 [cit. 2010-05-10]. Dostupné z WWW: <<http://iss.bu.edu/jkonrad/Publications/cpapers/Litv03ivcp.pdf>>.
- [14] MATSUSHITA, Yasuyuki, et al Full-frame Video Stabilization. In *Computer Vision and Pattern Recognition*. San Diego : A, 2005 [cit. 2010-04-02]. Dostupné z WWW: <http://research.microsoft.com/en-us/people/yasumat/fullframe_cvpr05.pdf>.
- [15] KRŠEK, Přemysl. Základy počítačové grafiky [studijní opora]. Brno: FIT VUT v Brně

Seznam příloh

Zdrojové i přeložené binární soubory, vzorky dat i ukázková výstupní videa lze nalézt na přiloženém DVD.