



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

SPOJOVÁNÍ A KOREKCE FOTOGRAFIÍ

MERGING AND CORRECTION OF PHOTOS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN TRČKA

VEDOUcí PRÁCE
SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Trčka Jan**

Obor: Informační technologie

Téma: **Spojování a korekce fotografií**
Merging and Correction of Photos

Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte dostupnou literaturu na téma spojování fotografií se zaměřením jak na hledání korespondencí a projekci, tak i na vzorkování a korekci zkreslení obrazu.
2. Vyberte vhodný postup pro dosažení vysoké kvality fotografií po jejich spojení a diskutujte možné vlastnosti navrženého řešení.
3. Navrhněte způsob implementace a vyberte vhodné prostředí pro naprogramování úlohy.
4. Implementujte postup spojování fotografií a jeho funkčnost demonstруйте na vhodném příkladě.
5. Diskutujte dosažené vlastnosti řešení a možnost pokračování v práci.

Literatura:

- Dle pokynů vedoucího

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, prof. Dr. Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat aplikaci pomocí knihovny OpenCV. Tato práce se zabývá problematikou korektního spojení více fotografií, pokouší se eliminovat vady, které vznikly při jejich pořizování, a které mohou vznikat při samotném procesu spojování fotografií ve výsledné panorama.

Abstract

The aim of this work is to design and implement an application using an OpenCV library. This paper describes issues of correct photo merging, attempts to eliminate defects which were created in process of taking these photos and which may occur in the actual merging process.

Klíčová slova

Panorama, Fotografie, Spojování fotografií, SURF, ORB, OpenCV

Keywords

Panorama, Photo, Mergin of photos, SURF, ORB, OpenCV

Citace

TRČKA, Jan. *Spojování a korekce fotografií*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Zemčík Pavel.

Spojování a korekce fotografií

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Trčka
18. května 2016

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Prof. Dr. Ing. Pavlu Zemčíkovi za jeho čas, odborný dohled, konzultace a vedení při řešení bakalářské práce a tvorbě technické zprávy.

© Jan Trčka, 2016.

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

1 Úvod	3
2 Technika tvorby panoramat	4
2.1 Repräsentace obrazu	4
2.2 Hledání klíčových bodů a jejich párování	7
2.3 Projekce fotografií	10
2.4 Vady fotografií a jejich kompenzace	12
2.5 Sloučení fotografií	13
2.6 Existující řešení	15
3 Návrh řešení	17
3.1 Požadavky aplikace	17
3.2 Návrh automatizace	18
3.3 Návrh skládání fotografií	18
3.4 Návrh postupu řešení	18
4 Implementace a vyhodnocení	20
4.1 Automatizace zpracování	20
4.2 Generování panoramat	22
4.3 Parametry programu	23
4.4 Vyhodnocení	24
5 Závěr	29
Literatura	30
Přílohy	32
A Obsah CD	34
B Manual	35
B.1 Požadavky	35
B.2 Překlad	35
C Popis tříd a souborů	36

Seznam zkratek

SIFT Scale-Invariant Feature Transform

SURF Speeded up robust features

ORB Oriented FAST and Rotated BRIEF

FAST Features from Accelerated Segment Test

BRIEF Binary Robust Independent Elementary Features

Kapitola 1

Úvod

Panoramatické fotografie může dnes pořizovat naprosto každý. S nástupem chytrých telefonů také přibylo situací, kdy si lidé neužívají daný moment v krajině, či s přáteli, ale pořizují vydeozáznam nebo fotografují. Fotoaparáty těchto telefonů sice nedosahují na moderní zrcadlovky, ale svou kvalitou otevírají možnosti obrazům, které, ještě před několika roky, byly pouze pro profesionální fotografy. S následným zvyšujícím se výkonem výpočetních jednotek, vznikla možnost úpravy a korekce pořízených snímků přímo uvnitř telefonů. Novější verze již obsahují algoritmy, které samy rozpoznají foceně panorama a jednotlivé snímky spojí do jednoho.

Máme-li však starší zařízení nebo fotoaparát bez těchto vymožeností, znamená to, že focená panoramata musíme dodatečně vyhledat a spojit pomocí jednoho z mnoha dostupných nástrojů. V těchto aplikacích je většinou nutné tvořit panoramata po jednom, nekvalitnější se navíc dotazují na řadu dodatečných parametrů, a tak se z tohoto úkolu může stát velmi úmorný proces.

V této bakalářské práci se budu zabývat procesem tvorby panoramatických fotografií. Získané znalosti následně využiji při návrhu a implementaci aplikace, jejímž účelem bude tvorba panoramat s důrazem na automatizaci tohoto procesu.

V následující kapitole jsou shrnuty informace, týkající se aktuální situace v oblasti spojování obrázků a fotografií. Jsou zde také představeny vady, které se u fotografií objevují, a analýza dostupných nástrojů. Kapitola 3 se zabývá zhodnocením současného stavu a využitím takto získaných znalostí v návrhu řešení, jak by měl výsledný program fungovat. Implementačními detaily se zabývá kapitola 4, která popisuje rozdělení celé aplikace do jednotlivých tříd, využití knihovny a vyhodnocení výstupních panoramat. V závěrečné kapitole 5 je popsáno shrnutí cílů práce, jejich naplnění a představení možností do budoucna.

Kapitola 2

Technika tvorby panoramat

Skládání panoramat je proces, při kterém se několik fotografií s překrývajícími se zornými poli zpracuje v jeden výsledný obraz. Samotné splynutí snímků je však až jedním z posledních kroků. Nejprve je nutné použité fotografie umístit do prostředí, simulujícího moment, kdy byly pořízeny. Pro vytvoření obrazu je důležitá jejich vzájemná poloha. Tu lze určit pomocí algoritmů pro vyhledání korespondujících si bodů napříč snímky. Tímto docílíme nejen správného umístění do prostoru, ale i korektní překrytí již umístěných fotografií.

V této kapitole jsou popsány základní informace, vztahující se k obrazu, fotografiím a jejich zpracování. V závěru kapitoly jsou také představeny vady, které se nejčastěji u fotografií objevují. Zde uvedené informace neslouží jako úplný přehled z této oblasti, ale jen jako shrnutí klíčových informací, důležitých pro tuto bakalářskou práci.

2.1 Reprezentace obrazu

Před samotným zpracováním jakéhokoliv obrazu, je potřeba získat teoretické znalosti, jakým způsobem je v počítači reprezentován. Jedině tak, s ním následně budeme moci korektně pracovat. Nejprve je nutné, převést obraz okolního reálného světa, který vidíme, do jeho digitální podoby. Tomuto procesu se říká digitalizace. Ten spočívá ve vzorkování, kvantování a kódování obrazového signálu.

Obraz okolního reálného světa lze reprezentovat jako spojitě rozložené jasů v rovině. Rozložení hodnot jasu je pro statistický obraz definováno obrazovou funkcí

$$f(x, y). \tag{2.1}$$

Kde f je reálná, spojitá funkce s reálnými argumenty a x, y definují bod v rovině. Její definiční obor D je dán:

$$x \in (x_{min}, x_{max}), y \in (y_{min}, y_{max}). \tag{2.2}$$

Během procesu digitalizace se definiční obor této funkce zpracovává vzorkováním a obor hodnot

$$0 \leq f(x, y) \leq f_{max}. \tag{2.3}$$

kvantováním. Takto definovaná obrazová funkce je schopná reprezentovat pouze obraz v odstínech šedi. Barevný obraz lze definovat jako soubor několika obrazových funkcí pro každou barevnou složku podle zvoleného barevného modelu [14].

Vzorkování je proces diskretizace souřadnic obrazu do matice s rozměry $M \times N$, která nám dá tzv. prostorové rozlišení. Pro výpočet vzorkovacího intervalu se využívá Shannonova teorému:

$$f_v > 2f_{max}[s^{-1}]. \quad (2.4)$$

Ten udává, že pro dvojrozměrné signály je zapotřebí zvolit interval vzorkování tak, aby byl minimálně dvakrát menší než velikost nejmenšího detailu v obraze. Obvykle se používá velikostně 5-krát menšího elementu, než je teoretická mez daná Shannonovým teorémem [9].

Při kvantování pro digitální zpracování se obvykle používá K stejně velkých intervalů. Dochází-li k počítačovému zpracování obrazu, je počet intervalů dán jako:

$$K = 2^b, \quad (2.5)$$

kde b je počet bitů, do kterých je hodnota kvantována. Někdy je vhodnější využívat kvantování s proměnnou délkou intervalu, tímto procesem lze předejít vzniku rušivých artefaktů v obraze, které vznikají při malém počtu kvantovacích hodnot [14].

Je-li obraz pomocí vzorkování převeden na matici $M \times N$, pak je každý bod obrazu reprezentován určitou hodnotou. Tato hodnota je určena počtem bitů, tvořících jeden pixel obrazu, a její maximální možná velikost se označuje pojmem barevná hloubka. Obraz ve stupních šedi nám udávají hodnoty 0-255 pro každý bod, což odpovídá 8bitovému obrazu. Naopak k reprezentaci barevného obrazu se využívá 24bitů, kde každá barevná složka má svých 8bitů. Tuto 24bitovou hodnotu je možno rozšířit o dalších 8bitů, které budou reprezentovat průhlednost obrazu v daném bodě. Dodatečná hodnota je známá také jako alfa kanál.

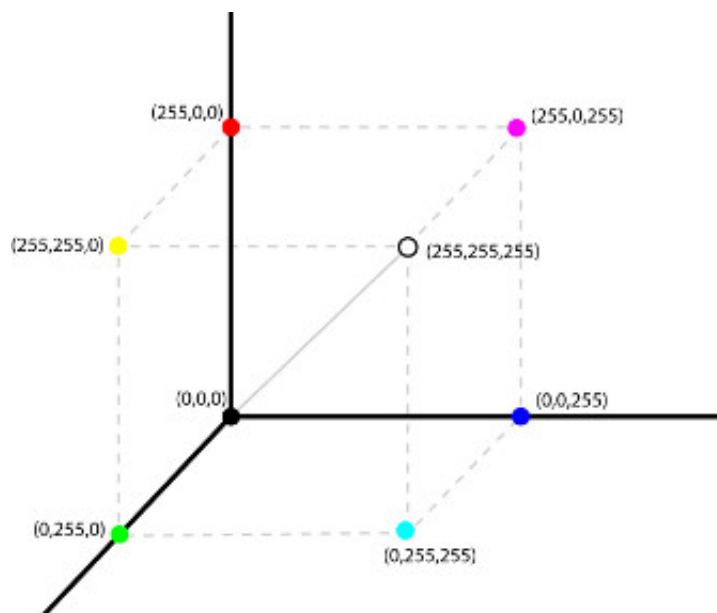
Obrazy jsou tvořeny pomocí barevných modelů, které se liší podle vhodnosti použití. K zpracování obrazu a fotografií jsou důležité zejména modely RGB a YCbCr. Mezi další modely patří CMYK, využívaný v tisku, nebo YUV pro přenos televizního signálu, avšak tyto modely jsou pro tuto práci nepotřebné a nebudou více rozebírány.

Model RGB

Jedním z nejrozšířenějších barevných modelů je RGB model, jehož název je odvozen podle použitých základních barev: Red, Green a Blue. S jejich pomocí je možné vytvořit téměř všechny základní barvy. Jedná se o tzv. aditivní model, který je založen na postupném přidávání světla. Na tomto základu pracují všechny dnešní televize, monitory, apod. Každý jejich pixel je tvořen třemi subpixely, každý pro jednu z barevných složek. Díky velikosti subpixelů a nedokonalosti oka, se výsledný pixel jeví jako jedna barva.

Transformujeme-li barevné spektrum modelu do podoby fyzického tělesa, vznikne nám tzv. jednotková krychle. V této krychli z obrázku 2.1 pak každá osa představuje červené, zelené nebo modré světlo. Jejich počáteční kombinace, s nulovými hodnotami všech tří složek, tvoří černou barvu se souřadnicemi (0, 0, 0). Naopak, maximální hodnoty všech tří složek vytvoří barvu bílou v souřadnicích (255, 255, 255). Zbylé vrcholy jsou tvořeny vždy kombinací dvou maximálních hodnot a jedné minimální. Takto získáme tyto barvy: Cyan (azurová), magenta (purpurová) a yellow (žlutá).

Jelikož model RGB obsahuje pouze informace o základních barvách, vznikla řada jeho variant. Nejrozšířeněji používaná varianta je RGBA, ve které byl základní model RGB rozšířen o alfa kanál A s informací o průhlednosti pixelu. RGBA se uplatnil hlavně u obrázků ve formátu PNG.



Obrázek 2.1: Jednotková krychle - Zobrazení barevného modelu RGB

Model YCbCr

Tento barevný model, využívaný především u videa a u digitálních fotografií, vychází z modelu YUV, používaného při televizních přenosech, který vznikl pro snadnější přechod z černobílého na barevné vysílání. Na rozdíl od modelu RGB, je zde barva zastoupena jako kombinace světlosti Y a chrominančních signálů Cb a Cr . Úkolem Y je odlišení luminance. Lidské oko je velmi citlivé na změnu světelnosti, avšak menší změny v chromalizačních signálech rozlišit nedokáže. Pokud obraz ztratí v chromalizačních kanálech okolo 50 % informací, je změna pro oko nepostřehnutelná a díky této kompresi lze ušetřit až 50 % informací na popis podobného obrazu [8].

K prezentaci obrazu na monitor je nutné jednotlivé složky YCbCr převést na RGB. Hodnota Y patří do intervalu $\langle 0 - 1 \rangle$ a chrominanční hodnoty Cb a Cr do intervalu $\langle -0.5; 0.5 \rangle$. Tyto intervaly jsou dány standardem CCIR-601. Před výpočtem je potřeba posunout hodnoty Cb a Cr na kladné celá čísla v intervalu $\langle 0 - 255 \rangle$ přičtením konstanty. Vzorec pro výpočet YCbCr hodnot z RGB:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.6)$$

Hodnoty Cb a Cr lze zkráceně definovat jako:

$$Cb = 0.5643(B - Y) \quad (2.7)$$

$$Cr = 0.7133(R - Y) \quad (2.8)$$

A naopak, převod z YCbCr na RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.344 & -0.714 \\ 1.0 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \quad (2.9)$$

Pro video se složka Y signál upraví tak, aby se vešla do rozmezí 16 – 235, zatímco Cb a Cr signály jsou upraveny pro rozmezí 16 – 240. U statických obrázků se využívá celé 8-bitové rozmezí [16].

2.2 Hledání klíčových bodů a jejich párování

Jednou z nejdůležitějších částí tvorby panoramat je správná detekce klíčových bodů na jednotlivých fotografiích a jejich spárování mezi na sebe navazujícími obrazy. Nenalezneme-li správně klíčové body, nebo nepodaří-li se nám je korektně přiřadit mezi sebou, jsou všechny další kroky zcela zbytečné. Ke správnému zpracování fotografií je doporučeno překrytí zorných polí alespoň 15 – 30%, tímto se docílí nalezení dostatečného počtu společných bodů k určení jejich vzájemné polohy.

Je nutné si ovšem uvědomit, že, známe-li pouze pozici takto nalezených bodů, stále s těmito fotografiemi nedokážeme nic provést. Z tohoto důvodu potřebujeme i jejich deskriptory. Tyto deskriptory obsahují souhrn charakteristik daného klíčového bodu a jeho okolí, pomocí něhož je možné tento bod zpětně identifikovat s naprostou jistotou. Tato charakteristika je obvykle uložena jako vektor určité délky. Důležitou vlastností deskriptorů je jejich nezávislost na souřadnicích klíčového bodu a měřítku obrazu. Měly by také být odolné vůči transformaci snímku, jako otočení nebo překlopení. Máme-li takto určené body a jejich deskriptory, je načase je porovnat se všemi body a deskriptory druhého snímku. K porovnání lze využít například Eukleidovskou metriku, jejíž výsledek určuje, jak moc jsou si dané dva body podobné.

Speeded up robust features - SURF

Hojně využívaný algoritmus SIFT (*Scale-Invariant Feature Transform*), jehož autorem je D. Lowe [3], je velice dobrým algoritmem k vyhledání klíčových bodů. S postupem času se však začaly zvedat výpočetní nároky a lidé potřebovali rychlejší alternativu. V reakci na tyto požadavky počítačové komunity vznikl algoritmus SURF [13], který je založen na stejném principu a krocích, ale využívá jiné schéma, dosahuje lepších výsledků a je rychlejší.

Pro zvýšení výpočetní rychlosti využili autoři integrálního obrazu. Díky integrálnímu obrazu je pak možné v jakémkoliv čtvercovém regionu vypočítat sumu všech pixelů pouze sečtením čtyř čísel, což výrazně urychluje výpočetní náročnost.

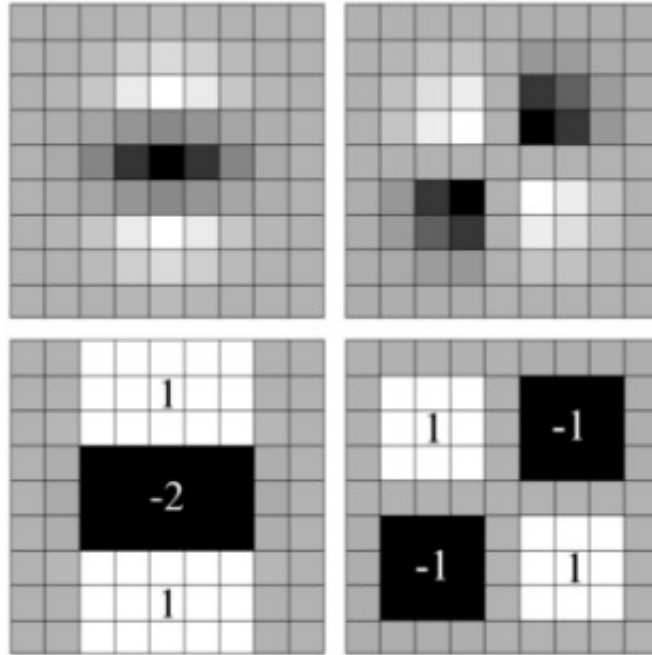
K samotnému nalezení významných bodů je využito Hessovy matice

$$H(x, \sigma) = \begin{pmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{pmatrix}, \quad (2.10)$$

kde x označuje souřadnice požadovaného bodu obrazu o měřítku σ . $L_{xx}(x, \sigma)$ je pak konvoluce druhé derivace Gaussovy funkce s originálním obrazem I v bodě x . Totéž platí pro $L_{xy}(x, \sigma)$ a $L_{yy}(x, \sigma)$.

Pro zjednodušení a urychlení výpočtů není tzv. scale space generováno postupnými konvolucemi Gausianu s originálním obrazem, ale jsou využity aproximace druhých derivací použitých v rovnici 2.10.

Aproximace na obrázku 2.2 odpovídají nejmenšímu možnému měřítku tedy 9×9 . Další měřítka jsou generována postupným zvětšováním konvoluční masky, která je pak aplikována na originální integrální obraz. Po nalezení determinantu zbývá určit klíčové body. Ty jsou označeny jako klíčové, jsou-li maximem ve svém $3 \times 3 \times 3$ okolí.



Obrázek 2.2: Diskretizované derivace Gaussovy funkce a jejich aproximace. Vlevo nahoře je zobrazena druhá derivace Gaussovy funkce podle y , pod ní její aproximace. Vpravo nahoře je druhá derivace Gaussovy funkce ve směru xy a pod ní její aproximace. Šedé plochy jsou rovny nule. (převzato z [13])

Deskriptor algoritmu SURF představuje 64-rozměrný vektor hodnot, vypočítaný z okolí daného klíčového bodu. Velikost tohoto okolí závisí na měřítku, na kterém byl daný bod detekován. Orientace tohoto okolí je určena před samotným procesem výpočtu deskriptoru. Pro výpočet deskriptoru je určeno čtvercové okolí kolem klíčového bodu, ze kterého se za pomoci filtru, využívajících Haarových vln, vypočítají jednotlivé hodnoty deskriptoru.

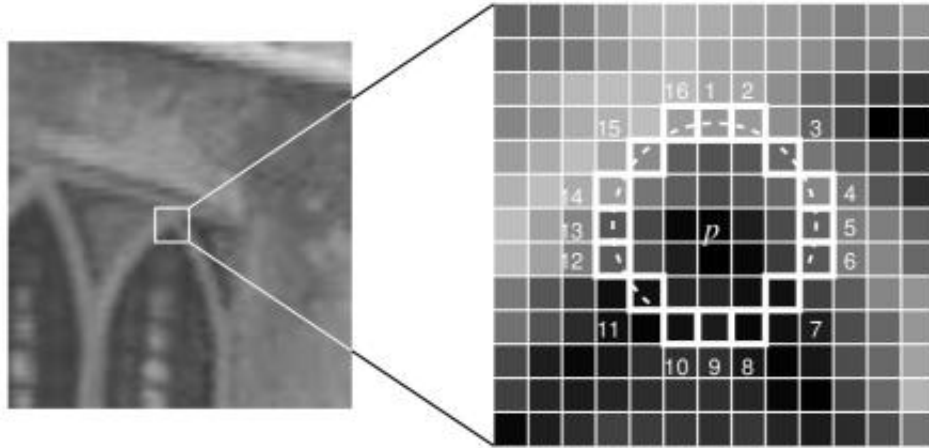
Oriented FAST and Rotated BRIEF - ORB

Metoda ORB využívá algoritmus *Features from Accelerated Segment Test* (FAST) k rychlému nalezení klíčových bodů a algoritmus *Binary Robust Independent Elementary Features* (BRIEF) k určení deskriptorů [4].

FAST je jedním z mnoha velmi dobrých algoritmů pro vyhledávání klíčových bodů, avšak svou konkurenci předčí díky své rychlosti. Tato metoda funguje na zcela jednoduchém principu. Vybere se pixel, který je zkoumán jakožto klíčový bod s intenzitou I_ρ , a ten je porovnán s hodnotami pixelů v kruhu okolo zkoumaného bodu. Tento proces je demonstrován na obrázku 2.3. Zkoumaný pixel je klíčovým bodem, pokud v jeho porovnávaném okolí existuje η na sebe navazujících pixelů, které jsou světlejší než $I_\rho + t$ nebo tmavší než $I_\rho - t$, kde t je zadaná prahová hodnota. Splňuje-li bod tyto podmínky, je zařazen mezi výsledné klíčové body [2].

Jelikož je takto nalezených bodů velké množství a mnoho z nich jsou navíc sousedící pixely, je na ně aplikována Harrisova metoda pro vyhledání rohů, která nám vybere nejlepších N klíčových bodů.

Jak již bylo zmíněno, k určení deskriptorů je využita metoda BRIEF. Zde nastává problém, jelikož tato metoda si nedovede příliš dobře poradit s rotací. K tomuto slouží



Obrázek 2.3: Pixel p porovnáváný s pixely ve svém okolí 1 – 16 (převzato z [2])

mírná modifikace tohoto algoritmu. Pro každou sadu n testů pro (x_i, y_i) vytvoříme $2 \times n$ matici S , která obsahuje souřadnice těchto pixelů. Poté je za pomoci opravné hodnoty θ nalezena otočená matice, s jejíž pomocí získáme pootočenou verzi matice S_θ [16].

Párování klíčových bodů

Jak již bylo řečeno v úvodu této podkapitoly, k vyhodnocení, zda si dva klíčové body odpovídají, jsou využity deskriptory těchto bodů. Postupně dojde k porovnání všech získaných deskriptorů z první fotografie s deskriptory fotografie druhé, výsledek tohoto procesu je ukázán na obrázku 2.4. K výpočtu vzdálenosti mezi jednotlivými dvojicemi vektorů jsou využity vhodné funkce a výsledná hodnota je porovnána se zadanou prahovou hodnotou, která určí, zda se konkrétní body shodují či nikoliv. Tuto prahovou hodnotu je třeba přizpůsobit metodám, které jsme využili k získání klíčových bodů. Tedy metoda SURF bude mít jinou prahovou hodnotu než metoda ORB.



Obrázek 2.4: Zobrazení spárovaných bodů (zdrojové snímky převzaty z [7])

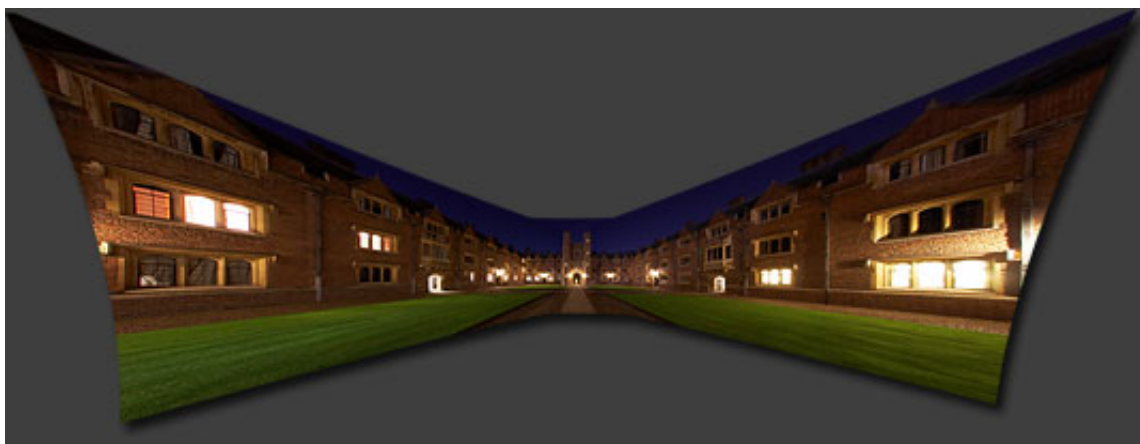
2.3 Projekce fotografií

Existuje mnoho způsobů pořizování panoramatických fotografií, ovšem nejčastějším případem je ten, při kterém dochází k čisté horizontální rotaci fotoaparátu bez jakékoliv jiné změny jeho polohy [16]. Zaměříme se proto na zpracování takto pořízených fotografií pomocí cylindrické projekce. Samozřejmě, ne vždy je tento způsob ideální, proto si představíme i další často využívané projekce, řekneme si jejich rozdíly a vhodnost využití.

Planární projekce

Planární projekce, nebo též rectilineární projekce, je nejpřirozenější projekce pro diváka, neboť zachovává zobrazení přímek, tak jak to vnímání přirozeně očekává. Této projekce se využívá především při tvorbě panoramat architektury, kdy vyžadujeme nezkreslené zobrazení objektů.

Tato projekce ovšem není použitelná u fotografií pořízených pomocí širokoúhlých objektivů, protože při tvorbě projekce je nutné okraje obrazu roztáhnout a u panoramat s úhlem záběru nad 120° by docházelo k velmi výrazné deformaci [12]. Problém motýlového efektu, který je vyobrazen na obrázku 2.5, je možné částečně eliminovat, zvolíme-li správně středový snímek. Tento efekt přesto nelze zcela odstranit a je nutno s ním počítat při skládání většího počtu fotografií.



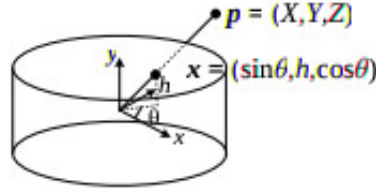
Obrázek 2.5: Ukázka motýlového efektu při použití planární projekce (převzato z [12])

Cylindrická projekce

Cylindrická projekce je velmi podobná planární s tím rozdílem, že roztahuje objekty čím jsou blíže k hornímu či spodnímu okraji válce, s nekonečným roztáhnutím na samotných okrajích. Z tohoto důvodu je cylindrická projekce nevhodná pro panoramata s velmi velkým vertikálním úhlem. Použitím této projekce dosáhneme přesnějšího zachování relativní velikosti objektů než u projekce planární. Tímto procesem ovšem dojde k zakřivení, což lze následně detekovat na výsledném obrazu [12].

Princip této projekce je takový, že se na plášť použitého válce přenesou obraz okolního prostoru. Její předností je vhodnost ke tvorbě horizontálních panoramat s úhlem pohledu až 360° . Při focení snímků pro tuto projekci, je doporučeno použití stativu. Díky tomu

budou fotografie téměř perfektně zarovnaný a při ořezávání nežádoucích okrajů u výsledného obrazu, získáme panoramatickou fotografii s největší plochou.



Obrázek 2.6: Projekce z 3D na souřadnice cylindrické projekce (převzato z [16])

Při zobrazení scény na povrch válce jsou jednotlivé body definovány úhlem θ a výškou h , s odpovídajícími souřadnicemi (θ, ϕ) v 3D cylindrickém zobrazení jež jsou dány

$$(\sin \theta, h, \cos \theta) \propto (x, y, f), \quad (2.11)$$

jak je ukázáno na obrázku 2.6. Z této závislosti můžeme získat rovnice souřadnic pro namapovanou scénu

$$x' = s\theta = s \tan^{-1} \frac{x}{f}, \quad (2.12)$$

$$y' = sh = s \frac{y}{\sqrt{x^2 + f^2}}, \quad (2.13)$$

kde s je rádius válce, který může být nastaven jako $s = f$ k minimalizaci zkreslení v oblasti středu obrázku[16]. Inverzní podoba těchto mapovacích rovnic je pak dána

$$x = f \tan \theta = f \tan \frac{x'}{s}, \quad (2.14)$$

$$y = h\sqrt{x^2 + f^2} = \frac{y'}{s} f \sqrt{1 + \tan^2 x'/s} = f \frac{y'}{s} \sec \frac{x'}{s}. \quad (2.15)$$

Po úspěšném namapování všech snímků stačí plášť válce rozvinout a tím získat výslednou neorezanou podobu panoramatu.

Sférická projekce

Tento způsob projekce je vhodný především tehdy, snažíme-li se vygenerovat panorama z fotografií, které byly pořízeny s určitým sklonem fotoaparátu, např. pohled z rozhledny na okolí pod ní.

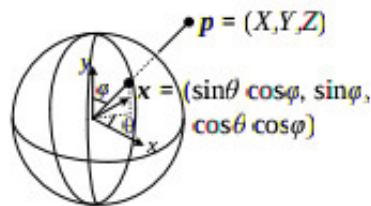
Další vhodné využití této projekce nastává, disponujeme-li mnoha fotografiemi scény viděné z jednoho místa. Při promítnutí takovýchto obrazů na povrch koule, jejíž střed určuje pozici fotografa, můžeme dosáhnout takové scény, jakou viděl původní pozorovatel. Takto lze pokrýt celou projekční plochu, bez nutnosti zachovat neměnnou pozici fotoaparátu s pouhou horizontální rotací.

Souřadnice v tomto typu projekce jsou dány jako dva úhly (θ, ϕ) , které jsou ve 3D dány

$$(\sin \theta \cos \phi, \sin \phi, \cos \theta \cos \phi) \propto (x, y, f), \quad (2.16)$$

jako je ukázáno na obrázku 2.7. Závislost mezi souřadnicemi je nyní

$$x' = s\theta = s \tan^{-1} \frac{x}{f}, \quad (2.17)$$



Obrázek 2.7: Projekce z 3D na sférické souřadnice (převzato z [16])

$$y' = s\phi = s \tan^{-1} \frac{y}{\sqrt{x^2 + f^2}}, \quad (2.18)$$

zatímco jejich inverzní podoba je dána

$$x = f \tan \theta = f \tan \frac{x'}{s}, \quad (2.19)$$

$$y = \sqrt{x^2 + f^2} \tan \phi = \tan \frac{y'}{s} f \sqrt{1 + \tan^2 x'/s} = f \tan \frac{y'}{s} \sec \frac{x'}{s}. \quad (2.20)$$

Berme v úvahu, že může být jednodušší pouhé generování a upravení (x, y, z) souřadnic z rovnice 2.16 s následným přizpůsobením pomocí z a změnou měřítka pomocí f [16].

2.4 Vady fotografií a jejich kompenzace

Během skládání panoramatických fotografií je možné se setkat s různými překážkami. Jednou z nich jsou vady ve snímcích samotných. Ty mohou být způsobeny nedokonalostí použitého objektivu, vlivem prostředí nebo fotografem samotným.

Chromatická vada

Jedná se o barevnou vadu čoček způsobenou závislostí ohniskové vzdálenosti čoček na vlnové délce světla. Každá barva světla se pak na čočce láme jinak, což ve výsledku způsobí barevné okraje, převážně viditelné v přechodech mezi tmavým pozadím a světlým objektem, nebo naopak. Tento jev se zvětšuje směrem od optické osy objektivu, tzn. u okrajů snímků bývá nejsilnější.

Tuto vadu lze částečně redukovat s pomocí speciálních čoček v objektivu, avšak v dnešní době již není problém tento jev odstranit pomocí různých programů.

Rozdílná expozice

S efektem „přeexponování“ a „podexponování“ se převážně můžeme setkat u fotografií pořízených v exteriéru, kde se během i několika málo sekund mohou změnit okolní podmínky a tím fotografie ovlivnit. Rozdílná expozice nejenže může způsobit problémy při párování fotografií, ale způsobí i viditelný přechod na výsledném panoramatu.

Vinětace

Běžným problémem při užívání širokoúhlých objektivů je, že výsledná fotografie má sklon k tmavším okrajům, jak je vidět na obrázku 2.8. Tento problém je obecně známým jako



Obrázek 2.8: Korekce vinětace samostatného obrázku [6] ©2008 IEEE: (a) Originální obrázek se silnou vinětací; (b) Vyladění vinětace metodou popsanou v dokumentu[10]; (c–d) Vyladění vinětace metodou popsanou v dokumentu[6] (převzato z [16])

vinětace a dochází k němu z mnoha různých důvodů včetně přirozené, optické a mechanické vinětace [16].

Jedním z možných způsobů, jak částečně eliminovat vinětaci u panoramatických fotografií je pořizování jednotlivých fotografií s dostatečným překrytím s fotografií předešlou. Jsou-li využity vhodné metody ke generování výsledného obrazu, problém vinětace může být potlačen při splynutí jednotlivých fotografií navzájem. Aplikováním tohoto postupu se ovšem nedocílí odstranění tmavších okrajů výsledného obrazu, pouze dojde k plynulejšímu přechodu mezi použitými snímky.

Změny ve scéně

Během pořizování snímků však nemusí být problémem pouze nepřesnosti způsobené objektivem a vlivy prostředí, ale i rozmístění objektů. Jakákoliv jejich změna může způsobit velké potíže [15]. Nejen že nepůsobí esteticky na výsledném obrazu, ale mohou velmi nepříznivě ovlivnit samotný proces spojování. Ovšem bez přímého zásahu do fotografií nelze s tímto problémem nic dělat.

2.5 Sloučení fotografií

Jakmile jsme docílili správného namapování scén na zvolenou plochu, stále nám zbývá vytvořit výsledný obraz. Můžeme fotografie bez dalších úprav pouze přeskládat přes sebe, ovšem okraje jednotlivých obrázků budou jednoduše rozpoznatelné a výsledek nebude působit profesionálním dojmem.

Musíme proto zvolit vhodný způsob, jak jednotlivé scény vzájemně sloučit, abychom odstranili viditelný přechod, ale nenarušili ostrost případných objektů. Nalezení takovéto metody je opravdu problematické, jelikož fotografie nemusí být pořízeny za stejných světelných podmínek. Také se zde mohou objevit pohybující se předměty, které na jednotlivých snímcích mění svou polohu (takzvaný ghosting).

V ideálním případě se při slučování posoudí každý pixel a přiřadí se mu váhová hodnota podle jeho důležitosti. Jak má ale stroj poznat, které body jsou důležitější než jiné? Vhodným způsobem k řešení tohoto problému je, rozpoznat potenciální duchy, pohybující se objekty, a nechat uživatele rozhodnout, zda se mají zcela vyřadit, nebo vhodně zakomponovat, třeba překrytím všech ostatních snímků.

Průměrování

Nejjednodušší nabízenou metodou je sečtení hodnot pixelu každého snímku v daném bodě a vypočítání jeho průměrné hodnoty

$$C(x) = \sum_k w_k(x) \tilde{I}_k(x) \Big/ \sum_k w_k(x), \quad (2.21)$$

kde $\tilde{I}_k(x)$ jsou namapované obrazy a $w_k(x)$ nabývá hodnoty 1 na validních pixelech a 0 jinde [16]. Přestože jsme docílili zdání pozvolnějšího přechodu mezi jednotlivými snímky, není tato metoda zcela ideální, jak je vidět na obrázku 2.9. Při kontrastnějších rozdílech na okrajích fotografie, budou jednotlivé snímky stále rozpoznatelné a narozdíl od obyčejného překrytí, se v obraze začnou vyskytovat průhledné předměty, nebyl-li daný objekt na všech překrývajících se fotografiích.



Obrázek 2.9: Použití metody (a) průměrování; (b) Feathering (převzato z [16] a upraveno)

Feathering

Tato hojněji využívaná metoda pracuje na podobném principu, ale do procesu výpočtu zavádí váhovou hodnotu pro jednotlivé pixely. Tato hodnota závisí na vzdálenosti pixelu od středu fotografie. Toho lze docílit vypočítáním vzdálenostní mapy nebo „grassfire“ transformace.

$$w_k(\mathbf{x}) = \arg \min_{\mathbf{y}} \{ \|\mathbf{y}\| \mid \tilde{I}_k(\mathbf{x} + \mathbf{y}) \text{ je neplatný} \}, \quad (2.22)$$

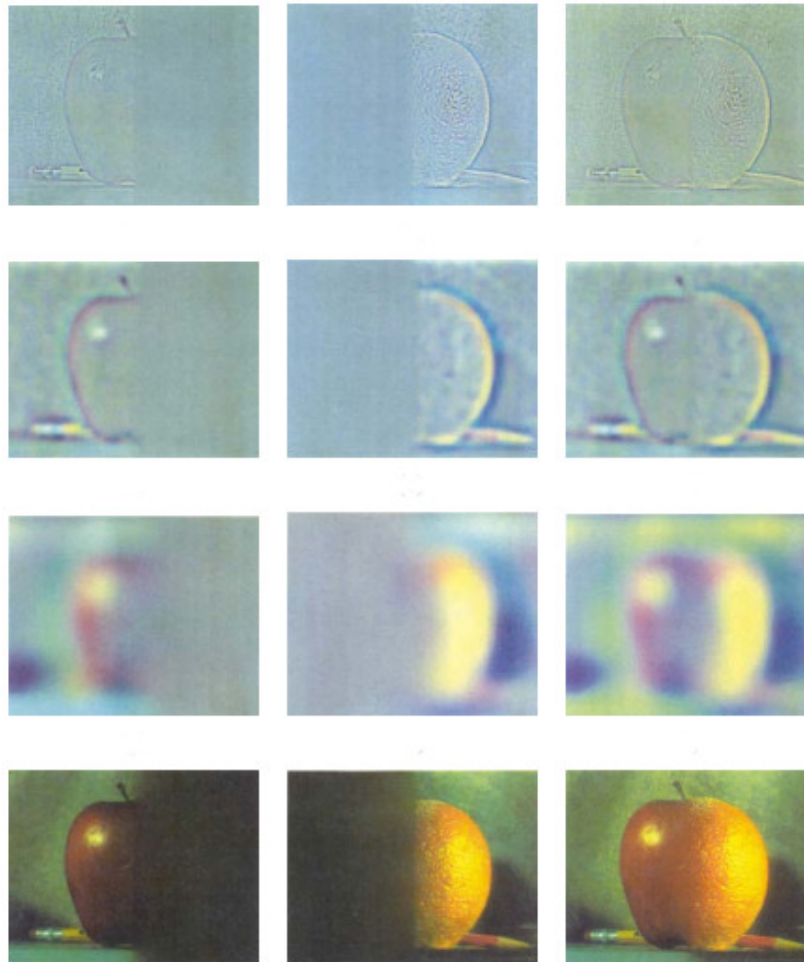
kde každý platný pixel je označen jeho Eukleidovou vzdáleností k nejbližšímu neplatnému pixelu [16].

Zavedením této váhové závislosti docílíme obstojných výsledků při slučování rozdílů snímků, avšak rozmazané a pohybující se předměty zůstávají i nadále problémem, jak lze vidět na obrázku 2.9.

Laplacian pyramid blending

Klíčem této metody je využití Laplaceovy pyramidy, která nám umožňuje postupný přechod podobných barevných odstínů, ale nedovolí přílišné rozšíření rozdílných textur, čímž se částečně předejde poloviditelným pohybujícím se předmětům.

K vytvoření výsledného obrazu je nejprve nutné každou vstupní fotografii rozložit na jejich vlastní Laplaceovu pyramidu 2.11. Každá skupina je pak vynásobena vyhlazovací váhovou funkcí, jejíž rozsah je úměrný úrovni pyramidy. Nejjednodušším způsobem vytvoření vah je použití masky snímku a z této masky pak vytvořit Gaussovou pyramidu. Každý obraz Laplaceovy pyramidy je pak vynásoben odpovídající Gaussovou maskou a obsah těchto dvou váhových pyramid je následně použit k sestavení výsledného obrazu.



Obrázek 2.10: Použití Laplaceovy pyramidy [5] ©1983, První tři řádky znázorňují vysokou, střední a nízkou frekvenci Laplaceovy pyramidy (z úrovní 0, 2 a 4). Levý a prostřední sloupec zobrazují originální snímky jablka a pomeranče vážené pomocí vyhlazovací interpolační funkce, zatímco pravý sloupec zobrazuje průměrný podíl (převzato z [16] a upraveno)

2.6 Existující řešení

Samotná myšlenka panoramatických obrazů vznikla již začátkem našeho letopočtu, ale až ve třicátých letech 19. století začaly vznikat první panoramatické fotografie. S nástupem digitálních fotoaparátů se spojování snímků přesunulo do oblasti počítačového vidění. Začalo vznikat mnoho aplikací, které se vzájemně předháněly ve kvalitě zpracování. Při návrhu řešení jsem některá tato řešení studoval a nechal se inspirovat v použitých metodách. Tímto jsem si také udělal představu o schopnostech konkurenčních programů a ujasnil si své požadované cíle.

Jak již bylo řečeno, existuje mnoho různých řešení pro tvorbu panoramat, avšak i v dnešní době jsou rozdíly mezi programy celkem značné. Hlavním rozdílem, který může odradit běžného fotografa, je cena těchto programů. Podle cenové kategorie se obvykle odvíjí i schopnosti aplikace, ale i zde se najdou výjimky.

PanoramaStudio 3

Nástroj pro tvorbu panoramat pod systémem Windows a Mac OS X. Tento nástroj je placený, ale umožňuje stažení zkušební verze. Je zde dostupná i dražší Pro verze, která k základním nástrojům přidává možnost zpracování fotografií ve více řadách, sférických panoramat a další. Obě verze disponují širokou nabídkou podporovaných formátů a automatickým detekováním fotoaparátu podle exif dat ze snímků určitých formátů.

Obsahuje široké množství nastavení, včetně 2D a 3D náhledu v editačním módu (pouze Pro verze). Jedná se o poměrně jednoduchý program, s jehož pomocí lze získat kvalitní panorama v několika krocích. Samozřejmostí jsou různé optimalizace k dosažení co nejvyššího výsledku.

Photoshop CC

Široce rozšířený nástroj umožňující nejen úpravu fotografií a tvorbu panoramat, ale také práci s jakoukoliv rastrovou grafikou. Tato aplikace od firmy Adobe je dostupná ve formě zkušební verze s následnou placenou verzí.

Na rozdíl od ostatních testovaných nástrojů, Photoshop CC výsledná panoramata neorezává, ale umožní uživateli dodatečnou úpravu podle jeho představ. K úpravě poskytne i masky jednotlivých fotografií. Mezi přední funkce rozhodně patří možnost dopočítání míst, která vůbec nebyla na fotografiích, jak je možné vidět na obrázku 2.6. K tomuto procesu se využijí informace z obrazu samotného. Musí se však počítat s případnými opakujícími se prvky, ale i přesto je výsledek velmi obstojný.



Obrázek 2.11: Ukázka dopočítání chybějící části panoramatu (převzato z [11] a upraveno)

Kapitola 3

Návrh řešení

V předchozí kapitole byly shrnuty informace z dosavadního vývoje, popsány některé z vyskytujících se vad fotografií a analyzovány dostupná řešení. Nyní je načase získané znalosti aplikovat do návrhu výsledného programu. Nejdříve definuji požadavky na aplikaci (sekce 3.1). Následovat bude návrh vyhledávání navazujících fotografií ve zdrojové složce (sekce 3.2). Ke generování výsledného obrazu budou využity výše zmíněné metody, jejichž výběr byl inspirován existujícími nástroji. Postup, který použiji při vývoji výsledného produktu, popíši v závěrečné části této kapitoly (sekce 3.3).

3.1 Požadavky aplikace

Cílem této práce je vytvořit aplikaci, která umožní automatické vyhledání panoramatických fotografií. Takto získané fotografie budou následně využity ke generování výsledných panoramat, pomocí kroků zmíněných v druhé kapitole této práce. Samotné skládání panoramat je velmi rozsáhlý problém a vytvoření algoritmu použitelného jako univerzální nástroj daleko přesahuje rozsah bakalářské práce. Zaměřil jsem se tedy na zpracování horizontálních fotografií s využitím cylindrické projekce.

Jak je z podkapitoly 2.6 patrné, existuje již dost kvalitních nástrojů k tvorbě panoramat, proto bych se rád zaměřil na samotnou automatizaci jejich vyhledávání a tvorby, jelikož tuto funkci žádný mnou testovaný nástroj neumožňoval. Aby byla konečná aplikace použitelná, měly by výsledné obrazy kvalitně dosahovat obrazů vytvořených jinými dostupnými produkty. Přehled všech požadavků je znázorněn tabulkou 3.1, ve které jednotlivé sloupce reprezentují testované nástroje a poslední sloupec určuje požadavky vytvořené aplikace.

	PanoramaStudio 3	Photoshop CC	panoramaStitcher
Automatizace	Ne	Ne	Ano
Detekce prostředí	Ano	Ano	Ano
Výběr projekce	Ne	Ano	Ano
odstranění přechodů	Ano	Ano	Ano
Ořezání obrazu	Ano	Ano	Ne

Tabulka 3.1: Přehled požadavků aplikace

3.2 Návrh automatizace

Uvažujme následující případ: Vrátili jsme se z dovolené a jako správný fotograf máme nafoceno stovky, ne-li tisíce, snímků. V této směsici jsou i panoramatické fotografie, které by jsme rádi zpracovali. Tento úkol si přímo žádá automatické zpracování, které za nás tyto snímky vyhledá a zpracuje. Z popsaného příkladu intuitivně vyplývá spuštění aplikace s adresou složky, ve které se nachází námi nafocené snímky.

Během implementace tohoto řešení jsem ovšem dospěl k názoru, že je třeba zauvažovat i nad jinou variantou automatizace. Jsou-li snímky jedné scény foceny nepravidelně, případně jsou-li mezi nimi i zcela nesouvisející fotografie, nemusí být korektně identifikovány a výsledný obraz bude neúplný. V této variantě navíc dochází i k analýze snímků, které nepatří do žádného panoramatu. To má za následek další navýšení časové a výpočetní náročnosti.

Problém také nastává, fotíme-li podobnou scénu, jako je scéna požadované panoramatické fotografie. V tomto případě dochází k nesprávnému porovnání klíčových bodů, které jsou na základě jejich podobného okolí označeny jako odpovídající si pár. Tak se do generování přimíchají i další, nechtěné snímky.

Vhodným řešením těchto problémů je poloautomatické zpracování. Fotografie rozdělíme do podsložek podle scény a aplikaci spustíme nad kořenovou složkou. Tímto způsobem je sice na uživateli, aby jednotlivé fotografie našel a roztrídil, na druhou stranu již víme, které soubory máme použít při generování, a proto můžeme využít jiný přístup při zpracování. Použitím této metody navíc celý proces výrazně urychlíme.

3.3 Návrh skládání fotografií

Základem celého projektu je možnost automatického zpracování. Tímto krokem je zahájena analýza snímků ve zdrojové složce a jejich korektní rozdělení podle toho, jakou scénu reprezentují.

Poté, co budou fotografie vyfiltrovány, se mohou přesunout k jejich individuálnímu zpracování, které, kromě několika krokového nastavení parametrů, poběží zcela samostatně. Tvorba panoramat bude rozdělena do dvou hlavních částí.

Přestože první část na první pohled nemusí vypadat důležitě, její úloha je mnohem větší, než by se mohlo zdát. Za pomoci vstupních snímků nastaví parametry prostředí a optimalizací, které se následně využijí při skládání „naostro.“ Ke zrychlení tohoto procesu, budu pracovat se zmenšenými verzemi originálních fotografií. V první řadě dojde k vyhledání klíčových bodů ve všech fotografiích a nalezení korespondence mezi nimi. Tyto údaje budou poté využity k nastavení parametrů prostředí a fotoaparátů. Podaří-li se správně odhadnout scénu, zakřívíme obrázky a jejich masky a použijí je k vyhledání okrajů.

Druhou částí je tvorba samotného panoramatu. Tento krok v sobě nese různé optimalizace, aby se docílilo co nejlepšího výsledku. V první řadě se aktualizuje měřítko, aby odpovídalo rozměrům originálních fotografií. Poté jen nastavíme slučovací metodu a s pomocí dat získaných v prvním kroku vytvoříme výsledný obraz.

3.4 Návrh postupu řešení

Jak je z popsaného návrhu vidět, celý proces je tvořen několika na sobě závisujícími kroky. Rozdělím tedy celý postup vývoje na bloky, abych přešel postupnému „probublávání“

chyb z počáteční fáze zpracování. Před dokončením všech bloků nebudu používat automatizaci z předešlé sekce, ale pouze dvě zvolené fotografie pro maximální eliminaci chyb.

V prvním bloku vyhledám klíčové body vstupních snímků pomocí metody SURF a tyto body následně spáruji. Ke kontrole si vyhledané body i vztahy mezi fotografiemi vykreslím do výstupních obrazů. Následovat bude nastavení prostředí z informací získaných analýzou snímků. Pro testovací účely zvolím planární projekci, pomocí které fotografie zakřívím. V závěrečném bloku fotografie sloučím průměrováním, abych zajistil základní funkčnost programu.

Jakmile budou výsledky dostatečně uspokojivé, začnu se soustředit na způsoby, jak potlačit, případně zcela eliminovat vady, které se mohou při slučování objevit. Budu-li s výslednými obrazy dostatečně spokojen, začnu se dle časových možností věnovat dalším cílům, jako rozšíření repertoáru použitých metod v jednotlivých krocích a porovnání jejich výsledků. Také bych rád aplikaci rozšířil o možnost zpracování snímků pořízených pod určitým úhlem.

Kapitola 4

Implementace a vyhodnocení

V této kapitole bude popsáno, co bylo ve výsledku této bakalářské práce implementováno. Bude přiblíženo, jakým způsobem byla implementovaná automatizace vyhledávání a následné zpracování. Na závěr budou ukázány vyrobené panoramata s použitím různých vstupů programu a jejich zhodnocením.

Implementaci jsem prováděl v jazyce C/C++, který mi dovoluje využití knihovny pro práci s fotografiemi.

Knihovna OpenCV

OpenCV je volně dostupná knihovna zaměřená na počítačové vidění a zpracování obrazu v reálném čase. Tato knihovna je multiplatformní, šířena pod BSD licencí a podporuje širokou škálu programovacích jazyků, patří k nim C/C++, Python a Java. S podporou OpenCL lze také využít hardwarové akcelerace[1].

Díky této knihovně budu moci jednoduše manipulovat se vstupními fotografiemi a s využitím již implementovaných funkcí pro spojování obrazů, hodlám vytvořit cílovou aplikaci.

4.1 Automatizace zpracování

Správný způsob implementace automatizace je velmi důležitý, protože právě tímto bodem se výsledný program bude lišit od existujících nástrojů.

Automatické zpracování vstupních souborů jsem rozdělil na dvě varianty, jak již bylo řečeno v podkapitole 3.2. Mezi těmito způsoby zpracování můžeme přepínat použitím volitelného parametru `-A`. Jako defaultní variantu jsem ponechal první uvedenou. Toto rozhodnutí jsem provedl na základě očekávaného chování programu, které uživatel předpokládá při zmínce o automatizaci.

Varianta mnoha souborů

Tato varianta je jak časově, tak i výpočetně velmi náročná. Toto souvisí s množstvím snímků, které je třeba zpracovat. Celý proces této varianty zpracování je demonstrován algoritmem 1.

V prvním kroku se prohledá zadaná složka. Fotografie, které v ní nalezneme, poté postupně načteme a vyhledáme jejich klíčové body. Pro ušetření paměťových nároků fotografie, ze kterých jsme získali klíčové body, opět uvolníme. Takto získané body následně využijeme


```

1 Vyhledej soubory;
2 Nastav vyhledávací metodu a měřítko;
3 Vyhledej klíčové body;
4 while Počet obrázků > 1 do
5   | Najdi odpovídající si klicové body;
6   | Najdi nejdelsi sekvenci navazujících snímků;
7   | if Nejdelsí sekvence <= 1 then
8   |   | Konec;
9   | end
10  | Vytvoř panorama;
11 end

```

Algorithm 1: Základní automatizace

ve vyhodnocení, zda jsou porovnávané snímky součástí stejné scény. Vzhledem k potenciálně velkému množství vstupních fotografií omezíme toto porovnání pouze na porovnání mezi sousední snímky. Takto se sníží časová náročnost vyhledávání, avšak se tímto docílí i nechtěného chování, které zamezí nalezení navazujících fotografií, nejsou-li v řadě hned za sebou.

Získané informace o návaznosti jednotlivých snímků využijeme k nalezení největší panoramatické fotografie. Snímky patřící do této scény vložíme ke zpracování do třídy **TPanorama**. Při tomto procesu snímky také odstraníme z našeho seznamu souborů. V tomto seznamu se opakovaně vyhledávají navazující obrazy s jejich následným zpracováním, dokud seznam obsahuje fotografie, nebo dokud nezjistíme, že již žádné dva snímky na sebe nenavazují.

Varianta složek

Tento způsob automatizace se příliš neliší od výše zmíněného, avšak jeho přední výhodou je rychlost a nižší výpočetní náročnost, tedy za předpokladu, že uživatel do podsložek umístil opravdu jen roztríděné fotografie panoramat. Ovšem i v tomto procesu dochází k ověřování, zda jsou všechny poskytnuté snímky z jedné scény, ale není nijak omezeno množství souborů. Hrubá podoba této metody zpracování je vyobrazen algoritmem 2.

```

1 Vyhledej složky;
2 for i < Počet složek do
3   | Vyhledej soubory;
4   | Nastav vyhledávací metodu a měřítko;
5   | Vyhledej klíčové body;
6   | Najdi odpovídající si klicové body;
7   | Najdi nejdelsi sekvenci navazujících snímků;
8   | Vytvoř panorama;
9 end

```

Algorithm 2: Složková automatizace

První krok probíhá obdobným způsobem, ale na rozdíl od předchozí varianty, která vyhledává fotografie pouze na úrovni kořenové složky, se snímky vyhledávají v podsložkách. Jsou-li v každé podsložce pouze fotografie jedné scény, docílíme zmenšení paměťových nároků, neboť nemusíme načítat všechny soubory, ale pouze snímky právě zpracovávaného

panoramatu. Tyto snímky můžeme navíc porovnávat mezi sebou navzájem a získat tak informace o jejich návaznosti, i když nejsou foceny postupně. Tento proces nám opět poskytne informace, že se jedná o jedno panorama a také přijatelné množství porovnávaných souborů.

4.2 Generování panoramat

Použitím knihovny *OpenCV* získám celou kolekci již předpřipravených metod pro analýzu, zpracování a spojování obrazů. Tyto funkce jsou navíc optimalizované pro co nejrychlejší vykonání operací. Hlavním úkolem bylo vybrat vhodné metody pro konkrétní problémy a zajistit jejich vzájemné propojení s minimalizací opakování stejných úkonů.

Jak již bylo zmíněno v kapitole 3, celý postup je rozdělen na dvě části, nastavení všech funkcí a následné vygenerování panoramtu.

Část první: Nastavení prostředí a optimalizací

Podle algoritmu 3 by se mohlo zdát, že se jedná vcelku o přímočarý proces. Skutečnost je poněkud jiná, protože v jakémkoliv kroku může nastat chyba, která celý dosavadní postup znehodnotí a je potřeba zajistit, aby se přerušila pouze tvorba tohoto panoramatu, ale samotná automatizace pokračovala. Největším problémem je úprava parametrů fotoaparátů, kde jsem se setkal s největším počtem selhání.

I nastavení centrální ohniskové vzdálenosti je neméně důležitým krokem. Tato hodnota se spočítá jako medián z hodnot ohniskových vzdáleností všech parametrů.

- 1 Nastav měřítko;
- 2 Vyhledej klíčové body a jejich odpovídající páry;
- 3 Použij klíčové body a jejich páry k vytvoření a nastavení fotoaparátů;
- 4 Nastav výchozí parametry fotoaparátů;
- 5 Uprav parametry podle klíčových bodů a jejich párů;
- 6 Vypočítej centrální ohniskovou vzdálenost;
- 7 Nastav korekci horizontálního pohybu;
- 8 Zakřiv obrazy a jejich masky;
- 9 Nastav kompenzaci expozice;
- 10 Vyhledej okraje;

Algorithm 3: Složková automatizace

V samotném procesu je mnoho drobných částí jako tvorba a ukládání do proměnných. Ty zde nejsou vyobrazeny, aby byl kód přehlednější a jsou zde shrnuty jen obecné kroky algoritmu.

Část druhá: Tvorba panoramatu

Jsou-li všechny parametry správně nastaveny, zbývá vytvořit výsledný obraz. K tomuto slouží druhá část tohoto procesu. Je nezbytně nutné, aby se upravily všechny potřebné parametry podle vypočítaného měřítka, jelikož do teď se pracovalo jen se zmenšenými verzemi fotografií. Následně vytvoříme **Blender**, který nakonec vygeneruje výsledek.

Než k tomuto kroku dojde, je potřeba upravit originální fotografie. Postupně dojde k načtení originálních snímků s jejich následným zakřivením. Dojde také k zakřivení jejich

masek. Následně se aplikuje proces pro kompenzaci expozice, odstranění okrajů a takto upravený obraz a maska se vloží do **Blender**. Jakmile dojde k zpracování všech snímků, nechá se vygenerovat výsledný obraz a ten se uloží do výstupní složky. Zjednodušená verze tohoto procesu je znázorněna algoritmem 4.

```
1 Uprav měřítko;
2 Vytvoř a nastav Blender;
3 for  $i < \text{Počet obrázků}$  do
4   Načti originální fotografii;
5   Zakřiv fotografii;
6   Vytvoř masku fotografie;
7   Zakřiv masku;
8   Kompenzuj expozici;
9   Zálohuj zakřivenou fotografii;
10  Odstraň okraje fotografie a masky;
11  Přidej do Blender;
12 end
13 Vytvoř výslednou scénu;
14 Ulož do souboru;
```

Algorithm 4: Složková automatizace

Proces generování panoramat je ve skutečnosti poněkud složitější. Pro zjednodušení jsem zde uvedl pouze hlavní kroky kroky.

4.3 Parametry programu

Jak již bylo zmíněno v dřívějších kapitolách, pro některé úlohy jsem implementoval více možných metod. Také zadávání různých vstupních hodnot v těchto metodách může ovlivnit výsledný obraz. Zvolil jsem proto jejich možnost úpravy pomocí vstupních parametrů.

Některé parametry jsou povinné a je nutné, aby je uživatel zadal. U ostatních lze ponechat výchozí hodnoty, avšak, nedojde-li k uspokojivému spojení, doporučuji vyzkoušet různé metody a hodnoty, které mohou být pro dané snímky vhodnější. Ukázky panoramat vytvořené se zadáváním různých vstupních hodnot jsou ukázány níže.

Spouštěcí parametry napsány mezi hranatými závorkami jsou nepovinné. Nemusí být dodrženo pořadí parametrů jako je v předpisu. Předpis:

```
-I cesta -O cesta [-A] [-R hodnota] [-F hodnota] [-M hodnota] [-W hodnota] [-B hodnota] [-S hodnota]
```

-I Cesta ke zdrojovým fotografiím

-O Cesta kam se budou ukládat vygenerované obrazy

-A Přepínání mezi variantami automatického zpracování

-R Hodnota určující s kolika sousedními snímky se bude daná fotografie porovnávat. Výchozí hodnota pro první typ automatického zpracování je **2**. Druhý typ automatického zpracování není porovnáváním omezeno, hodnota je tedy **-1**

- F** Použitá metoda pro vyhledání klíčových bodů. Lze použít **SURF** a **ORB**. Výchozí hodnota je **SURF**
- M** Prahová hodnota, podle které se určují odpovídající si klíčové body. Výchozí hodnota pro **SURF** je **0.65** a pro **ORB** je **0.3**
- W** Použitá metoda projekce. Podporované metody jsou: **plane**, **cylindrical** a **spherical**. Výchozí hodnota je **cylindrical**
- B** Metoda použitá ke splynutí obrazů ve výslednou scénu. Lze použít hodnoty **no** pro žádné splynutí, **feather** a **multiband**. Výchozí hodnota je **multiband**
- S** Síla splynutí obrázků. Kde platí $0 \leq \text{hodnota} \leq 100$. Výchozí hodnota je **5**



Obrázek 4.1: Ukázka použitých parametrů: -W plane (použity 4 fotografie)



Obrázek 4.2: Ukázka použitých parametrů: -B no (použity 4 fotografie)

4.4 Vyhodnocení

Během testování se aplikaci podařilo zpracovat všechny testované snímky. Pouze při skládání panoramatu výhledu na město Tallinn 4.5 se ojediněle nedařilo správně nastavit vlastnosti fotoaparátů. Tento jev může být způsoben pohybem fotografa mezi jednotlivými snímky, konkrétně pak mezi pořizováním fotografií v levé části výsledného obrazu.

Dojde-li k pohybu fotografa při snímání objektů ve velké vzdálenosti, nemusí být vzniklá chyba příliš velká a často lze snímky zpracovat. Problém nastává, vyskytují-li se ve snímcích



Obrázek 4.3: Ukázka použitých parametrů: -B feather (použity 4 fotografie)



Obrázek 4.4: Ukázka použitých parametrů: -W feather -S 50 (použity 4 fotografie)

předměty, na nichž je změna polohy patrná. Při zpracování se program snaží umístit snímek tak, aby vyhovoval všem ostatním a zde dochází k častému selhání. Díky nepříliš rozdílné poloze v tomto případě se mi podařilo docílit úspěšného složení použitím porovnávání fotografií pouze s jejími sousedními fotografiemi. Tímto se daný snímek nesnaží přizpůsobit všem, se kterými se překrývá, ale pouze předchozímu. Vhodnější způsob k úspěšnému složení by byla možnost eliminace nechtěných klíčových bodů. Tak by se ponechaly pouze ty, u kterých je změna pohledu zanedbatelná. V obou případech je nutno počítat s nedokonalým spojením, jako je zídka vlevo dole na obrázcích 4.5 (a) a (b).

U panoramatu 4.5 (a) si lze také povšimnout artefaktů v obloze, která nemá souvislou modrou barvu, jak by se očekávalo. Na rozdíl od vinětace, u které se očekávají tmavší okraje fotografií, se zde setkáváme se světlejší barvou v místech spojů. V tomto případě nám pomůže, úprava vstupních hodnot programu. Jak je vidět na obrázku 4.5 (b), zadáním vyšší hodnoty přechodu lze dosáhnout postupnější změny. Přechod nám zde pomůže s vyhlazením barev v obloze.



Obrázek 4.5: Výsledné obrazy, výhled na Tallinn (a) Hodnota splynutí 10 (b) Hodnota splynutí 100



Obrázek 4.6: Výsledné obrazy (a) Jyväskylä; (b) Repovesi



Obrázek 4.7: Výsledné obrazy (a) kostel Helsinky; (b) Dlouhé stráně

Kapitola 5

Závěr

Cílem bakalářské práce bylo navrhnout a implementovat algoritmus umožňující automatické vyhledání na sebe navazujících snímků ve zdrojové složce a jejich následné spojování ve výsledné obrazy. Tento cíl byl splněn.

K úspěšnému vypracování práce jsem nastudoval různé techniky spojování obrazů a na základě svého úsudku z nich vybral ty nejvhodnější. Jejich teoretické shrnutí je možno nalézt v kapitole 2. Dále jsem si vyzkoušel a otestoval dostupné nástroje, následně si určil požadavky aplikace se zaměřením na automatizaci vyhledávání, kterou tyto nástroje postrádají. Jelikož je aplikace zaměřená na automatické zpracování bez nutnosti uživatelských vstupů během jejího běhu, rozhodl jsem se pro konzolovou verzi implementovanou podle návrhu v jazyce C/C++.

Automatizace vytvořené aplikace funguje podle navrhnutého řešení. Podle zvoleného typu dokáže nalézt navazující fotografie a ty následně zpracovat. Nepodaří-li se některou ze scén vygenerovat, dané panorama se přeskočí a pokračuje se. Toto byl cíl této práce, aby výsledný nástroj pracoval samostatně, neukončil se při výskytu první chyby a vytvořené obrazy byly vysoké kvality.

Díky této práci jsem získal velmi užitečné zkušenosti z oblasti počítačového vidění. Seznámil jsem se s metodami pro práci s obrazem. Z těchto metod mě konkrétně zaujaly funkce k detekci objektů v obraze a jejich možnosti využití i v jiných problémech, než je skládání obrazu. Také jsem si uvědomil důležitost této oblasti a nelehké úkoly, kterými se člověk pracující v tomto odvětví musí zabývat.

V této práci plánuji pokračovat i nadále. Rád bych výslednou aplikaci dostal do stavu, který bude umožňovat její běžné používání. K dosažení tohoto cíle je nutné aplikaci výkonnostně zlepšit v části automatického vyhledávání. Také bych rád vytvořil jednoduché uživatelské rozhraní, pomocí kterého lze nastavit parametry programu. Tímto se samotný nástroj ničím neobohatí, ale stane se přívětivějším, díky čemuž k sobě přitáhne širší komunitu. Dalším vhodným rozšířením, kterým se však již zabývat nebudu, je uživatelské rozhraní, které umožní uživateli dopravit si výsledné panoramata podle jeho představ. Lze uvažovat o možnosti ořezání obrazu nebo jeho „narovnání“ podle zvolené přímky. Dalším užitečným nástrojem by byla možnost určení oblasti obrazu, ze které se mají vybírat klíčové body. Takto by se dalo předcházet selhání při nastavování parametrů fotoaparátu.

Literatura

- [1] *Open Source Computer Vision*. San Francisco, CA, USA: OpenCV, 2016.
URL <http://opencv.org/>
- [2] *FAST Algorithm for Corner Detection*. San Francisco, CA, USA: OpenCV, c2011-2014.
URL http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html
- [3] *Introduction to SIFT (Scale-Invariant Feature Transform)*. San Francisco, CA, USA: OpenCV, c2011-2014, [Online; navštíveno 11.1.2016].
URL http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#sift-intro
- [4] *ORB (Oriented FAST and Rotated BRIEF)*. San Francisco, CA, USA: OpenCV, c2011-2014, [Online; navštíveno 20.1.2016].
URL http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html
- [5] Burt, P. J.; Adelson, E. H.: *A Multiresolution Spline with Application to Image Mosaics*. ACM Trans. Graph., ročník 2, č. 4, Říjen 1983: s. 217–236, ISSN 0730-0301, doi:10.1145/245.247.
URL <http://doi.acm.org/10.1145/245.247>
- [6] FORSYTH, D.; TORR, P.: *Computer Vision – ECCV 2008*. Berlin: Springer, 2008, ISBN ISBN 978-3-540-88690-7.
- [7] GOLLA, R.: *Panorama – Image Stitching in OpenCV*. San Francisco, CA, USA: Ramsrigoutham, 2012, [Online; navštíveno 20.1.2016].
URL <https://ramsrighoutham.com/2012/11/22/panorama-image-stitching-in-opencv/>
- [8] HLAVATÝ, J.: *Soubor programů pro podporu výuky zpracování obrazů*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Brno ČR, 2013, vedoucí práce byl Mgr. Pavel Rajmíc, Ph.D.
URL https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=69323
- [9] HÁJOVSKÝ, R.; PUSTKOVÁ, R.; KUTÁLEK, F.: *Zpracování obrazu v měřicí a řídicí technice*. Vysoká škola báňská - Technická univerzita Ostrava, Ostrava ČR, 2012.
URL <http://www.person.usb.cz/archivcd/FEI/ZOMRT/Zpracovani%20obrazu%20v%20merici%20a%20ridici%20technice.pdf>

- [10] LEONARDIS, A.; BISCHOF, H.: *Computer vision - ECCV 2006*. Berlin: Springer, 2006, ISBN ISBN 978-3-540-33838-3.
- [11] MCCLELLAND, D.: *Correcting a distorted panorama | Photoshop | lynda.com*. Youtube [online], 8. 4. 2014.
URL <https://www.youtube.com/watch?v=QZFfJgRVqrE>
- [12] MCHUGH, S.: *Panoramic image projections*. Cambridge, UK: Cambridge in Colour, c2016, [Online; navštíveno 8.2.2016].
URL <http://www.cambridgeincolour.com/tutorials/image-projections.htm>
- [13] PEDERSEN, J. T.: *Study group SURF: Feature detection & description*. Technická zpráva, Aarhus University, Department of Computer Science, 2011, [Online; navštíveno 13.1.2016].
URL <http://cs.au.dk/~jtp/SURF/report.pdf>
- [14] PETYOVSKÝ, P.: *Přednáška kurzu MPOV Reprezentace a vlastnosti obrazových dat*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Brno ČR, 2010.
URL http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/03_Reprezentace_a_vlastnosti_obrazovych_dat.pdf
- [15] SEEMAN, M.: *Metody spojování obrazů*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, Brno ČR, 2005.
URL <http://www.fit.vutbr.cz/study/DP/DP.php?id=3390&file=t>
- [16] SZELISKI, R.: *Computer Vision, Algorithms and Applications*. London: Springer, 2010, ISBN ISBN 978-1-84882-935-0.

Přílohy

Seznam příloh

A	Obsah CD	34
B	Manual	35
	B.1 Požadavky	35
	B.2 Příklad	35
C	Popis tříd a souborů	36

Příloha A

Obsah CD

`/README.txt` - Manuál s obsahem CD. Součástí tohoto manuálu jsou také požadavky na překlad a spuštění.

`/panorama` - Přeložená výsledná aplikace

`/bp-xtrcka07.pdf` - Výsledná technická zpráva ve formátu PDF

`/doc/` - Zdrojové soubory technické zprávy

`/samples/` - Testovací fotografie

`/output/` - Výsledné scény testovacích fotografií

`/application/src/` - Zdrojové soubory výsledné aplikace

Příloha B

Manual

Tato kapitola obsahuje popis požadavků pro překlad a spuštění aplikace.

B.1 Požadavky

Pro překlad a správnou funkčnost aplikace je nutné disponovat volně dostupnou knihovnou `OpenCV` společně s jejími rozšiřujícími moduly. Při vývoji aplikace byla použita verze 3.1.

B.2 Překlad

Před samotným překladem aplikace je nutné zkontrolovat správné odkazování na `OpenCV` knihovny v souboru `application/Makefile`. Bez správného nastavení, nebude možno aplikaci přeložit.

Příloha C

Popis tříd a souborů

Pro přehlednost celého projektu je každá třída v samostatném souboru. Tyto soubory lze nalézt v `/application/src/`. Všechny hlavičkové soubory jsou také komentovány pro program Doxygen.

Hlavní soubor

Soubor `main.cpp` příliš mnoho důležitého kódu neobsahuje. Provádí pouze inicializaci třídy k automatizaci s následným zpracováním vstupních parametrů, jejichž hodnoty ověřuje a následně předává do `TAutomatization`. Je-li některý z parametrů nesprávně nastaven, dojde k vypsání příslušné chyby na standartní chybový výstup s následným ukončením.

V tomto souboru je na základě vstupních parametrů také zajištěno vypsání nápovědy, která obsahuje rozpis parametrů programu a jejich validní hodnoty.

Třída pro automatizaci

Je implementována v souboru `automatization.cpp`. Po její inicializaci dojde k nastavení výchozích hodnot. Povinné parametry pro vstupní a výstupní složku je nutné donastavit z `main.cpp` pomocí funkcí `setInputFolder(cesta)` a `setOutputFolder()`.

Pomocí funkce `run()` započne samotné zpracování fotografií podle zvolené varianty, jejichž popis je v podkapitole 4.1. Funkce `getDirContent(cesta)` získá všechny soubory ze zadané cesty a podle požadavku následným voláním `isDir` vybere pouze soubory nebo složky. Informace o všech snímcích jsou následně uloženy ve třídě `TPhotos`. Po nalezení navazujících snímků dojde k jejich vyjmutí z `TPhotos` a následnému předání do třídy `TPanorama` ke generování výsledného obrazu.

Třída pro fotografie

Třída `TPhotos` má na starost funkce pro zpracování zdrojových fotografií s následným uložením ve `vector`. Tento typ uložení jsem zvolil kvůli opakovanému používání `vectorů` s informacemi o všech snímcích. Tímto způsobem dojde k jejich oddělení a přitom snadnému použití bez opakovaného získávání potřebných dat.

Tato třída zajišťuje vše s obrazy spojené. Podle potřeby je načítá ze souborů, vyhledává klíčové body a odpovídající si páry, ale i ukládá všechny data.

Třída fotoaparátu

Velmi jednoduchá třída **TCameras**, ve které jsou uloženy informace o fotoaparátech, kterými byly snímky panoramatu pořizovány. Zavoláním několika funkcí z hlavní části programu dojde k jejich nastavení.

Třída pro tvorbu panoramat

Třída **TPanorama** je nejkompexnější z celého projektu. Neobsahuje příliš složitého kódu, ale jeho množství a vzájemná návaznost může být zpočátku matoucí.

Kromě třídy snímků **TPhotos** a fotoaparátů **TCameras** je možné v ní naléznout funkce k nastavení a použití projekce k zakřivení obrazů a jejich masek. Hlavní funkce této třídy zajišťuje správnou návaznost jednotlivých kroků pro tvorbu výsledného obrazu.