

# 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

## ŽIVÉ PANORAMA

BAKALÁŘSKÁ PRÁCE

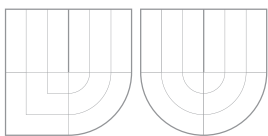
BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LENKA JALŮVKOVÁ

BRNO 2015



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

## ŽIVÉ PANORAMA

LIVE PANORAMA

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LENKA JALŮVKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2015

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2014/2015

**Zadání bakalářské práce**

Řešitel: **Jalůvková Lenka, Ing.**

Obor: Informační technologie

Téma: **Živé panorama  
Live Panorama**

Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte problematiku spojování fotografií do panoramat.
2. Seznamte se s problematikou zpracování videa ze streamujících kamer.
3. Navrhněte a implementujte algoritmus pro spojování snímků z videa kamery rotující kolem své osy.
4. Pořídte vhodné datové sady pro vyhodnocení algoritmu.
5. Vyhodnoťte vlastnosti algoritmu na pořizovaných datech.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Brown, Lowe: Automatic Panoramic Image Stitching using Invariant Features
- dále dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- body 1 a 2, značné rozpracování bodu 3 a případně bodů 4 a 5.

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: **Herout Adam, doc. Ing., Ph.D.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2014

Datum odevzdání: 20. května 2015

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Bžetěchova 2  
L.S.



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce je zaměřena na tvorbu živých panoramat. Jedná se o inovativní reprezentaci dat ze streamujících kamer. Vstupem je video ze streamujících kamer, výstupem je také stream - panorama neustále se překreslující podle aktuální polohy kamery. Výsledné řešení je založeno na algoritmu SIFT pro hledání klíčových bodů ve snímcích, dále na algoritmu RANSAC pro výpočet homografie sloužící k následnému plynulému spojování snímků. Spojení snímků jsou váhovány pro lepší překreslování v panoramatu. Funkčnost navrhnutého řešení byla otestována na řadě testovacích video nahrávek.

## Abstract

This bachelor's thesis is focused on creation of live panorama. This is an innovative representation of data from streaming cameras. The input is a video from the streaming camera and the output is also stream - constantly re-drawing panorama according to the current camera position. The resulting solution is based on the SIFT algorithm for finding keypoints in the frames. Furthermore, the RANSAC algorithm is used for calculating homography for stitching images together. Re-drawing the panorama requires weighting of edges. The functionality of the proposed solution was tested on a series of testing video recordings.

## Klíčová slova

panorama, spojování snímků, RANSAC, homografie, SIFT, planární projekce, cylindrická projekce

## Keywords

panorama, stitching, RANSAC, homography, SIFT, planar projection, cylindrical projection

## Citace

Lenka Jalůvková: Živé panorama, bakalářská práce, Brno, FIT VUT v Brně, 2015.

# Živé panorama

## Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Živé panorama“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

.....

Lenka Jalůvková

19. května 2015

## Poděkování

Na tomto místě bych ráda poděkovala mému vedoucímu doc. Ing. Adamu Heroutovi, Ph.D. za odborné vedení a za poskytnuté rady. Dále bych chtěla poděkovat mé rodině a svému příteli Jakubovi za podporu a obrovskou trpělivost.

# Obsah

<b>Seznam zkratk</b>	<b>3</b>
<b>1 Úvod</b>	<b>4</b>
<b>2 Panorama</b>	<b>5</b>
2.1 Získání panoramat	5
2.2 Planární projekce	7
2.3 Cylindrická projekce	8
2.4 Sférická projekce	8
2.5 Kubická projekce	8
<b>3 Tvorba panoramat</b>	<b>10</b>
3.1 Stanovení projekce	10
3.1.1 Planární projekce	10
3.1.2 Cylindrická projekce	11
3.2 SIFT	13
3.2.1 Detekce lokálních extrémů ve <i>scale-space</i>	13
3.2.2 Lokalizace klíčových bodů	15
3.2.3 Přiřazení orientací ke klíčovým bodům	15
3.2.4 Extrakce lokálních deskriptorů	15
3.3 RANSAC	16
<b>4 Realizace živých panoramat</b>	<b>18</b>
4.1 Extrakce snímků	19
4.2 Volba obrazové projekce	19
4.3 Nalezení shodných bodů SIFT	21
4.4 Homografie	21
4.5 Spojování snímků	22
4.6 Vytvoření videa	23
4.7 Uživatelské rozhraní	24

---

<b>5</b>	<b>Experimentální výsledky</b>	<b>26</b>
5.1	Tvorba testovacích sad . . . . .	26
5.2	Testování . . . . .	27
5.3	Výsledná panoramata . . . . .	27
5.4	Testování na reálných datech . . . . .	31
<b>6</b>	<b>Závěr</b>	<b>34</b>
	<b>Literatura</b>	<b>36</b>
<b>A</b>	<b>Obsah CD</b>	<b>38</b>

# Seznam zkratek

<b>2D</b>	Two Dimensional
<b>3D</b>	Three Dimensional
<b>DoG</b>	Difference of Gaussian
<b>EXIF</b>	Exchangeable Image File Format
<b>RANSAC</b>	RANdom SAmples Consensus
<b>SIFT</b>	Scale Invariant Feature Transform
<b>SURF</b>	Speeded Up Robust Features



# Kapitola 1

## Úvod

Panorama je vizuální reprezentace prostředí, které je viděno z jedné pozice v prostoru [21]. Živé panorama reprezentuje jiné pojetí klasických panoramat, jeho vstupem je video z kamer rotujících kolem své osy a výstupem je opět video - panorama neustále se překreslující podle aktuální polohy původní kamery.

Velkou výhodou je zobrazení celé snímané scény a nejen části, kterou zrovna vidíme na rotující kameře. Jedná se o inovativní využití mediálních dat, například u prezentace obcí, informací ze sjezdovek, povětrnostního zpravodajství a podobně.

Cílem této práce je navrhnout a implementovat metody, které umožní tvorbu těchto živých panoramat.

Samotný termín panorama, jeho historie, způsob jeho získávání a jednotlivé typy projekčních ploch, které jsou pro panoramata využívány jsou popsány v druhé kapitole.

V rámci třetí kapitoly bylo zapotřebí nastudovat a popsat použitelné projekce při tvorbě panoramat, typy zkreslení a způsob jejich eliminace, hledání klíčových bodů a výpočet homografie.

Samotná realizace živých panoramat je rozebrána ve čtvrté kapitole, která popisuje jednotlivé kroky realizace, a to extrakci snímků z videa, volbu obrazové projekce, hledání klíčových bodů, určení vzájemné polohy snímků a samotné spojení snímků. Dále vyhlazení přechodů mezi spojovanými snímky a jejich překreslování v panoramatu dle pohybu kamery.

Kapitola poslední shrnuje experimentální výsledky navrženého řešení, které bylo otestováno na sadě testovacích nahrávek. Parametry testovacích nahrávek jsou zde přehledně uvedeny v tabulkách, dále zde najdeme také obrázky znázorňující hotová panoramata. Výsledná živá panoramata jsou k dispozici na CD. V závěru je navržené řešení celkově zhodnoceno a navrženo možné pokračování projektu.

# Kapitola 2

## Panorama

Označení panorama se používá při širokoúhlé reprezentaci obrazu ve fotografii, kresbě, filmu či videu. Je to vizuální reprezentace prostředí, které je viděno z jedné pozice v prostoru [21].

Panoramatem je většinou chápán horizontální širokoúhlý snímek, ale existují i panoramata vertikální, ta však nejsou tak častá a používaná.

První panoramata a panoramatické kamery jsou doloženy již z roku 1843 [20], kdy Joseph Puchberger patentoval v Rakousku první panoramatickou kameru, která vytvářela až 150° panoramata. Jedno z panoramat pořízených v tomto období vidíme na obrázku 2.1. Snímek byl pořízen Georgem N. Barnardem z vrcholu hory v Tennessee r. 1864.



Obrázek 2.1: Tennessee r. 1864 [13].

### 2.1 Získání panoramat

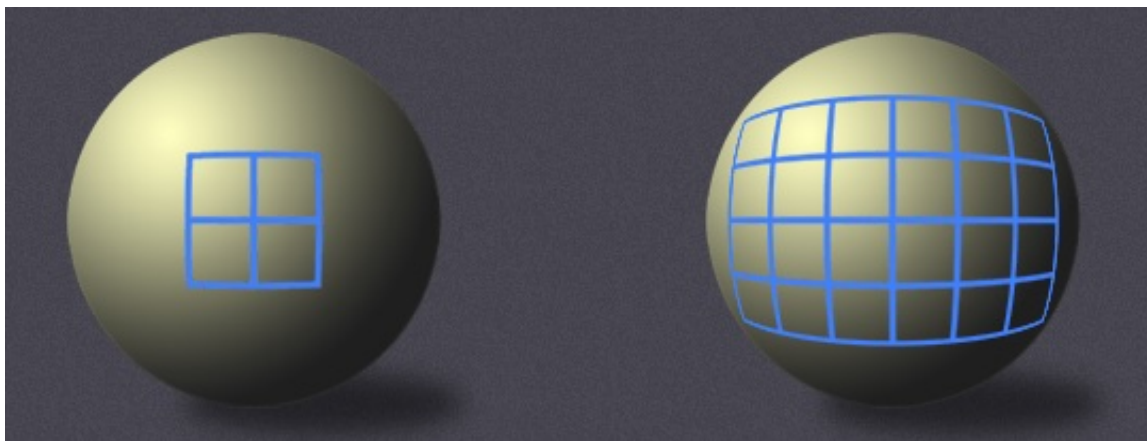
Existují dva hlavní způsoby, jak lze panorama získat.

(1) Pomocí speciálních rotujících panoramatických kamer, které vytvářejí panorama bez potřeby spojování jednotlivých snímků. Příkladem takovéto kamery je na-

příklad Panoscan MK-3<sup>1</sup>. Tyto kamery jsou však díky svým parametrům dost nákladné<sup>2</sup>, proto se spíše využívá druhého způsobu.

(2) Pomocí pořízení několika snímků jedné scény a následného spojování do jednoho širokoúhlého panoramatu. Při tomto spojování je potřeba dodržovat několik zásad již při pořizování dat. Jednotlivé dvojice snímků by se měly překrývat alespoň 25% plochy [19] a při jejich snímání je potřeba držet vodorovnou hladinu, nejlépe za pomoci stativu a vodováhy. Algoritmy používané pro toto spojování snímků (tzv. *image stitching*) patří mezi nejstarší a nejrozšířenější v počítačovém vidění [12]. Před samotným spojením jednotlivých snímků je nejprve nutné vyhledat vzájemné korespondence mezi nimi, k čemuž slouží algoritmy SIFT (*Scale Invariant Feature Transform*), SURF (*Speeded Up Robust Features*) a dále nalezení vzájemné polohy jednotlivých snímků, což je popsáno homografií algoritmu RANSAC (*RANdom Sam-ple Consensus*). Tyto postupy jsou blíže popsány v kapitole 3.

Panoramata existuje několik typů podle druhu obrazové projekce. Projekce nastávají v případech, kdy je plochý obraz mapován na zakřivené plochy či naopak. Při přímém úhlu pohledu je namapována na plochu téměř čtvercově, zatímco u širšího úhlu pohledu je patrné větší zkreslení, jak vidíme na obrázku 2.2.



Obrázek 2.2: vlevo: přímý úhel pohledu, vpravo: širší úhel pohledu [10].

Projekce, tedy i panoramata rozdělujeme do několika druhů, a to planární (přímochará), cylindrická, sférická a kubická [15]. Při spojování snímků do panoramat jsou nejčastěji využívány první dvě zmíněné projekce.

<sup>1</sup><http://www.panoscan.com/index.html>

<sup>2</sup>[http://luxurylaunches.com/gadgets/with\\_experience\\_training\\_and\\_a.php](http://luxurylaunches.com/gadgets/with_experience_training_and_a.php)

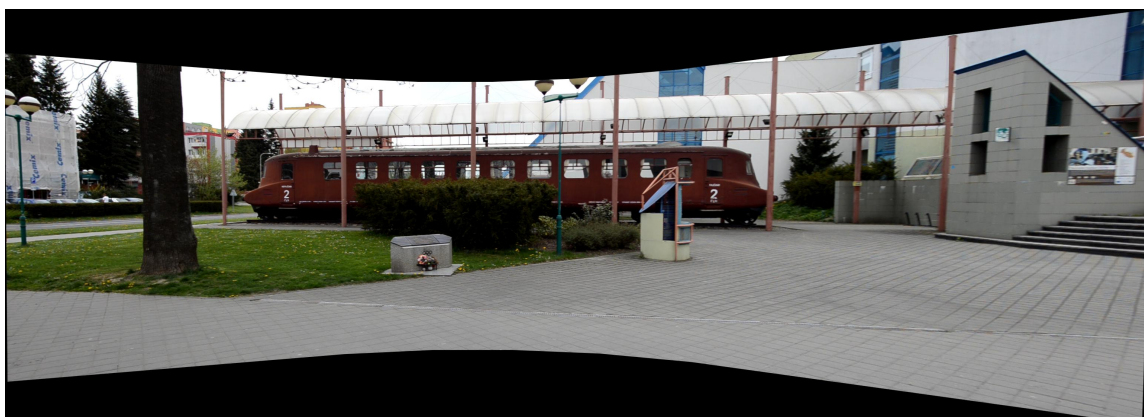
## 2.2 Planární projekce

Tento typ je vhodný při panoramatech architektury, protože jako jediný zachovává rovné čáry a nijak je neohýbá. Planární projekci lze však použít pouze v případě, kdy úhel snímání je malý ( $< 120^\circ$ ). Nad tento úhel je výsledné panorama nepřijatelné, protože roztažení obrazu způsobuje ztrátu ostrosti [3]. Vzniká tzv. motýlkový efekt, kdy se panorama stále více rozšiřuje, což je nežádoucí [2]. Oba tyto nedostatky vidíme na obrázku 2.3.



Obrázek 2.3: Planární panorama muzea TATRA.

Tento problém lze částečně řešit vhodným zvolením středového snímku, jak vidíme na obrázku 2.4, ale výsledek není ani přesto optimální. Při spojení většího počtu snímků je panorama silně prohnuto.



Obrázek 2.4: Planární panorama muzea TATRA se zvoleným středovým snímkem.

## 2.3 Cylindrická projekce

Cylindrická projekce je vhodná pro horizontální panoramata až do  $360^\circ$  [3]. Je to standardní typ používaný u panoramatických kamer s otočným objektivem. Cylindrická projekce zachová přesnější relativní velikosti objektů než planární projekce. Toto je však provedeno na úkor vykreslování rovnoběžných čar, které se v důsledku jeví jako zakřivené.

Příklad cylindrického panoramatu vidíme na obrázku 2.5. Princip jeho tvorby je dále popsán v kapitole 4.



Obrázek 2.5: Cylindrické panorama muzea TATRA.

## 2.4 Sférická projekce

Sférická projekce se též označuje jako kulová či  $360^\circ \times 360^\circ$ . Vzniká rotací snímáčiho přístroje okolo dvou kolmých os. Tento typ tedy zobrazuje plný rozsah, a to  $360^\circ$  svisle i  $360^\circ$  vodorovně. Nasnímaný prostor je jakoby zobrazen na vnitřní plochu koule. Příklad vidíme v levé části obrázku 2.6.

## 2.5 Kubická projekce

Kubická projekce nebo též označovaná jako krychlová, je obdobou projekce kulové. Rozdíl mezi těmito dvěma projekcemi je pouze v použité zobrazovací technologii. Kubické panorama je běžně použito jako zdroj několika panoramatických prohlížečů, jako QuickTime a Shockwave Panorama Viewer [4]. Kulový prostor je promítán na stěny krychle, jak vidíme na obrázku 2.6.



Obrázek 2.6: vlevo: sférická projekce, vpravo: kubická projekce [15].



# Kapitola 3

## Tvorba panoramat

V této práci je živé panorama tvořeno druhým způsobem tvorby panoramat, a to spojováním jednotlivých snímků, jak bylo zmíněno v předchozí kapitole. Při spojování snímků do panoramat je nutno nejprve určit vhodnou obrazovou projekci popsanou matematickým modelem. Dále najít shodné body v jednotlivých snímcích a následně je potřeba snímky správně vyrovnat, spojit a vyhladit přechody. Tento postup je níže podrobněji rozebrán.

### 3.1 Stanovení projekce

Předtím než jsou hledány klíčové body ve snímcích a jejich vzájemná pozice, je potřeba stanovit vhodnou obrazovou projekci. Při implementaci byly využity dvě hlavní projekce – planární a cylindrická, které jsou díky svým vlastnostem nejpoužívanější pro tvorbu panoramat [10]. Konkrétní příklady snímků těchto projekcí byly ukázány v předchozí kapitole, níže jsou popsány z matematického hlediska. Více se věnujeme projekci cylindrické, protože je univerzálnější, není zatížena motýlkovým efektem a ztrátou ostroty.

#### 3.1.1 Planární projekce

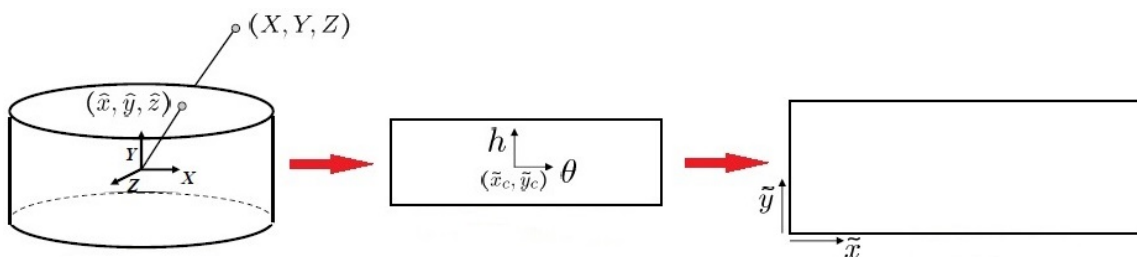
Planární projekce je jedna z nejjednodušších metod pro spojování snímků, které se pouze překryjí a rotují ve 2D [17]. Planární projektivní transformace je definována jako [17]:

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad (3.1)$$

neboli  $\tilde{x} = Hx$ , kde  $H$  je  $3 \times 3$  matice homografie. Dle tohoto principu je možné převést rovinné body v obecné poloze na jiné body pomocí projektivity.

### 3.1.2 Cylindrická projekce

Cylindrické spojování snímků je nejčastěji používáno v případě, že kamera rotuje pouze okolo své vertikální osy [5]. Za těchto podmínek jsou různě rotované snímky spojeny horizontálním překryvem. Tvorba cylindrického snímku je schematicky znázorněna na obrázku 3.1.



Obrázek 3.1: Schematický postup při tvorbě cylindrického snímku. Nejprve jsou body namapovány na válec (první zleva), poté je válec rozložen (uprostřed) a následně převeden na cylindrický snímek (vpravo) (převzato z [16] a upraveno).

Pro tento převod je nejprve potřeba namapovat 3D body  $(X, Y, Z)$  na válec podle vzorce [16]:

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Z^2}}(X, Y, Z). \quad (3.2)$$

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

$$(\sin\theta, h, \cos\theta) = (\hat{x}, \hat{y}, \hat{z}). \quad (3.3)$$

Nakonec je nutný převod do cylindrických souřadnic snímku:

$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c), \quad (3.4)$$

kde  $f$  značí ohniskovou vzdálenost v pixelech.

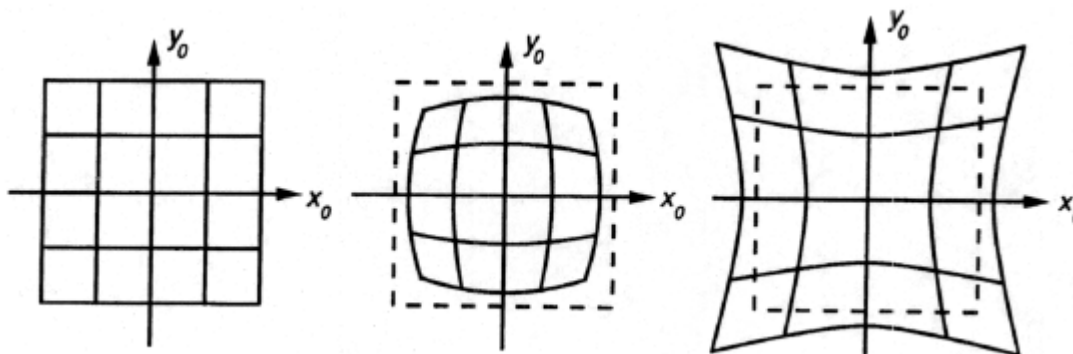
Ohniskovou vzdálenost lze většinou vyčíst z tzv. EXIF (*Exchangeable Image File Format*) informací snímků. Pokud však u snímků není uvedena, je potřeba ji odhadnout nebo spočítat podle vztahu [14]:

$$f_{pixel} = \frac{f_{mm}}{Sensor\_width_{mm}} * Image\_width_{pixel}, \quad (3.5)$$



kde  $f_{mm}$  je ohnisková v milimetrech,  $Sensor\_width_{mm}$  je šířka senzoru kamery či fotoaparátu, kterým byl snímek pořízen a  $Image\_width_{pixel}$  je šířka snímku v pixelech. Z videa by bylo možno ohniskovou vzdálenost dynamicky počítat, jak je blíže popsáno v článku [18], avšak takový výpočet je již nad rámec této práce.

Kvůli nepřesnému tvaru snímací čočky a převodu snímků do cylindrických souřadnic vzniká radiální zkreslení snímků. Odchyšky jsou nejvíce patrné u těch částí, které projdou skrz hranu čočky. Radiální zkreslení je nulové v optickém středu kamery a zvyšuje se, když se přesouváme k periferii. Typy zkreslení jsou znázorněny na obrázku 3.2.



Obrázek 3.2: Znázornění zkreslení, vlevo: bez zkreslení, uprostřed: soudkovité, vpravo: polštářkové [1].

V praxi je zkreslení malé a může být charakterizováno několika prvními členy Taylorova rozvoje okolo  $r = 0$ , většinou jsou použity první dva, jež jsou tradičně nazývány  $k_1$  a  $k_2$  [7].

Zkreslení se aproximuje polynomem sudého stupně:

$$x_d = x_n * (1 + k_1 * r + k_2 * r^2), \quad (3.6)$$

$$y_d = y_n * (1 + k_1 * r + k_2 * r^2), \quad (3.7)$$

kde  $(x_n, y_n)$  určuje původní umístění zkresleného bodu a  $(x_d, y_d)$  je nová pozice, jako výsledek korekce zkreslení.

## 3.2 SIFT

*Scale Invariant Feature Transform* byl navržen Davidem Lowem, publikován a patentován v r. 2004. Publikace ve které byl zveřejněn [9] patří mezi jednu z nejvýznamnějších v oboru počítačového vidění, což potvrzuje i téměř 30000 jeho citací. Mezi hlavní výhody tohoto algoritmu patří nezávislost na měřítku, změna snímacího úhlu, velká odolnost vůči šumu a změně osvětlení [8].

V kontextu SIFT se rozlišují tzv. klíčové body (lokální extrémů v prostoru měřítek, v tzv. *scale-space*) a *features*, které jsou tvořeny okolím klíčových bodů. Algoritmus SIFT slouží právě k detekci a popisu tzv. *features* v obrázku. Každá *feature* je popsána tzv. lokálním deskriptorem, který má podobu histogramu gradientů v okolí klíčového bodu.

David Lowe rozděluje algoritmus SIFT do čtyř fází [9], a to (1) detekci lokálních extrémů ve *scale-space*, (2) lokalizaci klíčových bodů, (3) přiřazení orientací ke klíčovým bodům a (4) extrakci lokálních deskriptorů.

### 3.2.1 Detekce lokálních extrémů ve *scale-space*

Aby bylo dosaženo nezávislosti na měřítku, musí být zohledněna všechna možná měřítka. Pro detekci napříč více měřítky je tedy potřeba sestavit třetí dimenzi, tedy údaj o prostorové frekvenci. Tohoto je dosaženo opakovaným vyhlazováním (konvolucí) vstupního obrazu  $I(x, y)$  pomocí Gaussovy funkce  $G(x, y, \sigma)$  [9].

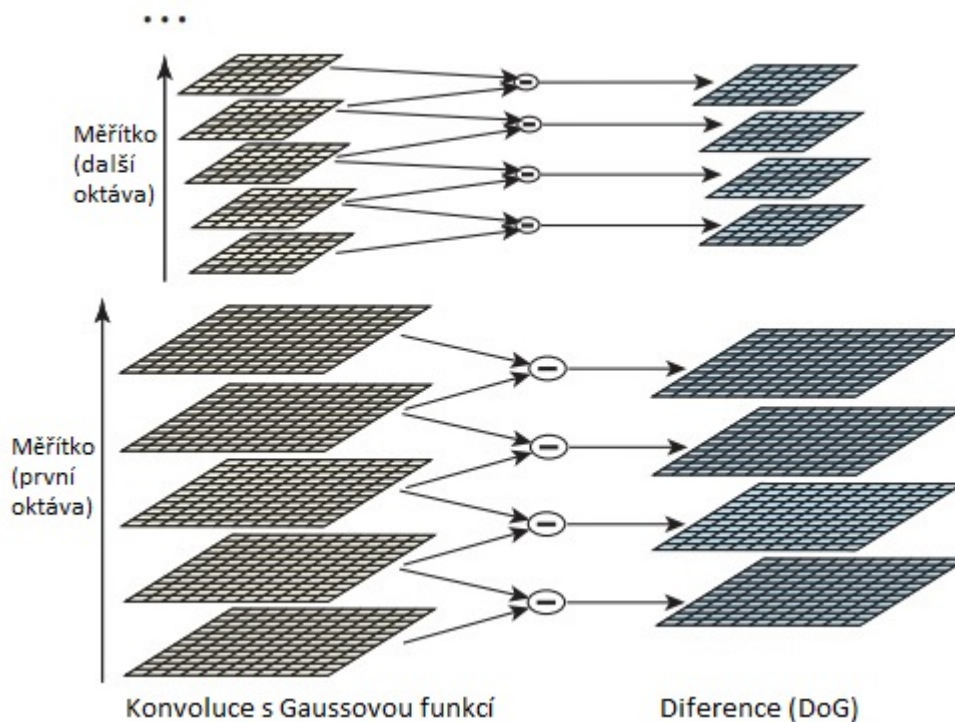
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (3.8)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-x^2+y^2/2\sigma^2}, \quad (3.9)$$

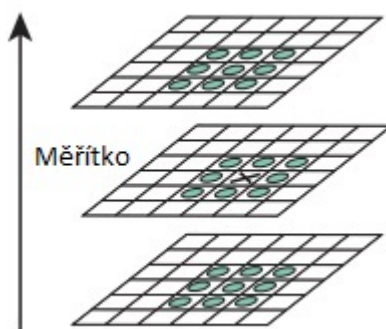
kde  $x$  a  $y$  značí souřadnice pixelu v obraze a parametr  $\sigma$  určuje šíři Gaussovy funkce, což odpovídá mezní frekvenci propusti nebo stupni vyhlazení.

Opakovaným filtrováním původního obrazu pak dostaneme více obrazů ve více měřítcích a sestavením těchto obrazů postupně od největšího po nejmenší měřítko dostáváme 3D prostor  $L(x, z, \sigma)$ , tedy jednu oktávu *scale-space*, jak vidíme na obrázku 3.3.

Z výsledných rozdílových snímků DoG *Difference of Gaussian* (vpravo 3.3) se detekují extrémů v každé oktávě. Maxima a minima jsou detekována porovnáváním jednotlivých bodů s jejich 26 sousedy (8 bodů v daném obrazu, 9 bodů nad a pod zvažovaným obrazem). Princip této detekce je názorně ukázán na obrázku 3.4.



Obrázek 3.3: Pro každou oktávu je původní snímek opakovaně konvoluován s Gaussovou funkcí a výsledkem je sada *scale-space* snímků (levá část). Výsledné snímky jsou mezi sebou odečteny a vznikají rozdílové snímky DoG (pravá část). Každá oktáva je podvzorkována na polovinu a poté použita pro další oktávy (převzato z [9] a upraveno).



Obrázek 3.4: Hledání extrémů ve *scale-space*. Křížkem je označen bod pro který se hledají extrémů, kolečky jsou vyznačeni jeho sousedé, se kterými je vyšetřovaný bod porovnáván (převzato z [9] a upraveno).

### 3.2.2 Lokalizace klíčových bodů

Dalším krokem, po nalezení lokálních extrémů metodou porovnávání se sousedními pixely, je ohodnocení každého bodu z hlediska umístění, měřítka a poměru zakřivení. Body s nízkým kontrastem jsou citlivé na šum a považují se za nestabilní. Tyto body pak mohou být odstraněny.

Nevýhodou funkce DoG je, že mohou být nalezeny i extrémy podél hran, které jsou nepřesně lokalizovány a citlivé na šum. Aby byla zvýšena stabilita detektoru je dalším krokem odstranění bodů, které byly lokalizovány podél hran. Toto je provedeno podobným způsobem jako u Harrisova detektoru, tedy použitím Hessovy matice a jejich vlastních čísel [11]. Použitím těchto postupů pak značně eliminujeme množství detekovaných bodů, přičemž nám zůstanou pouze stabilní body.

### 3.2.3 Přiřazení orientací ke klíčovým bodům

Klíčovým bodům je přiřazena orientace, čímž je dosažena nezávislost na rotaci. Pro každý klíčový bod je vybráno měřítka z vyhlazeného obrazu, jehož tvorba byla popsána výše. U každého obrázku  $L(x, y)$  v tomto měřítku se počítá velikost gradientu  $m(x, y)$  a jeho orientace  $\theta(x, y)$  pro každý pixel [9]:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}, \quad (3.10)$$

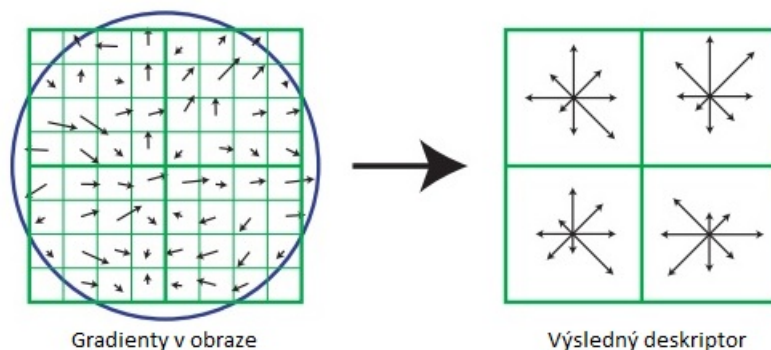
$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right), \quad (3.11)$$

Z gradientů vypočtených z okolí klíčového bodu je vytvořen histogram o 36 položkách pokrývajících úhel  $360^\circ$ . Každý vzorek souseda v oknu rámce je přidán do histogramu a jako váha je použita velikost gradientu a gaussovská váha kruhového okna s  $\sigma$ , které je  $1,5 \times$  větší než měřítka klíčového bodu. Poté je detekována nejvyšší hodnota v histogramu a také další místní vrchol, který dosahuje 80% výšky nejvyššího. Tyto hodnoty jsou pak použity jako orientace klíčového bodu.

### 3.2.4 Extrakce lokálních deskriptorů

Předchozí kroky přiřadily každému klíčovému bodu polohu v obrázku, pozici ve *scale-space* a orientaci. To zajišťuje znovu vytvořitelný 2D systém souřadnic, což poskytuje invarianci vzhledem k těmto parametrům, ve kterém se popisuje deskriptor. Dalším krokem je spočítání lokálního deskriptoru, který dosáhl nezávislosti na dalších parametrech, jako jsou změna osvětlení či místa při pořizování snímku. Pro rozpoznání 3D objektu je nejlepší popis bodů pomocí gradientů. Na podobném principu funguje

i biologický proces vidění. Princip výpočtu deskriptorů je znázorněn na obrázku 3.5. V okolí klíčového bodu jsou vypočteny velikosti gradientů a orientace dle vzorců 3.10 a 3.11.



Obrázek 3.5: Deskriptor je tořen gradienty, které jsou vypočteny v okolí každého klíčového bodu, jak je naznačeno vlevo. Ty jsou váhovány Gaussovým oknem, které je naznačeno modrým kruhem. Z každého regionu  $4 \times 4$  je sestrojen histogram orientací. Výsledný deskriptor ve velikosti  $2 \times 2$  je znázorněn vpravo. V algoritmu je použit deskriptor  $4 \times 4$  vypočtený ze  $16 \times 16$  pixelů (převzato z [9] a upraveno).

Jak vidíme na obrázku 3.5, pro každou orientaci každého klíčového bodu je okolí klíčového bodu rozděleno do  $4 \times 4$  oblastí. Pro každou oblast je spočten histogram gradientů o 8 směrech. Výsledný deskriptor klíčového bodu je tvořen 3D histogramem o  $4 \times 4 \times 8$ , což odpovídá 128 složkám.

### 3.3 RANSAC

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

Princip algoritmu je shrnut v bodech níže [6]:

---

#### Algorithm 1 RANSAC

---

- 1: Náhodný výběr  $n$  prvotních korespondencí
  - 2: Výpočet odhadu matematického modelu
  - 3: Ověření přesnosti vypočteného odhadu modelu pro všechny korespondence
  - 4: Zjištění, kolik korespondencí je vhodných pro hledaný model vyhovující zvolenému prahu
  - 5: Opakování kroků 1 až 4 po zadaný počet opakování (max.  $N$ )
-

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

Počet opakování  $N$ , které je potřebné pro výpočet korespondencí je tedy nastaven podle [6]:

$$1 - p = (1 - u^m)^N, \quad (3.12)$$

po vyjádření  $N$ :

$$N = \frac{\log(1 - p)}{\log(1 - (1 - v)^m)}, \quad (3.13)$$

kde  $u$  značí pravděpodobnost, že označený bod odpovídá matematickému modelu a je tím pádem *inlier* a  $v = 1 - u$  je pravděpodobnost získání *outlier*. Parametr  $m$  značí počet vybraných korespondencí pro výpočet odhadu modelu.

Dále se stanoví pravděpodobnost  $p$  nalezení správného modelu. Nejčastěji se volí jako 0,99, což značí, že alespoň jeden ze souborů náhodných vzorků neobsahuje *outlier*.

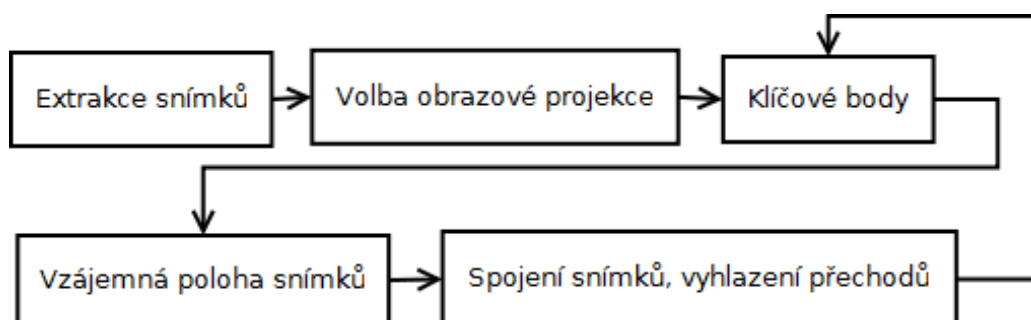
# Kapitola 4

## Realizace živých panoramat

Tvorba panoramatických fotografií nemusí být tak jednoduchá, jak se na první pohled může zdát. Celý postup pro získání živého panoramatu z videí se skládá z několika na sebe navazujících kroků:

- (1) Extrakce snímků z videa a jejich případné zmenšení.
- (2) Převod jednotlivých video snímků do vhodné projekční plochy, jejíž výběr je závislý od rotačního úhlu kamery v původním videu.
- (3) Nalezení klíčových bodů a deskriptorů ve snímcích.
- (4) Výpočet homografie pro nalezení vzájemné pozice klíčových bodů.
- (5) Plynulé spojení snímků a vyhlazení přechodů.
- (6) Překreslování panoramatu podle pohybu kamery zprava doleva.
- (7) Vytvoření videa.

Schematický postup je znázorněn na obrázku 4.1, kdy první dva kroky jsou provedeny pouze jednou, zbytek je nalézán ve smyčce až do maximálního počtu zpracovávaných snímků.



Obrázek 4.1: Schematický postup hlavních kroků tvorby živého panoramatu.

Tato kapitola popisuje vlastní implementaci výše uvedeného postupu. Při realizaci bylo vytvořeno několik samostatných funkcí v prostředí *Matlab R2013a*, za použití

*Image processing*, *Computer Vision System*, *Symbolic Math* a dalších standardních toolboxů a využití SIFT implementace Davida Lowe<sup>1</sup>. Všechny vytvořené funkce jsou komentovány a v následujícím textu stručně charakterizovány.

## 4.1 Extrakce snímků

Nejprve jsou z videa vyextrahovány jednotlivé snímky. Z videí se cyklicky vybírají snímky s určitým krokem, který je dán video frekvencí (většinou 25). Tyto snímky se ukládají a dále zpracovávají dle níže popsaných postupů. Uživatel dále nastaví, zda chce snímky ponechat v originální velikosti nebo zda dojde k jejich zmenšení. Snímky by však měly zůstat přiměřené velikosti, např.  $640 \times 360$ , protože doba výpočtu pro větší snímky je podstatně delší. Extrakce snímků je implementována ve funkci `videoParsing.m`.

## 4.2 Volba obrazové projekce

Jak již bylo zmíněno, cylindrická projekce je pro náš účel univerzálnější a použitelnější. Pro planární projekci byl vytvořen pouze ukázkový skript `planarProjection.m` vhodný k ověření chování tohoto typu projekce a určen pro menší počet snímků. Výstup již byl ukázán na obrázcích 2.3 a 2.4.

Pro tvorbu živých panoramat byla použita projekce cylindrická, při které je potřeba každý snímek převést do cylindrických souřadnic dle rovnic 3.2, 3.3, 3.4. Díky tomuto převodu a nepřesnému tvaru snímáčí čočky vzniká radiální zkreslení snímků. Eliminace zkreslení je řešeno pomocí vztahů 3.6 a 3.7.

Tato funkcionalita je řešena ve funkci `cylCoord.m` s parametry *image*, *f*,  $k_1$  a  $k_2$ . Parametr *image* reprezentuje snímek, který chceme převést do cylindrických souřadnic, *f* je ohnisková vzdálenost v pixelech, při které byl snímek pořízen a parametry  $k_1$ ,  $k_2$  jsou potřeba pro eliminování radiálního zkreslení.

V implementaci je počítáno s fixní ohniskovou vzdáleností, kterou je potřeba před spuštěním nastavit. Pokud není uvedena v EXIF informacích snímků, lze ji spočítat dle vztahu 3.5. Ohniskovou vzdálenost lze také zjistit například použitím kalibračního toolboxu<sup>2</sup>, nebo ji kreativně změřit za pomoci knihy a krabice<sup>3</sup>, jak je blíže popsáno Brettem Allenem.

<sup>1</sup><http://www.cs.ubc.ca/~lowe/keypoints/>

<sup>2</sup>[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)

<sup>3</sup><http://bit.ly/1zRytWT>



Nejlepší kombinace parametrů  $k_1$  a  $k_2$  u jednotlivých testovacích videí byla stanovena na základě prostého prohledávání a následného subjektivního vyhodnocení. Prosté prohledávání bylo realizováno paralelním forem. Tato funkčnost je realizována ve skriptu `optimization.m`, kde si uživatel ve funkci `matlabpool` nastaví příslušný počet jader, na kterých bude výpočet probíhat.

Prosté prohledávání bylo realizováno nejprve v rozsahu  $-2; +2$ , jako nejvhodnější se však projevil interval  $-0,3; +0,3$ . Ve většině případů vycházela nejlépe kombinace  $k_1 = -0,2$  a  $k_2 = 0,2$ . Tuto kombinaci při spojení dvou snímků vidíme na obrázku 4.2. Pro srovnání je na obrázku 4.3 v levé části kombinace  $k_1 = -0,2$  a  $k_2 = -0,2$  a v pravé části  $k_1 = 0,2$  a  $k_2 = -0,2$ .



Obrázek 4.2: Spojené dva cylindrické snímky s  $k_1 = -0,2$  a  $k_2 = 0,2$ , nejlepší kombinace parametrů.

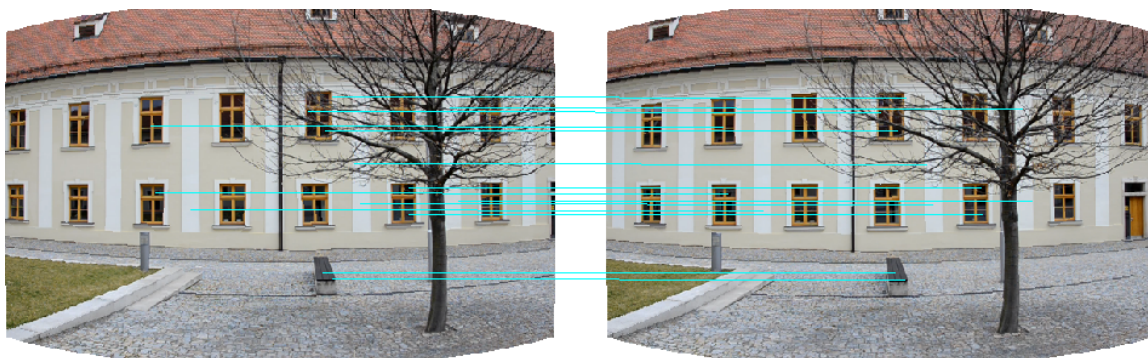


Obrázek 4.3: Spojené cylindrické snímky: vlevo  $k_1 = -0,2$  a  $k_2 = -0,2$ , okraje výsledného snímku jsou již příliš rozmazané a zkreslené; vpravo:  $k_1 = 0,2$  a  $k_2 = -0,2$  spojené snímky jsou vůči sobě mírně posunuty, jak lze vidět na okapové rouře přibližně uprostřed snímku.

### 4.3 Nalezení shodných bodů SIFT

Aby bylo možné jednotlivé cylindrické snímky spojit dohromady, je potřeba nalézt klíčové body a *features* mezi jednotlivými dvojicemi snímků. K tomuto účelu byl použit detektor klíčových bodů SIFT od Davida Lowe, jehož princip byl blíže popsán v kapitole 3.2, licence společně se všemi dalšími informacemi je přiložena na CD. Pro správné nalezení klíčových bodů je obecně doporučeno překrytí jednotlivých snímků alespoň 25% [19], při menším překrytí nemusí být nalezeny správné klíčové body, což by následně vedlo ke špatnému spojování snímků.

Dvojici snímků a jim odpovídající vyznačené klíčové body vidíme na obrázku 4.4. Klíčových bodů je většinou nalezeno mnohonásobně více (ve snímku  $640 \times 360$  okolo 1000) než vidíme na obrázku 4.4, na něm je jich však pro lepší přehlednost zobrazeno jen několik.



Obrázek 4.4: Nelezené klíčové body mezi dvojicí snímků.

### 4.4 Homografie

Po nalezení klíčových bodů, které popisují jednotlivé snímky je potřeba nalézt jejich vzájemnou polohu. Toto je realizováno ve skriptu `ransac.m`, který na základě podobnostních metod a výpočtu matice homografie nalezne nejpravděpodobnější souvislost mezi významnými body.

Počet iterací potřebných k nalezení matice homografie je stanoven podle rovnice 3.13 a míra pravděpodobnosti nalezení správného modelu je nastavená na 0,99.

## 4.5 Spojování snímků

Finálním krokem k sestavení statického panoramatu je samotné spojování snímků a vyhlazení přechodů mezi nimi. Tato funkčnost je realizována ve skriptu `stitch.m`.

V implementaci při spojování jednotlivých snímků kontrolujeme, zda jsou oba snímky neprázdné. Pokud je jeden ze snímků prázdný, ponecháme ve výsledku pouze neprázdný snímek. Pokud jsou oba neprázdné, spojíme první čtvrtinu z prvního snímku, zbylou část ze snímku druhého, jak je naznačeno na obrázku 4.5. Experimentálně tento postup poskytoval nejlepší výsledky.



Obrázek 4.5: Princip spojování dvou snímků.

Okraje připojovaného snímku váhujeme, aby byl spoj plynulejší. Důležitou roli hraje váhovaná šířka okraje, jak vidíme na obrázcích 4.6 a 4.7, kde v prvním případě jsou přechody mnohem více zřetelné než v případě druhém. Při váhování přechodů bylo potřeba váhovat i pravý okraj snímku, který při překreslování zprava doleva vytvářel výrazné artefakty.



Obrázek 4.6: Váhování 5 pixelového okraje, přechody mezi spojovanými snímky jsou zřetelnější.

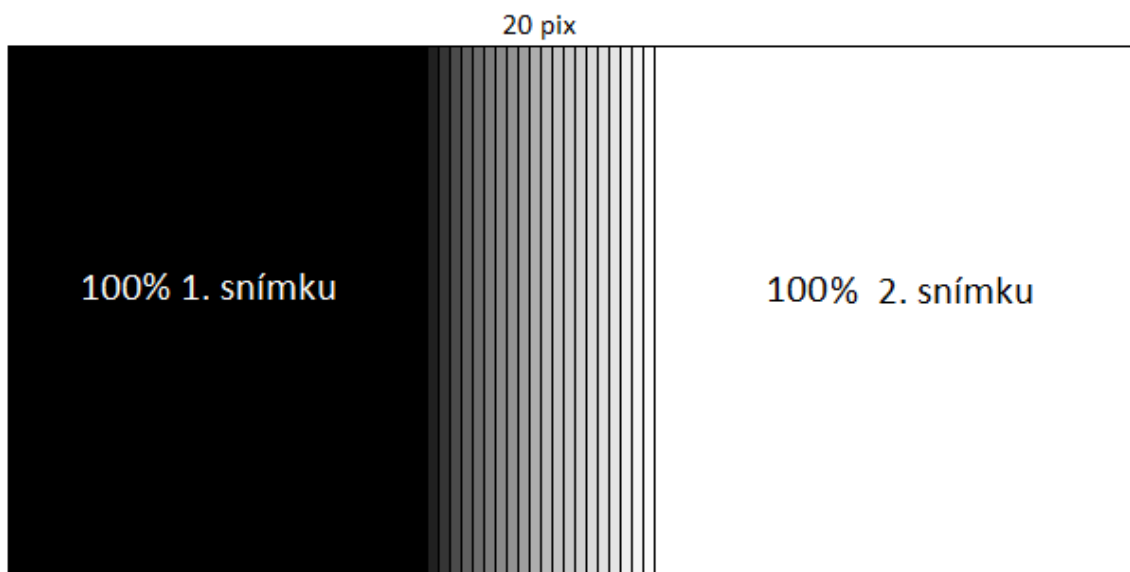




Obrázek 4.7: Váhování 20 pixelového okraje, přechod již není tak zřetelný.

Šířka váhovaného okraje nesmí být ani moc úzká (příklad 5 pixelů) ani moc široká. Experimentálně byla stanovena vhodná šířka okraje na 20 pixelů, protože poskytovala subjektivně nejlepší výsledky. I při měnící se intenzitě světla mezi jednotlivými snímky poskytuje váhovaný 20 pixelový přechod vhodný výsledek.

Váhovaný okraj je kombinací obou spojovaných snímků. První sloupec pixelů obsahuje 95% hodnoty daného pixelu z prvního snímku a 5% z druhého snímku, další sloupec je kombinací 90% z prvního a 10% z druhého, obdobně poslední sloupec pixelů obsahuje 0% z prvního snímku a 100% ze snímku druhého. Toto váhování při nastaveném okraji 20 pixelů je schematicky naznačeno na obrázku 4.8.

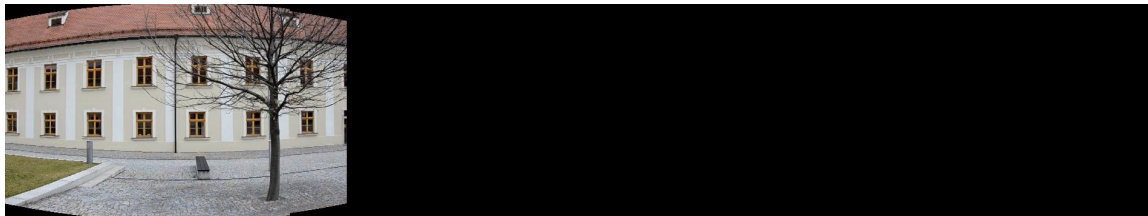


Obrázek 4.8: Princip váhování 20 pixelového okraje při spojování dvou snímků.

## 4.6 Vytvoření videa

V prvním běhu kamery mají jednotlivé spojené snímky rozdílnou velikost. Abychom z nich mohli vytvořit video, musíme převést tyto snímky na jednotnou velikost, jakou mají snímky v dalších bězích.

Tento převod implementuje funkce ve `videoPreprocessing.m`, která spočítá velikost posledního ze snímků a rozšíří menší snímky na tuto velikost. Jeden z prvních snímků vidíme na obrázku 4.9.



Obrázek 4.9: První snímek běhu kamery rozšířen na finální velikost.

Z těchto snímků se dále vytváří video typu *Audio Video Interleave* (avi), což je realizováno v `makeAvivideo.m`. Výsledné video je pojmenováno jako `panorama.avi` a je uloženo do složky, která je při spuštění vytvořena a pojmenována uživatelem. V této složce jsou dále uloženy i rozšířené snímky, které byly využity pro tvorbu videa.

Výsledná videa jsou v nekomprimované formě. Komprese byla provedena následně v programu Movie Maker na formát MPEG-4/H.264, aby bylo možné všechna ukázková videa uložit na CD.

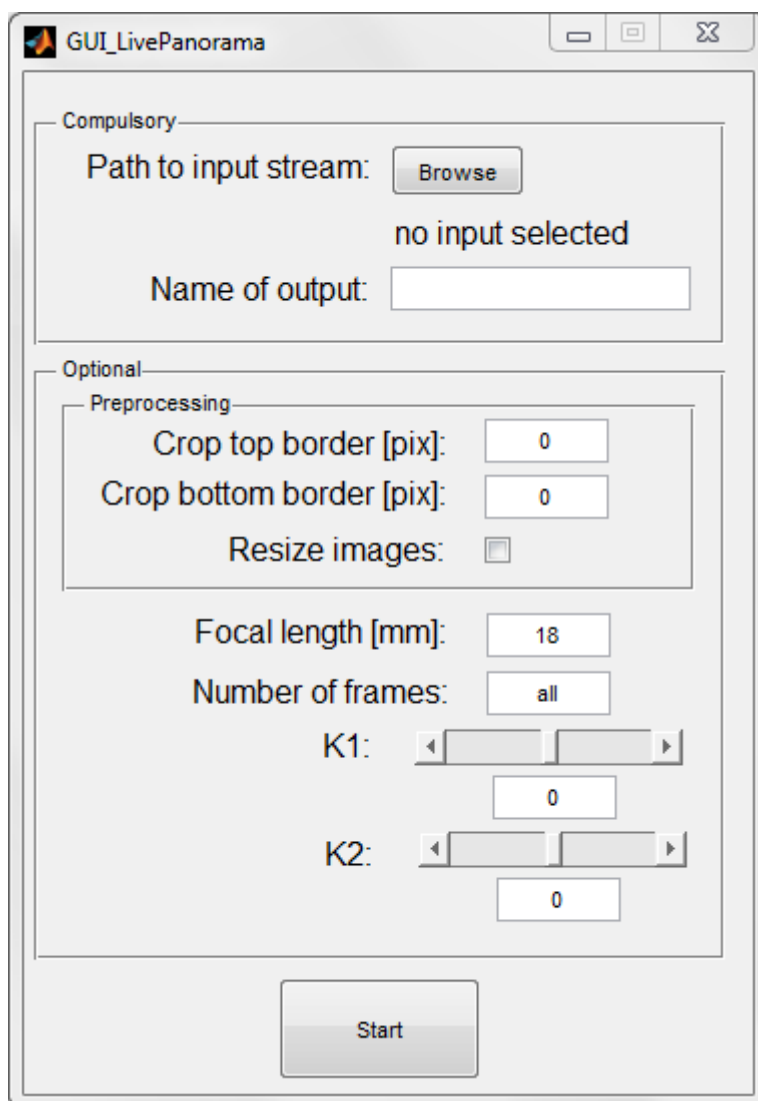
## 4.7 Uživatelské rozhraní

Pro lepší ovládání a propojení jednotlivých funkcí bylo vytvořeno také uživatelské rozhraní, které vidíme na obrázku 4.10.

Rozhraní je rozděleno do dvou hlavních částí, a to na část povinnou (*Compulsory*) a volitelnou (*Optional*). Je nutné nejprve vybrat vstupní video tlačítkem Procházet (*Browse*). Po vybrání se název tohoto videa zobrazí pod tímto tlačítkem. Do pole jméno výstupu (*Name of output*) napíšeme název složky, ve které bude uložen zpracovaný výsledek.

Ve volitelné části si můžeme nastavit další informace. V případě, že video obsahuje popisky nebo časová razítka a chceme je odstranit, nastavíme požadovaný ořez v pixelech v polích (*Crop top border*), či (*Crop bottom border*), podle toho, zda se popisky nacházejí v horní či spodní části videa. Dále zde nastavujeme, zda chceme zmenšit snímky (*Resize images*), ohniskovou vzdálenost v *mm*, při které bylo video natočeno (*Focal length*), počet zpracovaných snímků (*Number of frames*) a velikost parametrů  $k_1$  a  $k_2$  v rozsahu  $-0,3; +0,3$ . Parametry  $k_1$  a  $k_2$  mohou být nastaveny buď na posuvníku nebo vyplněním hodnoty do příslušného políčka.

Po spuštění tlačítkem *Start* začne tvorba živého panoramatu ze vstupního videa a o průběžích jednotlivých kroků je uživatel informován indikátory průběhu.



Obrázek 4.10: Náhled uživatelského rozhraní.

# Kapitola 5

## Experimentální výsledky

Pro testovací účely byla natočena sada videí různých scén a časových rozsahů, jak je shrnuto v tabulce 5.1. Všechna tato vstupní videa i jejich zpracovaná forma jsou dále přiložena na CD.

Tabulka 5.1: Testovací sady videí.

Popis	Soubor	Čas [s]
Klidná ulice	ulice.mov	77
Křižovatka	auta.mov	45
Mánesovy koleje	koleje.mov	71
Nádvoří FIT	nadvori.mov	333
Palackého vrch 1	palecak.mov	81
Palackého vrch 2	palecak2.mov	86
Petrov rozhled	petrov.mov	233
Špilberk	spilberk.mov	282
Petrov	petrov.mov	488
Špilberk	spilberk.mov	488
Muzeum Tatra	tatra.mov	59
Štramberská Trúba	truba.mov	91

### 5.1 Tvorba testovacích sad

Testovací sady byly natočeny digitální zrcadlovkou Nikon D5100 (s velikostí snímáče  $23,6 \times 15,6 \text{ mm}$ ) s objektivem NIKKOR 18 – 105 mm F3,5-5,6 ED a s využitím video stativu Manfrotto 745Xb s otočnou hlavou. Všechna videa mají rozlišení  $1280 \times 720$  s frekvencí 25 snímků za sekundu.

Při nasnímání vhodné testovací sady bylo potřeba při natáčení držet vodorovnou rovinu, k čemuž dobře posloužila integrovaná vodováha na stativu.

## 5.2 Testování

Při testování byly všechny snímky zmenšeny na rozměr  $640 \times 360$  a dále zpracovávány. U takto velkých snímků trvá přidání dalšího snímku do panoramatu přibližně 14 sekund (Intel Core i7-4770@3,4GHz, 16GB RAM, Windows 7 64-bit).

Většina videí byla natočena s ohniskovou vzdáleností 18 mm, kterou bylo nutno dále přepočítat dle vztahu 3.5. V případě zpracovávaného snímku velikosti  $640 \times 360$ , je výpočet ohniskové vzdálenosti v pixelech následující:

$$f[\text{pix}] = \frac{18}{23,6} * 640, \quad (5.1)$$

což odpovídá ohniskové vzdálenosti 488 pixelů, jak je uvedeno u většiny videí níže v tabulce. Obdobným způsobem byly dopočítány i ostatní hodnoty ohniskových vzdáleností.

Tabulka 5.2: Testovací sady videí.

Soubor	f [pix]	$k_1$	$k_2$
ulice.mov	488	-0,10	0,00
krizovatka.mov	949	-0,20	0,30
koleje.mov	488	-0,20	0,20
nadvori.mov	488	-0,20	0,20
palecak.mov	949	-0,20	0,20
palecak2.mov	2847	-0,20	0,20
petrov.mov	488	-0,20	0,20
spilberk.mov	949	-0,10	0,10
tatra.mov	678	-0,20	0,20
truba.mov	949	-0,20	0,20

Parametry  $k_1$  a  $k_2$  uvedeny výše v tabulce 5.2 byly nalezeny pomocí prostého prohledávání a následného subjektivního vyhodnocení.

## 5.3 Výsledná panoramata

Na obrázcích 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10 a 5.11 vidíme výsledná panoramata s uvedenou délkou výsledného videa. Všechna videa jsou v komprimované formě dostupná na CD, které je přiloženo k této práci.

Ze všech testovacích sad vznikla kvalitní živá panoramata. V některých případech však překreslování není úplně dokonalé a v aktuálním běhu kamery se může stát, že na scéně zůstane část překreslovaného pohybuujícího se objektu, jak vidíme na



obrázku 5.1. V následujícím běhu se však tato scéna opět překreslí. Vznik takovýchto artefaktů se dá částečně odstranit vhodným nastavením velikosti kroku.



Obrázek 5.1: Artefakt při překreslování scény.



Obrázek 5.2: Klidná ulice – 77s.

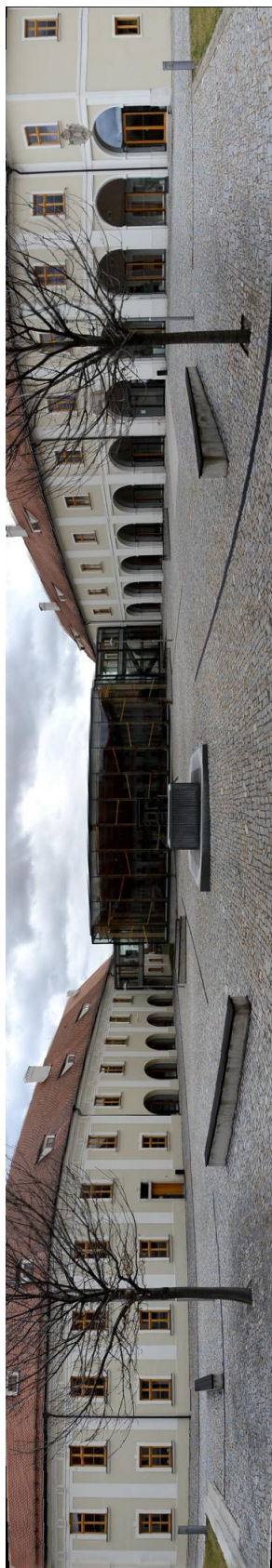


Obrázek 5.3: Křižovatka – 18s.

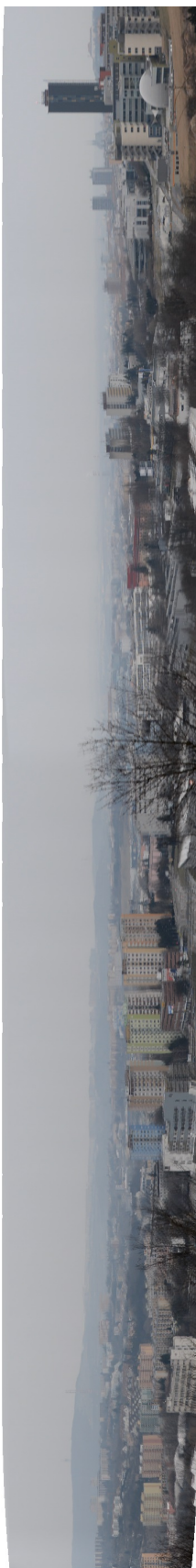


Obrázek 5.4: Mánesovy koleje – 14s.





Obrázek 5.5: Nádvoří FIT – 66s.

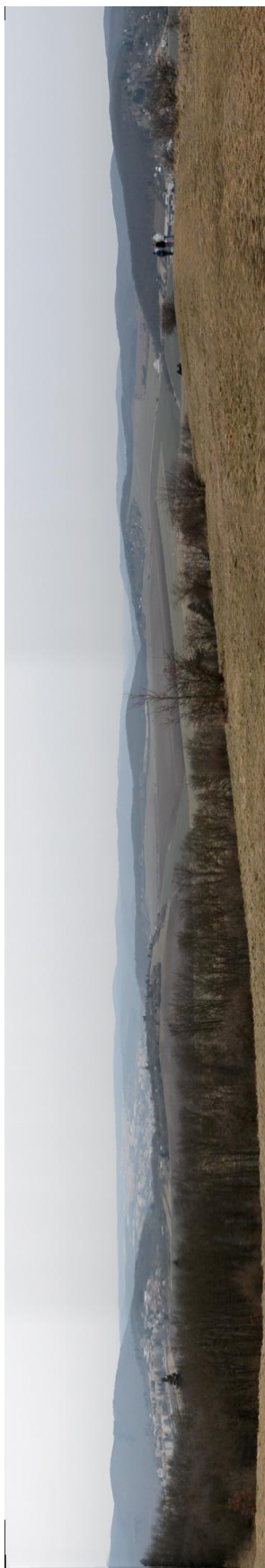


Obrázek 5.6: Palackého vrch – 17s.



Obrázek 5.7: Petrov – 46s.





Obrázek 5.8: Palackého vrch 2 – 16s.



Obrázek 5.9: Výhled ze Špilberku – 93s.



Obrázek 5.10: Štrambercká trůba – 18s.



Obrázek 5.11: Muzeum TATRA – 11s.

## 5.4 Testování na reálných datech

Testování proběhlo i na reálných datech z rozhleden a jiných dostupných rotujících kamer. Algoritmus dosahoval dobrých výsledků u kamer, které snímají scénu konzistentně, nezasekávají se a nemění za svého běhu přiblížení jednotlivých kusů scény.

Na obrázku 5.12 vidíme výsledné panorama z nahrávky *cam3.avi*, kdy v prvním běhu kamera rotovala plynule a bez zaseknutí. Avšak při návratu do výchozí pozice se vrací nárazově, takže se při cestě zpátky panorama nestihne překreslit. Z tohoto důvodu algoritmus panorama překresluje na špatné pozici, jak vidíme na obrázku 5.13. Vzhledem k časové náročnosti hledání klíčových bodů ve snímcích jsou hledány pouze mezi dvěma sousedními snímky a ne v celém panoramatu (časová náročnost by výrazně stoupla), proto při nekonzistentních nahrávkách algoritmus poskytuje ne příliš kvalitní výsledky.

Pokud se však kamera zasekává ve směru svého běhu a nezmění skokově svou pozici, algoritmus několikrát překreslí snímek na dané pozici a dále pokračuje. Toto chování se vyskytovalo například u kamery na rozhledně Klínovec a je znázorněno na obrázku 5.14, kdy červenými okýnkami jsou vyznačeny pozice zaseknutí.

V některých případech je hledání klíčových bodů komplikováno také popisky a časovými razítky na pořízených videích. V případě vstupního videa *cam3.avi* byly popisky pouze na dolním okraji snímků, takže nijak neovlivnily výsledné spojování snímků. Pokud však byly popisky výrazné a vyskytovalo se jich ve videu více, jako v případě vstupního videa *cam2.avi*, pak je spojování snímků velmi ovlivněno, jak vidíme na obrázku 5.15. V případě *cam2.avi* byla většina klíčových bodů nalezena v časovém razítku vpravo nahoře a v popisích v dolních rozích. Shoda tedy byla vyhodnocena dle klíčových bodů popisků vždy stejně a snímky se tedy překreslovaly stále na stejné pozici.

Do funkce na extrakci snímků byla tedy přidána možnost ořezání okrajů, a tak odstranění popisků. Při ořezání popisků byly klíčové body opět nalezeny v pořádku a výsledné panorama je znázorněno na obrázku 5.16.





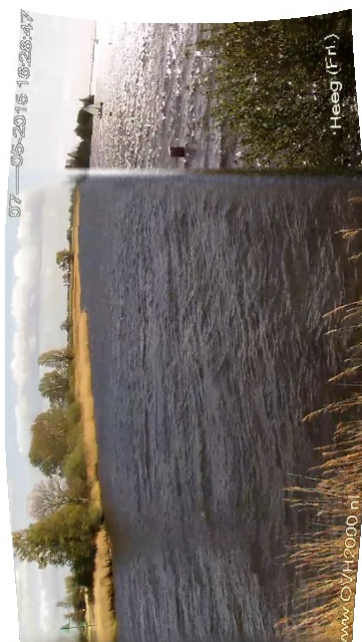
Obrázek 5.12: První běh kamery, průběh konzistentní rychlostí bez zasekávání.



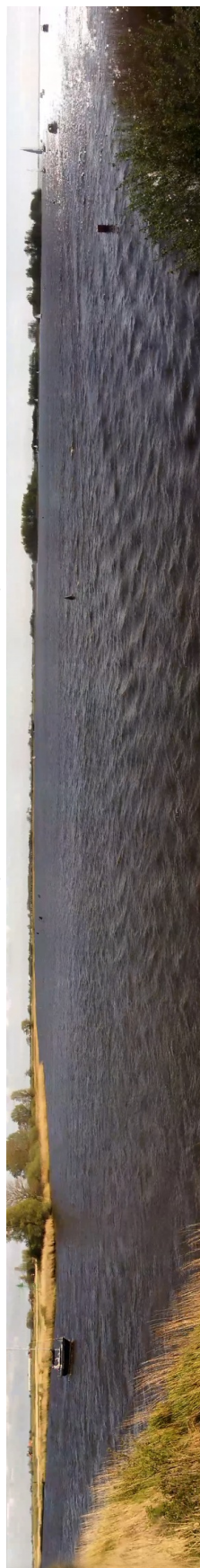
Obrázek 5.13: Zpětný nárazový běh kamery.



Obrázek 5.14: Pohled z rozhledny Klínovec při neskokovém zaseknutí kamery ve směru běhu. Červeně jsou vyznačeny pozice zaseknutí a několiknásobné překreslení snímků na těchto pozicích.



Obrázek 5.15: Výsledek ovlivněn popisky videa.



Obrázek 5.16: Výsledek po odstranění popisků.

# Kapitola 6

## Závěr

V rámci této bakalářské práce byla implementována řada funkcí pro realizaci živých panoramat. Byly využity metody založené na známých poznacích pro tvorbu statických panoramat, především na algoritmech SIFT a RANSAC, které byly blíže rozebrány v kapitole 3.

Naše řešení poskytuje inovativní možnost využití mediálních dat (např. prezentace obcí, informace ze sjezdovek a rozhleden, zabezpečení objektů apod.), hlavní výhodou je zobrazení celé snímané scény a nejen části, která je zrovna zobrazena rotující kamerou.

Samotná implementace se skládá z několika na sebe navazujících kroků, které byly blíže rozebrány v kapitole 4, a to extrakce snímků, jejich převod do cylindrických souřadnic, nalezení klíčových bodů, výpočet homografie, spojení snímků a vyhlazení přechodů, dále plynulé překreslování a vytvoření videa.

Pro převod snímků do cylindrických souřadnic je potřeba hodnoty ohniskové vzdálenosti při které je video snímáno a parametrů  $k_1$  a  $k_2$ . Ohniskovou vzdálenost lze vyčíst z EXIF informací snímků, vypočítat či odhadnout a parametry  $k_1$ ,  $k_2$  jsou optimalizovány prostým prohledáváním, jak je blíže popsáno v podkapitolách 3.1.2 a 4.2. U většiny testovacích videí byla prostým prohledáváním vyhodnocena jako nejlepší kombinace parametrů  $k_1 = -0,2$  a  $k_2 = 0,2$ .

Algoritmus funguje spolehlivě v případě, kdy kamera snímá scénu konzistentně nebo pokud se snímání zasekává postupně ve směru běhu. Pokud však kamera scénu snímá nekonzistentně nebo se zasekává skokově na různých místech scény, či přibližuje jednotlivé kusy scény, pak není výsledné panorama překresleno zcela správně.

Jak ukázaly experimentální výsledky, metoda poskytuje kvalitní živá panoramata, avšak v některých ohledech má prostor pro zlepšení.

V rámci zadání byl kladen důraz především na ověření proveditelnosti konceptu živých panoramat. V rámci budoucího využití a rozšiřování by ale měla být apli-

kace optimalizována, aby zvládala tvorbu těchto panoramat v reálném čase. To může být realizováno například přepisem programu do C++ kódu, který má potenciál pro zrychlení aplikace (oproti stávající implementaci v Matlabu). Rovněž by mohlo dojít k paralelizaci výpočtu nebo využití GPU akcelerace například v podobě knihovny Jacket. Dalším možným zlepšením by bylo zařazování snímků kamkoliv do celého panoramatu namísto stávajícího přiřazení pouze k sousedním snímkům (to silně souvisí s výpočetní náročností zmíněnou výše). V neposlední řadě by v pokračování projektu bylo potřeba důkladné otestování nejen na kamerových vzorcích, ale také na videích z mobilních telefonů a dalších zařízení.



# Literatura

- [1] *Radial Distortion Correction*. online. URL [http://www.uni-koeln.de/~al001/radcor\\_files/hs100.htm](http://www.uni-koeln.de/~al001/radcor_files/hs100.htm).
- [2] *Vytváření panoramatických obrazů pomocí Photomerge*. online. URL <https://helpx.adobe.com/cz/photoshop/using/create-panoramic-images-photomerge.html>.
- [3] *Autopano Giga, Panorama Editor, Projections*. online, 2004–2014. URL [http://www.kolor.com/wiki-en/action/view/Autopano\\_Giga\\_-\\_Panorama\\_Editor\\_-\\_Projections](http://www.kolor.com/wiki-en/action/view/Autopano_Giga_-_Panorama_Editor_-_Projections).
- [4] *Cubic Projection*. online, 2007. URL [http://wiki.panotools.org/Cubic\\_Projection](http://wiki.panotools.org/Cubic_Projection).
- [5] S. E. Chen. *QuickTime VR: An Image-based Approach to Virtual Environment Navigation*. In Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95, pages 29–38, New York, NY, USA, 1995. ACM. ISBN 0-89791-701-4. URL <http://doi.acm.org/10.1145/218380.218395>.
- [6] K. G. Derpanis. *Overview of the RANSAC Algorithm*. online, May 2010. URL [http://www.cse.yorku.ca/~kosta/CompVis\\_Notes/ransac.pdf](http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf).
- [7] J. Dolz. *How to solve the Image Distortion Problem*. online, 2012. URL <http://www.arlab.com/blog/how-to-solve-the-image-distortion-problem-code-snippet/>.
- [8] D. G. Lowe. *Object Recognition from Local Scale-Invariant Features*. In Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2, ICCV '99, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8. URL <http://dl.acm.org/citation.cfm?id=850924.851523>.

- [9] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, 60(2):91–110, November 2004. ISSN 0920-5691.
- [10] S. Mchugh. *Panoramic Image Projections*. online. URL <http://www.cambridgeincolour.com/tutorials/image-projections.htm>.
- [11] K. Mikolajczyk and C. Schmid. *A performance evaluation of local descriptors*. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(10):1615–1630, 2005. URL <http://lear.inrialpes.fr/pubs/2005/MS05>.
- [12] D.L. Milgram. *Computer Methods for Creating Photomosaics*. Computers, IEEE Transactions on, C-24(11):1113–1119, Nov 1975. ISSN 0018-9340.
- [13] J. Smith. *History of Panoramic Photography*. online, 2010. [http://www.onassignmentstudios.com/panoramic\\_history.html](http://www.onassignmentstudios.com/panoramic_history.html).
- [14] N. Snavely. *Estimating the focal length of a photo from EXIF tags*. online, January 2008. URL <http://phototour.cs.washington.edu/focal.html>. <http://phototour.cs.washington.edu/focal.html>.
- [15] M. Srna. *Panoramatická fotografie snadno a rychle*. online, 2010. URL <http://www.digimanie.cz/panoramaticka-fotografie-snadno-a-rychle/3441>.
- [16] R. Szeliski. *Image Stitching*. online. URL [http://staff.ustc.edu.cn/~xjchen99/teaching/ImageUnderstanding/Chapter9\\_Stitching.pdf](http://staff.ustc.edu.cn/~xjchen99/teaching/ImageUnderstanding/Chapter9_Stitching.pdf).
- [17] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 1848829345, 9781848829343.
- [18] R. Szeliski and H. Shum. *Creating Full View Panoramic Image Mosaics and Environment Maps*. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pages 251–258, 1997. ISBN 0-89791-896-7. URL <http://dx.doi.org/10.1145/258734.258861>.
- [19] R. Thomas. *Landscape Photography*. blogpost (english). <http://photoblogstop.com/photography/landscape-photography>.
- [20] M. Vanvolsem. *The art of strip photography : making still images with a moving camera*. Leuven University Press, Leuven, Belgium, 2011. ISBN 9789058678409.
- [21] Y. Xiong and K. Turkowski. *Registration, Calibration and Blending in Creating High Quality Panoramas*. In Fourth IEEE Workshop on Applications of Computer Vision, 1998.

# Příloha A

## Obsah CD

Součástí práce je CD, které obsahuje tyto položky:

- soubor `README.txt` popisující obsah CD
- adresář `Panorama` s implementovanými funkcemi pro tvorbu živých panoramat popsaných v kapitole 4, obsahující položky:
  - skript `cylCoord.m` pro převod snímků do cylindrických souřadnic.
  - skript `focalLength.m` pro přepočítání ohniskové vzdálenosti.
  - skript `GUI_LivePanorama.m` GUI pro ovládání a nastavení příslušných hodnot.
  - skript `isEmpty.m` pro kontrolu obsahu snímku.
  - skript `livePanorama.m` pro výpočet jednotlivých kroků tvorby živých panoramat.
  - skript `Lowe_showMatches.m` pro vykreslení klíčových bodů.
  - skript `Lowe_sift.m` pro hledání klíčových bodů ve snímcích.
  - aplikaci `Lowe_siftWin32.exe` pro hledání klíčových bodů ve snímcích.
  - skript `makeAviVideo.m` pro tvorbu výsledného videa.
  - skript `matcher.m` pro nalezení nejlepších odpovídajících si klíčových bodů z dvojic snímků.
  - jedno z testovacích videí `nadvori.mov`.
  - skript `optimization.m` pro nalezení nejvhodnější kombinace parametrů  $k_1$  a  $k_2$ .
  - skript `planarProjection.m` ukázka planární projekce.

- uživatelská příručka `prirucka.pdf` popisující obsluhu programu.
- skript `ransac.m` pro nalezení vzájemné pozice klíčových bodů.
- skript `stitch.m` pro spojení jednotlivých snímků.
- skript `videoParsing.m` pro extrakci snímků ze vstupu.
- skript `videoPreprocessing.m` pro rozšíření snímků na jednotnou velikost.
- adresář `SIFT_license` obsahující licenci k SIFT algoritmu.
- adresář `LaTeX` se zdrojovými texty a obrázky pro textovou část bakalářské práce.
- samotná textová část bakalářské práce `xjaluv02.pdf`.
- plakát `xjaluv02Poster.pdf`.
- video `xjaluv02Video.mp4`.
- adresář `Testovani` s vstupními a výstupními videi popsány v kapitole 5, obsahující položky (čísla v názvu jsou kombinací nastavené ohniskové vzdálenosti, parametrů  $k_1$  a  $k_2$ ) :
  - adresář `Camera2_100_-0.2+0.2`.
  - adresář `Camera3_105_-0.1+0.1`.
  - adresář `KlidnaUlice_18_-0.1+0`.
  - adresář `Klinovec_800_-0.1+0`.
  - adresář `Křižovatka_35_-0.20+0.30`.
  - adresář `ManesovyKoleje_18_-0.2+0.2`.
  - adresář `Nádvoří FIT_18_-0.20+0.20`.
  - adresář `Palecak_35_-02+02`.
  - adresář `Palecak_105_-0.2+0.2`.
  - adresář `PetrovRozhled_18_-0.2+0.2`.
  - adresář `Špilberk_35_-0.1-0.1`.
  - adresář `Tatra_25_-0.2+0.2`.
  - adresář `Truba_35_-0.2+0.2`.