

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

AUTOMATICKÉ ZPRACOVÁNÍ ZÁZNAMU Z VOZIDLOVÉ KAMERY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIK KLOS

BRNO 2012



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

AUTOMATICKÉ ZPRACOVÁNÍ ZÁZNAMU Z VOZIDLOVÉ KAMERY

AUTOMATIC PROCESSING OF RECORDINGS FROM IN-CAR CAMERA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIK KLOS

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MICHAL ŠPANĚL, Ph.D.

BRNO 2012

Abstrakt

Tato práce se zabývá automatickým zpracováním záznamu z vozidlové kamery, který je zde využit pro odhadování vzdáleností v natočené scéně. K dosažení tohoto cíle jsou využity metody pro zpracování obrazu, geometrii prostorové scény a 3D rekonstrukci, jejichž konečným výstupem je částečný model. K tomuto účelu je využit nástroj Bundler. Výsledky rekonstrukce jsou vizualizovány v aplikaci, která poté umožňuje uživateli odhadovat vzdálenost objektů od kamery nebo jejich vzdálenosti vůči sobě navzájem.

Abstract

This thesis deals with the automatic processing of recordings from in-car camera, which is used for estimating distances in a scene filmed. To achieve this goal, it uses the methods of image processing, spatial geometry of the scene and 3D reconstruction, whose final output is a partial model. For this aim, the Bundler tool is used. Results of reconstruction are visualized in the application, which then allows to estimate the distance of objects from the camera or the distance between each other.

Klíčová slova

odhad vzdálenosti, vozidlová kamera, 3D rekonstrukce, pinhole camera model, epipolární geometrie, fundamentální matice, SIFT, SfM, Bundler

Keywords

estimate the distance, in-car camera, 3D reconstruction, pinhole camera model, epipolar geometry, fundamental matrix, SIFT, SfM, Bundler

Citace

Dominik Klos: Automatické zpracování záznamu z vozidlové kamery, bakalářská práce, Brno, FIT VUT v Brně, 2012

Automatické zpracování záznamu z vozidlové kamery

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Španěla, Ph.D.

.....
Dominik Klos
16. května 2012

Poděkování

Rád bych poděkovat především panu Ing. Michalu Španělovi, Ph.D. za jeho vedení a rady, které mi poskytoval během tvorby mé práce. Moje poděkování patří také panu Doc. Ing. Adamovi Heroutovi, Ph.D., který mi poskytl testovací data a zapůjčil kameru ke kalibraci.

© Dominik Klos, 2012.

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	Teoretické základy	4
2.1	Pinhole camera model	4
2.2	Epipolární geometrie	5
2.3	Kalibrace kamery	7
2.4	3D rekonstrukce	9
3	Analýza dopravních nehod	10
3.1	Vozidlová kamera	10
3.2	Software využívaný v praxi	11
4	Nátroj Bundler	13
4.1	Photo Tourism	14
4.2	Building Rome in a Day	15
5	Návrh aplikace pro zpracování záznamu z vozidlové kamery	17
5.1	Kalibrace kamery	18
5.2	Vybrání snímků pro 3D rekonstrukci	19
5.3	SIFT – hledání význačných bodů	19
5.4	Hledání korespondujících bodů	19
5.5	Bundler – rekonstrukce scény	20
5.6	Vizualizace výsledků rekonstrukce	20
5.7	Kalibrace scény a výpočet vzdálenosti	21
5.8	Návrh grafického uživatelského rozhraní	21
5.9	Testovací data	22
6	Implementace	23
6.1	Kalibrace kamery	23
6.2	Přehrávání videa	24
6.3	Spuštění externích aplikací	24
6.4	Zpracování výstupu z Bundleru	25
6.5	Vizualizace výsledku rekonstrukce	26
7	Výsledky	27
7.1	Testovací měření	27
7.2	Shrnutí výsledků	30
7.3	Možná vylepšení	32

8 Závěr	33
A Obsah DVD	36
B Plakát	37

Kapitola 1

Úvod

Obor počítačové grafiky má v dnešní době široké využití. Toto využití začíná jednoduchým zpracováním fotografií v grafických editorech, přes využití počítačového vidění pro řízení autonomních robotů, až po náročnější analýzu medicínských dat. Přínosy počítačové grafiky jsou neoddiskutovatelné. Z tohoto důvodu jsem si i já zvolil téma svojí bakalářské práce, jejíž výsledek by mohl mít potenciální využití v praxi.

Tématem mé bakalářské práce je *Automatické zpracování záznamu z vozidlové kamery*. Rozhodl jsem se v ní věnovat odhadování vzdáleností mezi objekty objevujícími se v těchto záznamech. Mým cílem je vytvořit aplikaci, která by toto umožňovala.

Znalost vzdáleností mezi objekty ve scéně je možné využít například při analýze dopravních nehod a dalších krizových situací, které mohou nastat v silničním provozu. Taktéž se nabízí využití například v autoškolách, kde lektor může studentovi ukazovat na záznamu chyby, kterých se student za jízdy dopustil.

I když se práce zaměřuje na záznam, který byl pořízen během jízdy vozidlem, je možné výslednou aplikaci použít i pro zpracování videa, které bylo natočeno uživatelem ručně.

Jako prostředek pro odhadování vzdáleností slouží 3D model scény. Model scény je výsledkem částečné 3D rekonstrukce. Částečné proto, že jejím výsledkem není kompletně zrekonstruovaná scéna, ale pouze body, o kterých je rozhodnuto, že jsou z hlediska grafiky významné. Vzdálenosti, které je možné odhadovat, jsou právě mezi těmito body.

Rekonstrukce scény není ve výsledné aplikaci přímo implementována, ale je využito externího nástroje *Bundler*, jehož výstup je zpracován, vizualizován a dále využit pro měření ve scéně.

Na následujících stránkách bude popsána teorie (kapitola 2), jejíž nastudování bylo nutné pro splnění cíle. Dále zde budou uvedena některá fakta z praxe užívání vozidlových kamer a podobná existující softwarová řešení, které slouží pro analýzu dopravních nehod (kapitola 3). V kapitole 4 bude představen nástroj *Bundler* a budou uvedeny některé projekty, ve kterých byl využit. Následovat bude kapitola 5, ve které bude popsán návrh aplikace. Poté kapitola 6 popíše konkrétní implementaci řešení. A na závěr kapitola 7, ve které budou shrnuty dosažené výsledky.

Kapitola 2

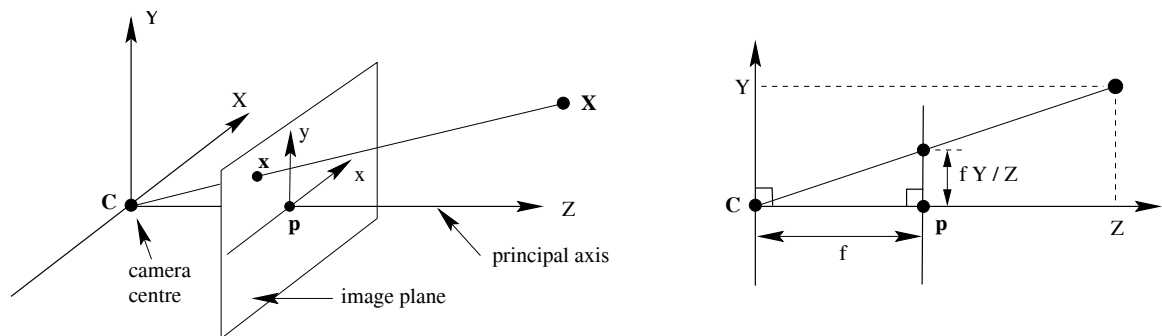
Teoretické základy

Tato kapitola popisuje teoretické základy, ze kterých práce vychází a které jsou buď přímo nebo nepřímo využity ve výsledném řešení.

2.1 Pinhole camera model

Nebo-li česky dírková kamera je model kamery s perspektivní projekcí, který zobrazuje body na projekční rovině pomocí přímek procházejících středovým bodem. Kamera mapuje body mezi 3D a 2D projekční rovinou. V tomto promítání není zachována rovnoběžnost hran a velikost objektu zobrazovaných na projekční rovinu je nepřímo úměrná vzdálenosti objektu od kamery. Čím je objekt blíže projekční rovině, tím je jeho obraz větší a naopak[15].

V praxi se nejčastěji používá právě tento model, protože se svými vlastnostmi nejvíce podobá lidskému vidění.



Obrázek 2.1: Pinhole camera, převzato z [14]

V modelu dírkové kamery je bod ve 3D se souřadnicemi $X = (X, Y, Z)^T$ zobrazován do obrazové roviny pomocí přímky, která spojuje bod X a *optický střed* C . V místě kde tato přímka protíná obrazovou rovinou se nachází obraz bodu X . Toto znázorňuje obrázek 2.1. Střed projekce je zde nazýván *optickým středem* (*camera center*). Přímka spojující optický střed a obrazovou rovinu se nazývá *hlavní osa* (*principal axis*). Bod, ve kterém protíná *hlavní osa* obrazovou plochu se nazývá *hlavní bod* (*principal point*). Vzdálenost mezi *optickým středem* a *hlavní bodem* se nazývá *ohnisková vzdálenost* a značí se f . Střed kamery je zde umístěn do počátku souřadného systému. Pomocí podobnosti trojúhelníků lze

snadno dopočítat zobrazení bodu $(X, Y, Z)^T$ ve 3D do bodu $(fX/Z, fY/Z, f)^T$ v obrazové rovině. Pro souřadnice bodů v obraze platí vztah 2.1[14].

$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T \quad (2.1)$$

Tento vztah popisuje převod z trojrozměrného souřadného systému do souřadného systému obrazu. Mapuje body z trojrozměrného Eukleidovského prostoru do dvourozměrného Eukleidovského prostoru.

Pokud je scéna a obraz bodu reprezentován pomocí homogenních vektorů, lze perspektivní projekci jednoduše vyjádřit jako lineární zobrazení mezi jejich homogenními souřadnicemi. Toto můžeme zapsat pomocí součinu matice a vektorů, jak ukazuje vztah 2.2.

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & 0 \\ & f & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.2)$$

Výše uvedená matice lze také vyjádřit jako $\text{diag}(f, f, f, 1)[I \mid 0]$, kde $\text{diag}(f, f, f, 1)$ je diagonální matice a $[I \mid 0]$ představuje matici rozdělenou na část o velikosti 3×3 a sloupcový vektor obsahující nuly. Tuto matici nazýváme *projekční maticí kamery* a značíme ji P .

$$P = \text{diag}(f, f, f, 1)[I \mid 0] \quad (2.3)$$

Pokud máme bod X ve 3D vyjádřený homogenním 4-vektorem $(X, Y, Z, 1)^T$, jeho obraz x vyjádřený homogenním 3-vektorem a P jako 3×4 homogenní projekční maticí kamery, potom platí vztah 2.4

$$x = PX \quad (2.4)$$

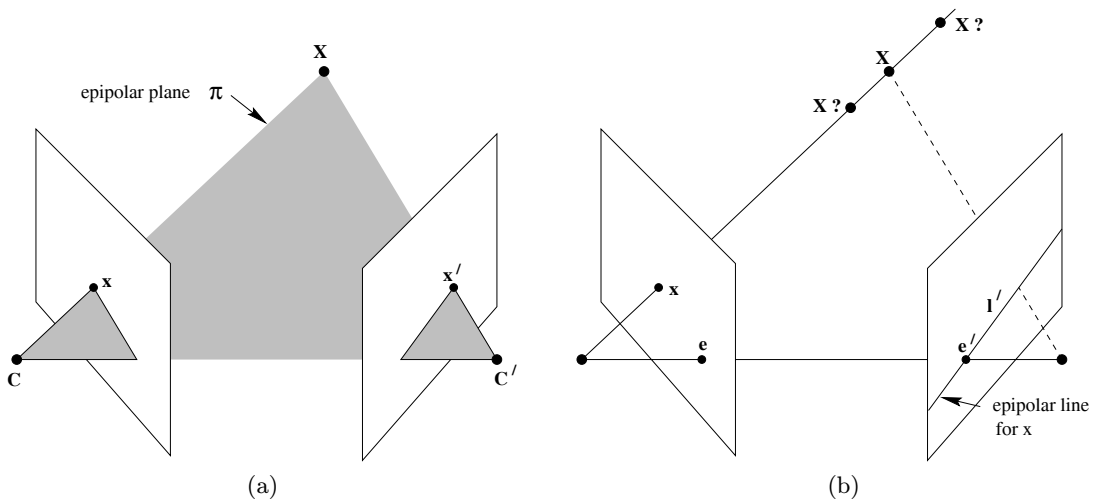
2.2 Epipolární geometrie

Epipolární geometrie je vnitřní projektivní geometrie mezi dvěma pohledy. Je nezávislá na struktuře scény, závisí pouze na vnitřních parametrech kamer a jejich relativní pozici [21]. Epipolární geometrie pomáhá usnadnit problém hledání korespondujících bodů mezi dvěma pohledy.

Na obrázku 2.2a je znázorněn bod X ve 3D a jeho dva obrazy x a x' . Tyto body společně s optickými středy C a C' leží v jedné rovině π . V této rovině leží také přímky zpětné projekce vedoucí z bodů x a x' do bodu X .

Nyní předpokládejme, že známe pouze bod x a chceme najít s ním korespondující bod x' . Rovina π je nyní definována spojnicí optických středů C , C' a projekční přímkou procházející bodem x . Z obrázku 2.2a již víme, že projekční přímka pro dosud neznámý bod x' musí ležet v rovině π . Za předpokladu, že hledaný x' musí ležet na projekční přímce, musí bod x' ležet i v rovině π . Z toho vyplývá, že bod x' se musí nacházet na průsečíku roviny π a druhé obrazové roviny. Tímto průsečíkem je přímka l' , která je znázorněna na obrázku 2.2b. Je to rovněž současně obraz projekční přímky procházející bodem x . Tento průsečík se na se také nazývá *epipolární přímkou*.

Díky výše uvedeným znalostem můžeme bod x' hledat pouze na epipolární přímce a nemusíme prohledávat celou obrazovou rovinu. K epipolární geometrii se vážou následující termíny [21]:



Obrázek 2.2: Epipolární geometrie, převzato z [14]

- *Epipól* je průsečík obrazové roviny a spojnice optických středů kamer. Současně je to také obraz optického středu druhé kamery. Na obrázku 2.2b jsou epipóly označeny jako e a e' .
- *Epipolární rovina* je rovina obsahující spojnici optických středů (bázi).
- *Epipolární přímka* je průsečnice epipolární roviny a obrazové roviny. Všechny epipolární přímky se protínají v epipólu. Na obrázku 2.2b je epipolární přímka označena jako l' .

Algebraickým vyjádřením epipolární geometrie je *fundamentální matice*. Tato matice představuje mapování mezi bodem v jednom obraze a jeho epipolární přímkou v obraze druhém. Když se podíváme na obrázek , vidíme, že pro každý bod x existuje korespondující epipolární přímka.

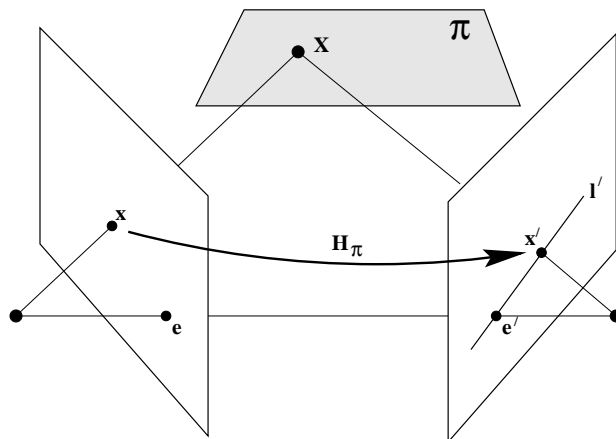
$$x \mapsto l' \tag{2.5}$$

Fundamentální matice

Nyní bude vysvětleno, jak lze fundamentální matici geometricky odvodit.

Toto odvození se skládá ze dvou kroků. V prvním kroku je bod x mapován na nějaký bod x' ležící na epipolární přímce v druhém obraze. V druhém kroku je získána epipolární přímka l' jako přímka procházející bodem x' a epipólem [14].

1. Jak vidíme na obrázku 2.3, máme v trojrozměrném prostoru zvolenou rovinu π , která neprochází optickými středy kamer. V místě, kde se protne projekční přímka bodu x s rovinou π se nachází bod X . Tento bod je promítnut do druhého obraze jako bod x' . Tento postup je znám jako přenos pomocí roviny π . Protože X leží na projekční přímce x , zobrazený bod x' musí ležet na epipolární přímce l' , která koresponduje s obrazem projekční přímky. Body x a x' jsou oba obrazy bodu X ve 3D ležícího v rovině π . Množina takových bodů x_i , v prvním obraze koresponduje s body x'_i



Obrázek 2.3: Odvození fundamentální matice, převzato z [14]

v obraze druhém jsou projektivně ekvivalentní, protože každý z nich je projektivně ekvivalentní k množině planárních bodů X_i . Existuje tedy 2D homografie H_π mapující každý x_i na x'_i .

2. Máme-li bod x' , epipolární přímku l' procházející bodem x' a epipól e' , můžeme napsat $l' = e' \times x' = [e']_\times X'^{-1}$. Protože x' může být zapsáno jako $x' = H_\pi x$, dostáváme vztah 2.6

$$l' = [e']_\times H_\pi X = Fx \quad (2.6)$$

kde definujeme fundamentální matici F podle vztahu 2.7.

$$F = [e']_\times H_\pi \quad (2.7)$$

Výše uvedené odvození vyžaduje rovinu π . Tato rovina ale není podmínkou pro existenci fundamentální matice. Rovina zde slouží pro zjednodušení mapování bodu z jednoho obrazu do druhého.

2.3 Kalibrace kamery

Kalibrace kamery v praxi znamená zjištění vnitřních a vnějších parametrů kamery. Toto se provádí například pomocí různých kalibračních vzorů.

Vnitřní parametry

Vnitřní parametry kamery jsou reprezentovány *kalibrační maticí* K . Kalibrační matice mapuje body z 3D souřadnic do pixelových souřadnic obrazu.

¹ $[e']_\times$ je zápis pro šikmo symetrickou matici, $[e']_\times = \begin{bmatrix} 0 & -e'_3 & e'_2 \\ e'_3 & 0 & -e'_1 \\ -e'_2 & e'_1 & 0 \end{bmatrix}$

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \quad (2.8)$$

Vyjádření matice K vidíme ve vztahu 2.8, ve kterém $\alpha_x = fm_x$ a $\alpha_y = fm_y$ reprezentují ohniskovou vzdálenost v pixelech, kde m_x a m_y jsou měřítka zvětšení v příslušných osách². s je parametr zkreslení, který určuje úhel mezi osami x a y . Často je tento parametr nulový. $x_0 = m_x p_x$ a $y_0 = m_y p_y$, kde p_x a p_y jsou souřadnice hlavního bodu, který nemusí být vždy ve středu obrazu.

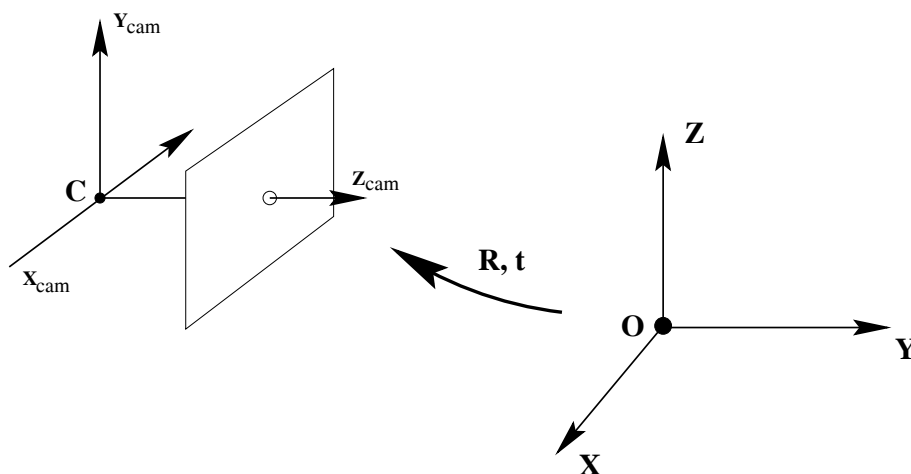
Vnitřní parametry jsou závislé pouze na vnitřní geometrii kamery, nejsou závislé na scéně. Kalibrační matice se proto může používat až do změny ohniskové vzdálenosti.

Vnější parametry

Vnější parametry kamery reprezentují pozici kamery ve scéně. Tato pozice je reprezentována pomocí *matice rotace* R a *vektoru posunutí* \vec{t} . Vnější parametry kamery mapují bod z trojrozměrného souřadného systému scény do trojrozměrného souřadného systému kamery, jak je zobrazeno na obrázku 2.4.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.9)$$

$$\vec{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2.10)$$



Obrázek 2.4: Rotace a posun kamery, převzato z [14]

Pokud chceme získat souřadnice optického středu C kamery ve scéně, použijeme vztah 2.11.

²V CCD kamerách nemusí být, oproti modelu dírkové kamery, pixely čtvercového tvaru. Toto zkreslení reprezentují právě m_x a m_y .

$$C = -R^T \vec{t} \quad (2.11)$$

Projekční matice

Projekční matice přepočítává bod X ve 3D do jeho obrazu x ležícím v obrazové rovině.

$$P = K[R \mid \vec{t}] \quad (2.12)$$

Pomocí P ve vztahu 2.12 [13] můžeme potom snadno dopočítat pozici bodu x v obrazové rovině, jak ukazuje vztah 2.13.

$$x = PX \quad (2.13)$$

2.4 3D rekonstrukce

3D rekonstrukce je proces, při kterém vzniká 3D model z dvojice obrazů scény. Je to opačný postup než při vzniku obrazu, kde se bod ve 3D zobrazuje do obrazové roviny [14].

Základem pro 3D rekonstrukci je množina obrazových korespondencí bodů $x_i \leftrightarrow x'_i$. Neznámou je zde množina bodů X_i ve 3D odpovídající svým obrazům x a x_i . Dále jsou také neznámé projekční matice kamer P a P' . Úkolem 3D rekonstrukce je zjistit neznámé X_i, P a P' , přičemž platí vztahy 2.14.

$$x_i = PX_i, \quad x'_i = P'X_i \quad (2.14)$$

3D rekonstrukce probíhá v několika krocích:

1. Výpočet fundamentální matice pomocí korespondujících bodů.
2. Výpočet projekční matice pomocí fundamentální matice.
3. Pro každou dvojici korespondujících bodů $x_i \leftrightarrow x'_i$ najít bod X_i ve 3D.

Kapitola 3

Analýza dopravních nehod

Kapitola popisuje některá základní fakta o vozidlových kamerách a dále rovněž zmiňuje software využívaný k analýze dopravních nehod.

3.1 Vozidlová kamera

Vozidlová kamera je zařízení, které zaznamenává obraz, zvuk a v některých případech i GPS souřadnice. Zařízení může také obsahovat 3D G-senzor, pomocí kterého lze zaznamenávat např. prudké brzdění, akcelerace, změny směru atd. Nahrávání většinou probíhá ve smyčce, proto nedochází k zastavení záznamu z důvodu zaplnění kapacity média.



(a) Eltrinox CarHD, převzato z [5]

(b) FULLHDCAM5, převzato z [6]

Obrázek 3.1: Vozidlové kamery

Vozidlová kamera je většinou umísťována za čelní sklo do oblasti kolem vnitřního zpětného zrcátka. Výhodou potom je, pokud je okraj čelního skla tónovaný.

Kamera je často využívána profesionálními řidiči. Záznam může posloužit jako důkazní materiál v případě autonehody, díky čemu lze usnadnit určení míry zavinění jednotlivých účastníků nehody. Pro tyto schopnosti je vozidlová kamera někdy označována jako černá skříňka.

Záznamy bývají často využívány soudními znalci, kteří je podrobují analýze. Mezi jeden z údajů, který lze takto zjistit, je například čas pohybu objektu ve scéně, jehož výpočet vychází ze snímkovací frekvence kamery.

Dále je možné záznam využít ke geometrické analýze obrazu. Pokud známe ohniskovou vzdálenost kamery a výšku jejího umístění vzhledem k vozovce, je možné odhadovat

vzdálenosti v rovině vozovky nebo v rovině s ní rovnoběžné. Tohle je ovšem založené na předpokladu, že je vozovka rovná. Pokud je vozovka v měřených místech nakloněná nebo zakřivená, není tato metoda vhodná [20].

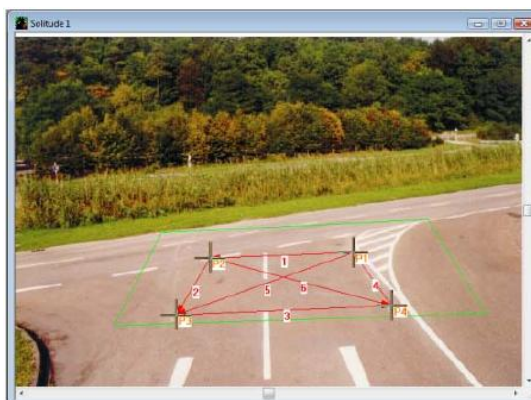
Na obrázku 3.1 můžeme vidět některé modely vozidlových kamer, které jsou běžně dostupné na našem trhu.

3.2 Software využívaný v praxi

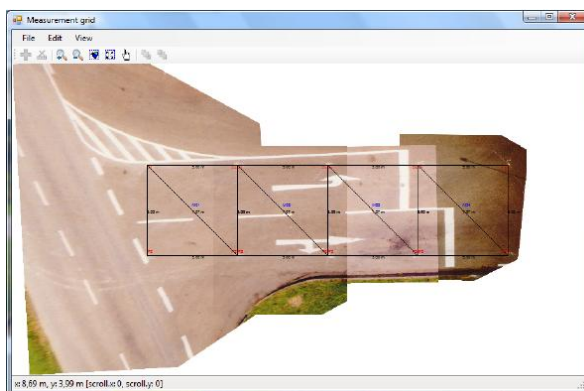
PC Rect a PC Crash

PC Rect je program pro rektifikaci fotografií a videí z místa dopravní nehody. Rektifikace znamená převedení zkoseného obrazu z fotografie do plánu v přesném měřítku. Pomocí něj lze odhadovat skutečné velikosti a vzdálenosti ve scéně.

Pro vytvoření plánu stačí zaznamenat daný úsek, manuálně rektifikovat první obrázek, na kterém je rektifikační kříž (obrázek 3.2a) a všechny ostatní snímky se poté automaticky propojí dohromady (obrázek 3.2b).



(a) Rektifikační kříž



(b) Plánek

Obrázek 3.2: PC Rect, převzato z [8]

PC Rect dokáže získat EXIF údaje z fotografií o tom, jakým fotoaparátém byl snímek pořízen, jaký byl použit objektiv, rotace, čas, clona, ostřicí bod atp. Tyto údaje jsou následně použity pro správné převedení obrázku tak, aby bylo eliminováno zkreslení a byl

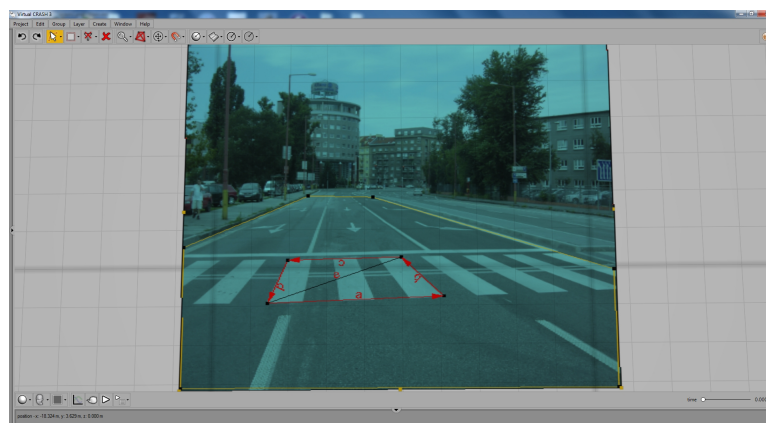
zachován poměr stran dle měřítka [7].

Od stejných autorů je také aplikace *PC Crash*, která se zabývá rekonstrukcí dopravních nehod. K těmto rekonstrukcím využívá 3D modely a reálné výpočty reakce těles. Její databanka obsahuje více než 1000 vozidel a stále se aktualizuje. Plánek, který je vytvořen v *PC Rect*, lze v *PC Crash* využít. Díky přenesení fotografií z místa nehody do plánu je poté projekt v *PC Crash* přesnější a autentičtější [7].

Virtual CRASH

Je program sloužící k simulaci dopravních nehod pomocí 3D modelů a znalostí fyzikálních zákonů. Simulace je možné provádět na plánech, které odpovídají měřítku.

Tyto plány vznikají rektifikací snímku z místa nehody. Proces rektifikace je podobný jako v řešení 3.2. Na snímku vyznačí uživatel rektifikační čtverec, kterému zadá skutečné rozměry (obrázek 3.3a). O samotnou rektifikaci se postará aplikace. Výsledek i s modelem automobilu můžeme vidět na obrázku 3.3b.



(a) Rektifikace



(b) Vyrenderovaný plánek s modelem

Obrázek 3.3: Virtual CRASH

Výsledky simulace jsou zobrazovány v 3D perspektivních pohledech, diagramech a tabulkách. Výsledná simulace lze exportovat jako video, případně jako sekvenci snímků. Aplikaci využívají některé pojišťovny a také například výrobce automobilů *ŠKODA AUTO a.s.* [9].

Kapitola 4

Nástroj Bundler

Celým názvem *Bundler: Structure from Motion (SfM) for Unordered Image Collections* je nástroj napsaný v C a C++, který vyvinul Noah Snavely z *University of Washington*. Jak už název napovídá, *Bundler* je nástroj, který z neuspořádané sbírky fotografií (např. získané z internetu) od různých autorů, vyfocených různými fotoaparáty, za rozdílného počasí, světelných podmínek a v rozdílnou roční dobu, dokáže vytvořit řídký 3D model scény a zrekonstruovat pozice kamer [17].

Jako vstup je zde sbírka fotografií, význačných bodů a korespondencí mezi těmito body. Na výstupu jsou soubory reprezentující výsledky rekonstrukce scény a pozice kamer. Tyto soubory jsou ve formátu *Polygon File Format* (.ply) a formátu .out, který je definován v dokumentaci. Protože aplikace pracuje iterativním způsobem vždy pouze s několika snímky najednou, dostáváme v průběhu rekonstrukce pravidelný výstup s aktuálním stavem scény pro daný okamžik. Výsledek 3D rekonstrukce je zde reprezentován point cloudem.

Nástroj *Bundler* implementuje a využívá následující metody [18]:

- RANSAC
- 8-bodový algoritmus pro výpočet fundamentální matice
- Odhad pozice dvou kalibrovaných kamer z 5 korespondujících obrazových bodů
- Triangulace z více paprsků

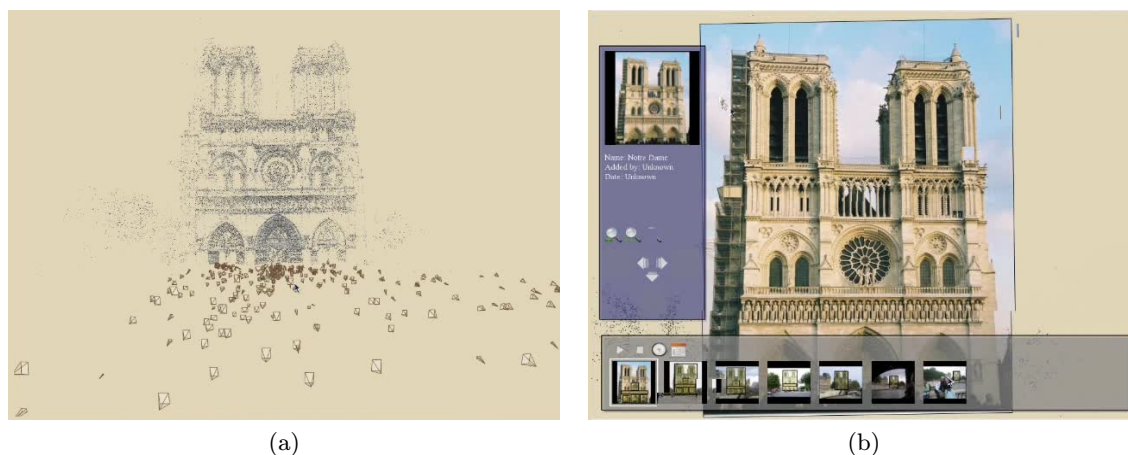
Samotný *Bundler* provádí až konečnou rekonstrukci scény a výpočet pozice kamer. Pro úpravu fotografií, hledání význačných bodů a určování jejich korespondencí využívá další externí aplikace:

- program *jhead* – pro získání EXIF informací z fotografií (např. ohnisková vzdálenost)
- program *ImageMagick* – převod z JPG do PGM formátu
- implementace algoritmu *SIFT* od *David Lowe* [16] – hledání význačných bodů
- program *gzip* pro komprimaci deskriptorů význačných bodů
- program *KeyMatchFull* – napsáno autorem *Bundleru*, slouží pro hledání korespondencí mezi význačnými body
- knihovna *libANN_char.so* – implementuje algoritmus hledání nejbližšího souseda

Nástroj *Bundler* byl prvotně vyvinut pro projekt **Photo Tourism**. Poté byl použit také v dalších projektech *University of Washington*, například v **Building Rome in a Day**.

4.1 Photo Tourism

Cílem autorů *Photo Tourism* bylo vytvořit program, který by sloužil pro procházení rozsáhlejších sbírek fotografií z jedné lokality pomocí 3D rozhraní. Program využívá nástroj *Bundler*, který mu vytvoří řídký 3D model. Tento model je zobrazen v grafickém uživatelském rozhraní a je dán do souvislosti s fotografiemi, ze kterých vznikl. Autoři kladou velký důraz právě na toto propojení snímku s 3D modelem. V programu lze označovat oblasti scény jak na snímcích, tak ve 3D modelu a přiřazovat jim příslušný popis, který se potom bude zobrazovat i na dalších snímcích obsahujících daný objekt [18].



Obrázek 4.1: Photo Tourism – rozhraní, převzato z [19]



Obrázek 4.2: Photo Tourism – mapa, převzato z [19]

Na obrázku 4.1a vidíme zobrazen 3D model katedrály Notre-Dame reprezentovaný point cloudem. Ve scéně jsou rovněž zaznamenány pozice kamer, ze kterých byly snímky pořízeny. Na obrázku 4.1b je zobrazeno přímo rozhraní pro procházení snímků.

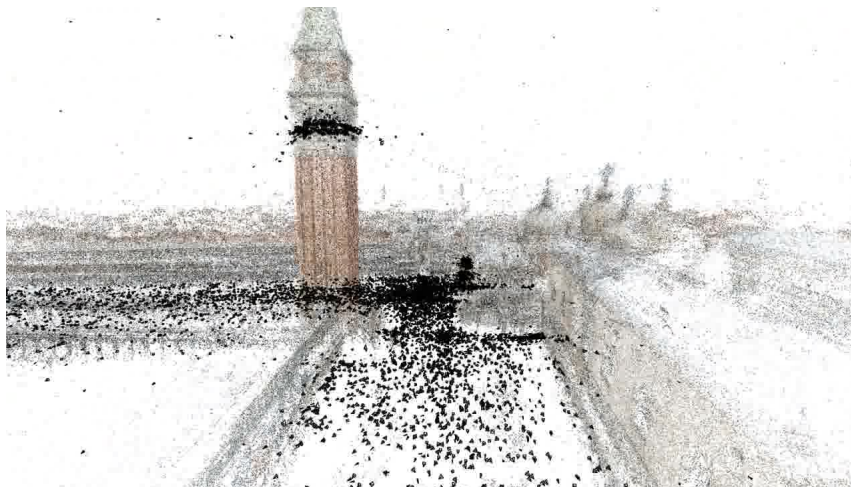
Autoři se také snažili dát výsledný model a snímky do souvislosti se zeměpisnou polohou. Údaj o GPS souřadnicích bohužel u většiny dnešních fotografií stále chybí. Využili proto

znalosti relativních pozic kamer ve scéně a ostatních objektů k umístění modelu do mapy nebo leteckého snímku, který je již umístěn do GPS souřadnic. Toto můžeme vidět na snímcích Prahy na obrázku 4.2.

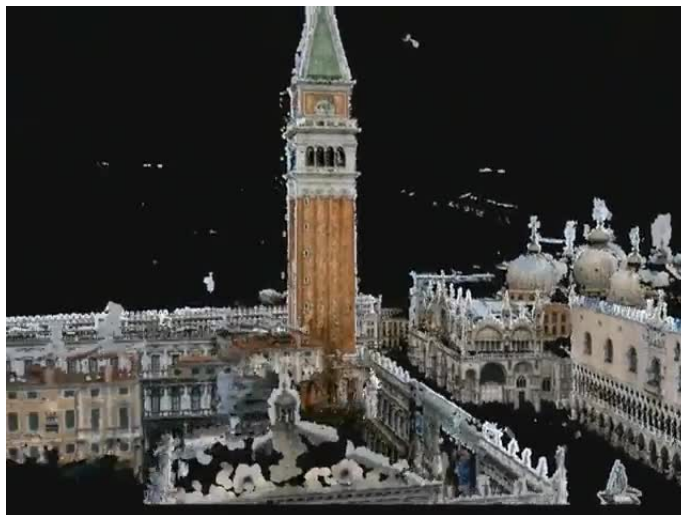
Na stránkách projektu je možné si vyzkoušet demo této aplikace [19].

4.2 Building Rome in a Day

V tomto projektu se autoři snažili řešit problém 3D rekonstrukce z fotek získaných z internetových sbírek. Zjistili, že například na serveru *Flickr.com* se nachází prakticky kompletní fotodokumentace významných částí města Řím, což je přivedlo k myšlence vytvořit systém, který stáhne ze serveru *Flickr.com* fotky spojené s požadovaným městem (původně Řím), které budou sloužit pro 3D rekonstrukci [11].



(a) SfM model, převzato z [4]



(b) MVS model, převzato z [12]

Obrázek 4.3: Náměstí svatého Marka, Benátky

Za pomoci paralelních distribuovaných systémů a několika stovek tisíc fotografií se po-

dařilo vytvořit rozsáhlé 3D modely měst jako Řím, Benátky a Dubrovník.

Jak již bylo řečeno v úvodu kapitoly 4, *Bundler* vytváří řídký model. Autoři proto spolupracují s projektem *Clustering Views for Multi-view Stereo*, který dokáže z řídkého *SfM* (*Structure from Motion*) modelu vytvořit hustý *MVS* (*Multi-view Stereo*) model [12]. Na obrázku 4.3 vidíme náměstí svatého Marka v Benátkách, které je zde reprezentováno SfM a MVS modelem.

Kapitola 5

Návrh aplikace pro zpracování záznamu z vozidlové kamery

Cílem mé práce je vytvořit aplikaci, která by sloužila pro odhad vzdáleností ve snímcích získaných ze záznamu z vozidlové kamery. Pro odhad vzdáleností využiji *mračna bodů* (*point cloud*), které bude získáno z 3D rekonstrukce snímků. Vzdálenosti, které budou odhadovány, budou právě mezi těmito body. Aplikace bude mít vlastní grafické uživatelské rozhraní, pomocí kterého jí uživatel bude moci ovládat. Aplikace bude mít následující funkce:

- Přehrávání videozáznamu
- Exportování snímků ze záznamu
- 3D rekonstrukce
- Uložení výsledků 3D rekonstrukce a ostatních výstupů do souboru
- Vizualizace point cloudu
- Odhadování vzdáleností mezi body

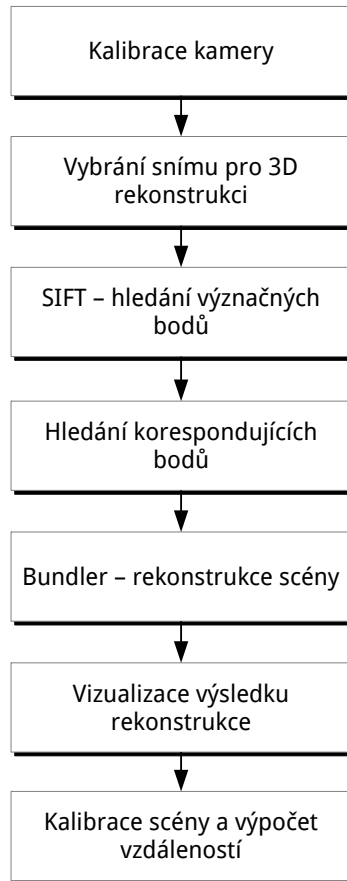
V aplikaci se počítá s aktivní spoluprací uživatele, který ručně vybere část videozáznamu pro rekonstrukci a ručně kalibruje scénu. Měření vzdálenosti je už pak velmi jednoduché. Uživatel kliká myší na body ve snímku nebo point cloudu a program mu vypisuje vzdálenosti.

Vstupem navrhované aplikace je videozáznam z vozidlové kamery, který lze v aplikaci přehrávat a který slouží jako vstup pro 3D rekonstrukci. Na výstupu aplikace je zrekonstruovaná scéna reprezentovaná point cloudem.

Výstupní point cloud bude aplikací vizualizován a bude s ním možné interagovat pomocí myši. Aplikace bude také zobrazovat snímky, ze kterých 3D rekonstrukce vychází a ve kterých budou zobrazeny význačné body, které byly využity při rekonstrukci scény.

Výsledky 3D rekonstrukce budou uloženy do souboru. Proto pro opakované měření ve scéně nebude nutné podstoupit celý proces 3D rekonstrukce znovu, ale postačí pouze otevřít příslušný soubor.

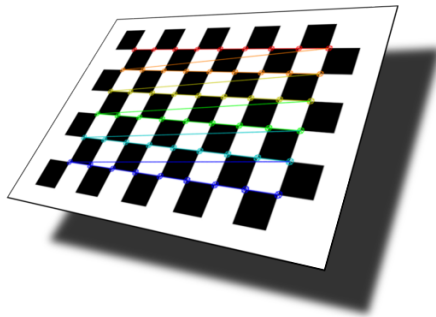
Dále se budu věnovat jednotlivým krokům, pomocí kterých jsem problém návrhu aplikace řešil.



Obrázek 5.1: Návrh řešení – schéma

5.1 Kalibrace kamery

Prvním krokem, který je nutné udělat, je kalibrace kamery. Ke kalibraci kamery se často používají různé *kalibrační vzory*. Využívá se zde faktu, že známe skutečné rozměry a proporce kalibračního vzoru. V mém řešení jsem využil šachovnicový kalibrační vzor.



Obrázek 5.2: Kalibrace v *OpenCV*, kalibrační vzor

Kalibrace se skládá z několika kroků. Nejprve je třeba mít vytištěn kalibrační vzor, který se připevní na rovný povrch, aby nedocházelo ke zkreslení. Poté se natočí kalibrační video s různými úhly pohledů a různými rotacemi vůči kalibračnímu vzoru. Z videa si vyberu několik snímků z různých pozic, které poté využiji pro samotnou kalibraci.

Výstupem kalibrace jsou vnitřní a vnější parametry kamery (viz. 2.3). Jediný údaj, který je ale pro projekt důležitý a který bude využit v dalších krocích, je ohnisková vzdálenost.

Pro výpočet kalibračních dat využiji knihovnu pro počítačové vidění *OpenCV*, která je dostupná pod BSD licenci [1].

5.2 Vybrání snímků pro 3D rekonstrukci

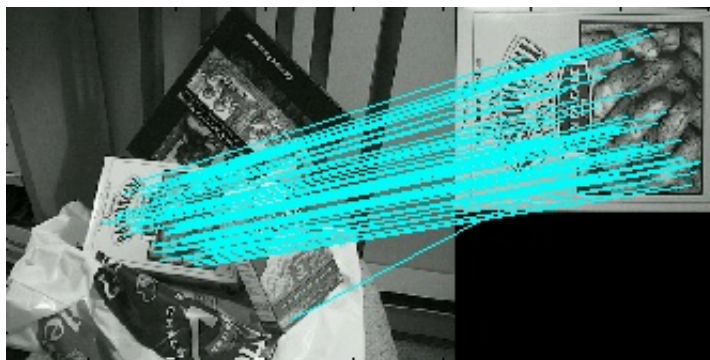
Protože vstupem pro nástroj *Bundler* (viz. kapitola 4) je sada snímků, je potřeba vyexportovat z vybrané části záznamu tyto snímky. Zde se očekává aktivní spolupráce uživatele, který po shlédnutí videa vybere část záznamu, kterou chce zrekonstruovat. Z této části záznamu se vyberou snímky pro další zpracování.

5.3 SIFT – hledání význačných bodů

Dalším nutným krokem bude vyhledání význačných bodů. K tomu mi bude sloužit algoritmus *SIFT* (*Scale Invariant Feature Transform*). Autorem toho algoritmu je David Lowe z *University of British Columbia*, který pro něj vytvořil demonstrační implementaci *SIFT demo program*. Algoritmus SIFT je chráněn patentem a je možné ho volně používat pouze pro výzkumné účely [16]. Pro každý význačný bod bude vytvořen deskriptor obsahující 128 celých čísel.

5.4 Hledání korespondujících bodů

Mezi jednotlivými snímky je třeba najít korespondující body. Při hledání korespondujících bodů se využívá deskriptorů vytvořených v kroku 5.3. Tyto deskriptory se vzájemně porovnávají pomocí Eukleidovských vzdáleností. K tomuto porovnávání slouží aplikace *Key-Match.Full*, která byla vytvořena na míru pro nástroj *Bundler* (viz. kapitola 4). Aplikace je dostupná pod licencí GNU GPL [17].



Obrázek 5.3: Ukázka korespondencí, převzato z [16]

5.5 Bundler – rekonstrukce scény

Samotnou 3D rekonstrukci scény bude zajišťovat nástroj *Bundler* (viz. kapitola 4). Vstupem pro rekonstrukci pomocí nástroje *Bundler* budou:

- Seznam snímků (získané v kroku 5.2).
- Jednotlivé ohniskové vzdálenosti pro každý snímek (získáno v kroku 5.1). V našem případě mají všechny snímky stejnou ohniskovou vzdálenost.
- Seznam význačných bodů a jejich vzájemné korespondence (získáno v krocích 5.3 a 5.4).

Výstupy nástroje *Bundler* budou:

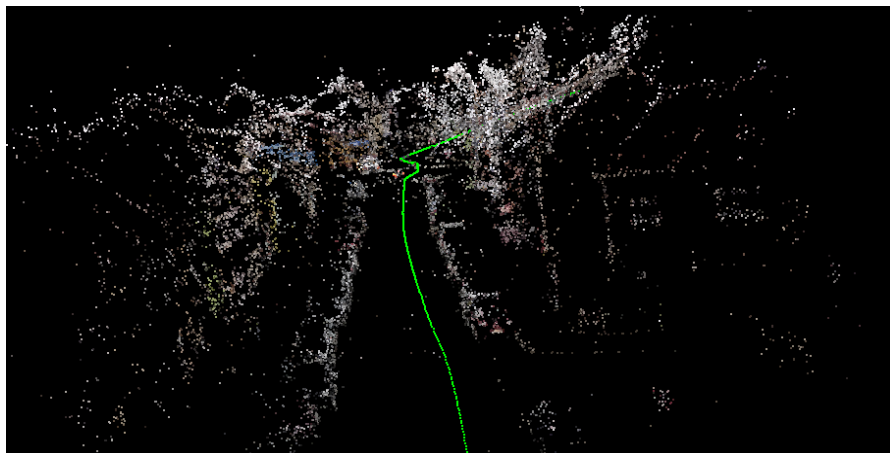
- Zrekonstruovaná scéna reprezentovaná point cloudem.
- Matice rotace a vektory posunutí kamery pro každý snímek.
- Korespondence mezi body point cloudu a význačnými body na jednotlivých snímcích včetně pozic těchto bodů na snímku.

Výstupní soubory bude poté třeba analyzovat a provést v nich drobné úpravy. Pomocí vztahu 2.11 budou dopočítány pozice kamer ve scéně.

Aplikace *Bundler* je dostupná pod licencí GNU GPL [17].

5.6 Vizualizace výsledků rekonstrukce

Pro vizualizaci point cloudu bude použita knihovna *Point Cloud Library (PCL)*, která je dostupná pod BSD licencí [2]. Point cloud bude možné přibližovat a manipulovat s ním ve všech třech osách. Body v point cloudu budou mít stejnou barvu jako byla barva bodů na snímcích, ze kterých byl bod zrekonstruován. Pozice kamer bude v point cloudu vyznačena výrazně jinou barvou než body scény. Na body bude možné klikat myší a tím mezi nimi měřit vzdálenosti.



Obrázek 5.4: Ukázka vizualizace v *PCL*

Body, ze kterých byla zrekonstruována scéna, budou vyznačeny ve zobrazovaných snímcích. Také na tyto body bude možné klikat a tím měřit vzdálenosti.

Obě dvě tato zobrazení budou provázány. Pokud bude vybrán bod ze snímku, vyznačí se s ním korespondující bod v point cloudu a naopak.

5.7 Kalibrace scény a výpočet vzdálenosti

U kalibrace scény se počítá s aktivní spoluprací uživatele, který na snímcích vyznačí pomocí význačných bodů skutečné rozměry některých objektů ve scéně, jejichž velikost je známá. V úvahu přicházejí například velikosti vodorovného a svislého značení nebo případně rozměry jiných automobilů ve scéně atd. Pomocí těchto známých rozměrů budou určeny kalibrační jednotky, které budou sloužit pro přepočítání mezi jednotkami point cloudu a skutečnými rozměry.

Vzdálenost mezi body v prostoru se vypočítá podle vztahu 5.1 pro Eukleidovskou vzdálenost. Dále jsou určeny kalibrační jednotky k_i pro i kalibračních objektů podle vztahu 5.2, kde dm_i je vzdálenost bodů v metrech zadaná uživatelem a dp_i vzdálenost bodů v jednotkách point cloudu.

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5.1)$$

$$k_i = \frac{dm_i}{dp_i} \quad (5.2)$$

Výpočet reálné vzdálenosti probíhá za pomoci aritmetického průměru součinů kalibračních jednotek a vzdálenosti v point cloudu, jak je uvedeno ve vztahu 5.3, kde n je počet kalibračních jednotek, dp je vzdálenost v jednotkách point cloudu a k_i je kalibrační jednotka. Aritmetický průměr by potenciálně mohl být nahrazen mediánem, který na rozdíl od aritmetického průměru nebývá tolik ovlivněn extrémními hodnotami.

$$d_r = \frac{1}{n} \sum_{i=1}^n dp \cdot k_i \quad (5.3)$$

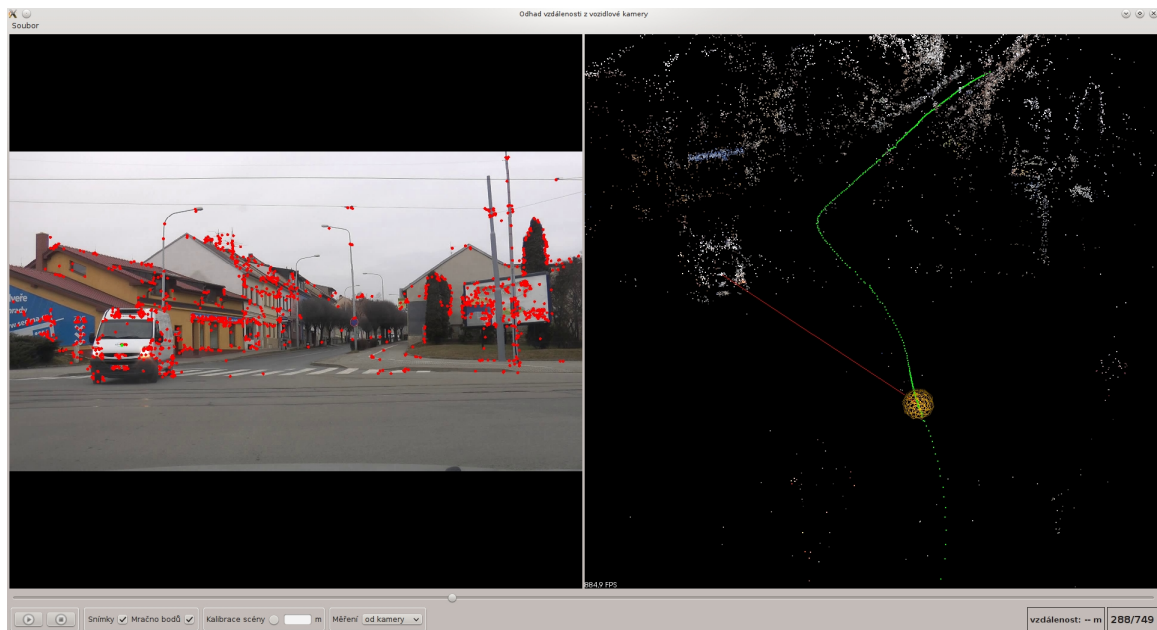
5.8 Návrh grafického uživatelského rozhraní

Pro tvorbu grafického uživatelského rozhraní bude použita knihovna *Qt*, která je dostupná pod licencí GNU LGPL [3]. Hlavní okno aplikace je rozděleno na několik částí, jak je vidět na obrázku 5.5.

V levé části okna je přehráván záznam a po provedení 3D rekonstrukce nebo načtení dřívějších výsledků slouží pro zobrazování snímků, které byly použity pro 3D rekonstrukci. Na snímcích jsou vyznačeny významné body.

V pravé části okna je zobrazen point cloud, se kterým lze interagovat pomocí myši. V point cloudu je také zaznačena aktuální pozice kamery ve scéně.

Ve spodní části okna jsou umístěny všechny důležité ovládací prvky aplikace. Tyto prvky nebudou dostupné vždy všechny najednou, ale bude záležet na tom, zda je zrovna přehráván záznam nebo jestli je zobrazen výsledek rekonstrukce. V pravém dolním rohu se zobrazují údaje o vzdálenosti, počtu snímků nebo případně času přehrávaného videa.



Obrázek 5.5: Návrh grafického rozhraní

V horní části okna se nachází menu, ve kterém se skrývají položky pro otevření záznamu, otevření analyzovaných dat, spuštění analýzy (3D rekonstrukce) a ukončení aplikace.

5.9 Testovací data

Jako testovací data byly použity záznamy z jízdy automobilem, které byly pořízeny běžnou kamerou *Panasonic HDC-TM700*. Kamera byla během jízdy upevněna za čelním sklem. Autorem těchto videí je pan Doc. Ing. Adam Herout, Ph.D. Video byla pořízena v rozlišení 1920x1080 pixelu, rychlostí 50 snímků za sekundu.



Obrázek 5.6: Snímek z kamery

Kapitola 6

Implementace

V této kapitole bude popsána implementace řešení aplikace. Aplikace byla vyvíjena pro operační systém Linux. Při implementaci byly využity následující knihovny:

- *OpenCV* – knihovna pro počítačové vidění
- *Point Cloud Library (PCL)* – knihovna pro práci s point cloudem
- *Qt* – knihovna pro vývoj grafického uživatelského rozhraní

6.1 Kalibrace kamery

Kalibrace kamery je implementovaná v souboru `camera_calibration.cpp`. Pro kalibraci je použit šachovnicový vzor, pro který existují v knihovně *OpenCV* funkce pomáhající s kalibrací. Implementace vychází do jisté míry z tutoriálu na webu *OpenCV* [10], který byl ale výrazně přepracován, aby vyhovoval potřebám projektu. Kalibrace probíhá na snímcích exportovaných uživatelem z videa nebo případně přímo na fotografiích.

Program má jeden povinný parametr, kterým je soubor ve formátu XML, který specifikuje rozměry šachovnice, velikost čtverečku, výstupní soubor ve formátu XML a seznam snímků pro kalibraci.

Výstupem programu je XML soubor, který obsahuje matici vnitřních parametrů kamery. Ostatní výstupy, které vyprodukovala funkce pro kalibraci, nejsou v souboru uvedeny, protože jejich znalost není dále v projektu potřebná.

Program pro kalibraci využívá funkci `cv::calibrateCamera`. Jejimi vstupními parametry jsou pole souřadnic rohů šachovnice v prostoru, pole souřadnic nalezených rohů na snímcích a velikost snímku. Výstupem jsou matice vnitřních parametrů, koeficienty zkreslení, vektory posunutí a matice rotace kamer pro každý snímek.

Pole souřadnic rohů šachovnice v prostoru lze snadno dopočítat ze znalostí rozměru šachovnice a velikostí jednotlivých čtverečků. Budeme předpokládat, že pozice šachovnice se v prostoru nemění a pohybuje se pouze kamera. Tento výpočet je implementován ve funkci `calcBoardCornerPositions`.

Nalezení šachovnice a souřadnic jejich rohů ve snímku má na starosti funkce `cv::findChessboardCorners`, která tyto hodnoty vrací v parametrech. Pro zpřesnění výsledků je použito funkce `cv::cornerSubPix`. Tyto funkce se volají pro každý snímek a jejich výsledky se průběžně ukládají do vektoru, aby mohly být později použity pro kalibraci.

Během hledání šachovnic jsou snímky zobrazovány funkcí `cv::imshow`. Pokud je šachovnice nalezena, jsou na snímku vyznačeny její rohy.

6.2 Přehrávání videa

Přehrávání videa je implementováno ve třídě `VideoFrame` v souborech `videoframe.cpp` a `videoframe.h`. Slouží k němu kombinace funkcí knihoven *OpenCV* a *Qt*.

Pro manipulace s videem je využita třída `cv::VideoCapture`, která umožňuje otevřít soubor s videem podle zadané cesty, načíst snímky a zjistit užitečné informace jako jsou snímkovací frekvence, délka videa atp.

Vykreslování snímků probíhá přímo na widgetu, který je provázán s třídou `VideoFrame`. V metodě `VideoFrame::paintEvent` je snímek zmenšen a posunut tak, aby vyhovoval velikosti okna a poté vykreslen.

Dále byl vytvořen časovač typu `QTimer`, který v pravidelných intervalech podle snímkovací frekvence volá metodu `VideoFrame::redrawFrame`, která načte nový snímek z videa a vykreslí ho. Pomocí časovače se řídí celé přehrávání videa. Pokud časovač neběží, vykreslování se zastaví na posledním načteném snímku.

6.3 Spuštění externích aplikací

Spuštění externích aplikací je implementováno ve třídě `BundlerMachine` v souborech `bundlermachine.cpp` a `bundlermachine.h`. Třída je typu `QThread`, což v praxi znamená, že všechny metody volané z metody `BundlerMachine::run` poběží v jiném vlákne než zbytek aplikace. Důvodem pro oddělené vlákno je fakt, že externí aplikace můžou běžet i několik hodin a pokud by běžely v hlavním vlákne, blokovaly by celé uživatelské rozhraní.

Ještě před samotným spuštěním externích aplikací je třeba vyexportovat z videa snímky a uložit je do souborů. Toto probíhá za účasti uživatele, který vybere časový úsek videa, načte XML soubor s vnitřními parametry kamery, určí počet snímků pro zpracování a složku, do které budou snímky uloženy. Do této složky budou ukládány i výstupy všech externích programů včetně konečného výstupu aplikace. Uložení snímků do souborů má na starosti metoda `BundlerMachine::extractImages`.

Externí aplikace jsou spouštěny pomocí třídy typu `QProcess`. Průběžné výstupy z aplikací jsou vypisovány v uživatelském rozhraní. Aplikace jsou spouštěny v tomto pořadí a mají tyto funkce:

- *mogrify* – nástroj dostupný s aplikací *ImageMagick*, slouží k převedení snímků z formátu jpg do pgm¹, s kterým pracuje *sift* od Davida Lowe
- *sift* – vyhledání význačných bodů, výstup se ukládá do souboru `.key`
- *gzip* – komprimace souborů `.key`
- *KeyMatchFull* – hledání společných význačných bodů mezi snímky, výstup se ukládá do souboru `matches.init.txt`
- *Bundler* – 3D rekonstrukce, vytvoření point cloudu, výstupy v souborech `.out` a `.ply`

Externí aplikace jsou spuštěny z metody `BundlerMachine::runBundler`. *KeyMatchFull* a *Bundler* běží pouze jednou, ostatní aplikace běží pro každý snímek.

Některé z těchto aplikací jsou přibaleny v archívu společně s nástrojem *Bundler*. Nacházejí se zde rovněž další aplikace, pomocí kterých se extrahují EXIF informace ze snímků.

¹Do pgm umí sice ukládat i knihovna *OpenCV*, tento formát ale není kompatibilní s implementací SIFTu od Davida Lowe

Jelikož snímky exportované z videa žádné takové informace neobsahují, nejsou aplikace využity. V archívu se nachází také skript, pomocí kterého se všechny tyto aplikace spouštějí. Tento skript není využit, protože neodpovídá požadavkům projektu.

6.4 Zpracování výstupu z Bundleru

O zpracování výstupu se rovněž stará třída `BundlerMachine`. Zpracování začíná zavoláním metody `BundlerMachine::processBundlerOutput`. Nástroj *Bundler* produkuje výstup již za svého běhu. Pro zpracování jsou využity pouze soubory s konečnou rekonstrukcí. Z výstupu lze získat informace o souřadnicích rekonstruovaných bodů, o pozicích kamer jednotlivých snímků a o význačných bodech, které byly použity pro rekonstrukci. Abychom získali tyto údaje, je třeba výstup patřičně zpracovat a upravit. Výstupy jsou uloženy v souborech dvojího typu.

Prvním formátem je *PLY* (*Polygon File Format*), ve kterém je uložený point cloud zrekonstruovaných bodů společně s body, které reprezentují pozice kamer.

Druhým formátem je soubor s koncovkou `.out`, jehož syntaxe je definována v dokumentaci *Bundleru* [17]. V podstatě se jedná pouze o posloupnosti čísel, které jsou odděleny pomocí mezer a nových řádků.

Pozice kamer

Informace o pozicích kamer se nacházejí na začátku souboru s koncovkou `.out`. Pozice jsou zde vyjádřeny pomocí matic rotace a vektorů posunutí. Tyto údaje jsou přepočítány na souřadnice pomocí vzorce 2.11. Pozice kamer jsou poté uloženy do souboru `cameras.ply`. Zpracování pozic kamer je implementováno v metodě `BundlerMachine::calculateCameras`.

Souřadnice kamer jsou uloženy i ve výstupním souboru typu *PLY*. Problémem však je, že nejsou uloženy ve stejném pořadí jako snímky, takže nelze určit, ke kterému snímku která kamera patří.

Upravení výstupu *PLY*

Zde je třeba upravit hlavičku souboru typu *PLY* tak, aby ho bylo možné otevřít pomocí knihovny *PCL*. Konkrétně se jedná o problém, kdy knihovna není schopná zpracovat barvy typu *diffuse*. Proto musí být tato specifikace odstraněna z hlavičky.

Jak již bylo zmíněno v předchozí kapitole, ve výstupu typu *PLY* jsou uloženy i souřadnice kamer. Ty je třeba odstranit, protože už jsou přepočítány a umístěny ve zvláštním souboru.

Výsledky úprav jsou uloženy do nového souboru `points.ply`. Úprava je implementována v metodě `BundlerMachine::editPly`.

Zpracování význačných bodů použitých pro rekonstrukci

Soubor s koncovkou `.out` nám také poskytuje informace o tom, ze kterých význačných bodů na jednotlivých snímcích byly body zrekonstruovány. Ke každému bodu je přiřazen seznam snímků, na kterých se nachází obraz bodu a jeho pozice v něm. Pro vizualizaci je ale potřeba znát opačný vztah, tedy jaké body jsou využity z daného snímku. Proto je nutné tento vztah převést, aby vyhovoval požadavkům. Transformace je implementována v metodě `BundlerMachine::writeImagesPointsFile` a uložena do souboru `images_points.txt`.

Uložení konečného výstupu aplikace

Aplikace generuje následující výstupy:

- vyexportované snímky z videa ve formátu jpg
- soubor `cameras.ply` se souřadnicemi kamer
- soubor `points.ply` s rekonstruovanými body
- soubor `images_points.txt` s význačnými body ve snímcích

Cesty k těmto výstupům jsou uloženy do XML souboru, který se využívá pro znovutevření výsledků. Vytvoření a uložení XML je implementováno v metodě `BundlerMachine::writeXML`.

6.5 Vizualizace výsledku rekonstrukce

Vizualizace je implementována ve třídě `PointCloudViewer` v souborech `pointcloudviewer.cpp` a `pointcloudviewer.h`. Třída je typu `QVTKWidget`, což umožňuje zobrazovat point cloud přímo na widget.

K vizualizaci slouží třída `pcl::visualization::PCLVisualizer`, pomocí které se načte ze souboru point cloud a která umožňuje do vizualizace přidávat další objekty. Současná pozice kamery je označena pomocí koule a pozice ostatních kamer jsou označeny body se zelenou barvou.

Na body ve vizualizaci je možné klikat pomocí kombinace *shift+levé tlačítko myši*. Klikání v knihovně *PCL* nefunguje příliš spolehlivě, proto občas není možné na vybraný bod kliknout nebo knihovna vyhodnotí nějaký jiný bod z okolí.

Ve widgetu třídy `VideoFrame` se místo videa zobrazují snímky, které byly využity pro rekonstrukci, ve kterých jsou vykresleny význačné body. I na tyto body lze klikat.

Snímky a point cloud jsou vzájemně provázány. Kliknutí na bod v jednom zobrazení se projeví i ve zobrazení druhém. Posun kamery a snímku je řízen pomocí slideru v uživatelském rozhraní.

Kapitola 7

Výsledky

V následující kapitole budou prezentovány dosažené výsledky pomocí testovacích měření. Tyto měření budou ke konci kapitoly shrnuty.

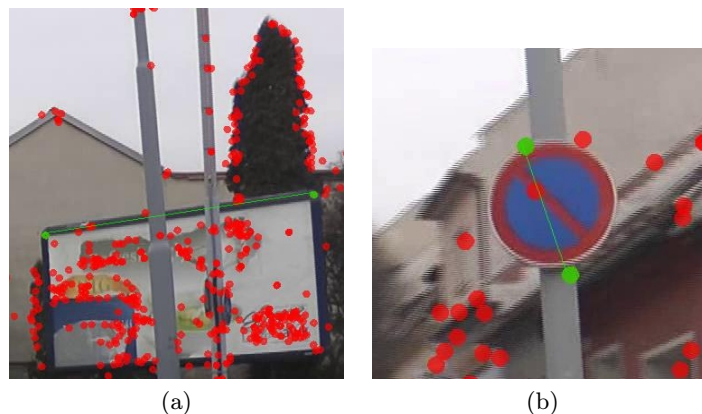
7.1 Testovací měření

Jak lze předpokládat, přesnost měření je přímo závislá na tom, jak dobře je schopen uživatel kalibrovat scénu a také na kvalitě rekonstrukce. Pokud se její výstup ani neblíží podobě skutečné scény, lze předpokládat, že výsledky měření budou velmi nepřesné, ne-li zcela špatné. Dosažené výsledky budou demonstrovány na následujících příkladech z testování.

Ověřování probíhalo pomocí znalostí rozměrů dalších skutečných objektů ve scéně. Protože je možné odhadovat vzdálenosti pouze mezi význačnými body, nepasuje měření přesně na hranice objektů. K měření byly vybrány body, které se těmto hranicím alespoň blížily. Proto zde dochází ke snížení přesnosti odhadu již ve fázi označování objektů.

Scéna 1

První demonstrační scénou jsou ulice *Stránského* a *Královopolská* v Brně. Pro kalibraci scény (obrázek 7.1) byl využit Euro billboard o známé velikosti **5,1 x 2,4 m** a značka *Zákaz stání* o průměru **0,7 m**. Názvy objektů, odhadované vzdálenosti a skutečné vzdálenosti jsou uvedeny v tabulce 7.1.



Obrázek 7.1: Kalibrace scény 1

Objekt	Skutečná vzdálenost	Odhadovaná vzdálenost
Renault Master – šířka	1,99 m	1,81 m
Renault Master – délka	5,9 m	5,47 m
Renault Master – výška	2,48 m	2,45 m
Renault Megane Coupe – šířka	1,8 m	1,59 m
Přechod pro chodce 1	1,5 m	1,58 m
Přechod pro chodce 2	5,5 m	5,21 m
značka Zákaz odbočení 1 – průměr	0,7 m	0,65 m
značka Zákaz odbočení 2 – průměr	0,7 m	0,54 m
Škoda Fabia – šířka	1,64 m	1,56 m
Škoda Fabia – rozvor	2,46 m	2,76 m
Škoda Superb – rozvor	2,8 m	3,2 m
Škoda Octavia – rozvor	2,57 m	3,02 m
Průměrná chyba měření		Směrodatná odchylka
0,222 m		0,144 m

Tabulka 7.1: Scéna 1 – výsledky

Scéna 2

Druhou demonstrační scénou je ulice *Husitská* v Brně. Ke kalibraci byly využity rozvor Volva XC90 **2,857 m** a velikost značky Přechod pro chodce **0,5 m**. Výsledky můžeme vidět v tabulce 7.2.

Objekt	Skutečná vzdálenost	Odhadovaná vzdálenost
Ford Fiesta – délka	3,82 m	3,7 m
značka Zákaz odbočení – průměr	0,7 m	0,66 m
SPZ	0,52 m	0,47 m
značka Kruhový objezd – průměr	0,7 m	0,26 m
Přechod pro chodce	1 m	0,41 m
Škoda Octavia – rozvor	2,57 m	2,82 m
BMW X5 – rozvor	2,82 m	2,16 m
Opel Corsa – šířka	1,73 m	1,42 m
Průměrná chyba měření		Směrodatná odchylka
0,306 m		0,223 m

Tabulka 7.2: Scéna 2 – výsledky

Scéna 3

Třetí scénou je další část ulice *Husitské* v Brně. Ke kalibraci slouží délka části přechodu **3 m** a značka Parkoviště s podélným stáním **0,7 m**. Výsledky jsou uvedeny v tabulce 7.3.

Objekt	Skutečná vzdálenost	Odhadovaná vzdálenost
Přechod pro chodce 1	1,5 m	1,61 m
Přechod pro chodce 2	1 m	0,98 m
značka Hlavní pozem. k. – úhlopříčka	0,7 m	0,6 m
Ford Focus – šířka	1,83 m	1,4 m
Ford Focus – výška	1,44 m	1,75 m
Ford Mondeo – šířka	1,75 m	1,66 m
SPZ	0,52 m	0,49 m
Irisbus Citybus 12M – šířka	2,5 m	2,37 m
Suzuki Grand Vitara – šířka	1,81 m	1,46 m
Ford Fusion – výška	1,48 m	1,59 m
Průměrná chyba měření		Směrodatná odchylka
0,168 m		0,135 m

Tabulka 7.3: Scéna 3 – výsledky

Scéna 4

Čtvrtou scénou je ulice *Sucheňova* v Třebíči. Pro kalibraci byl využit Euro billboard o rozměrech **5,1 x 2,4 m** a značka Odbočovací pruh o rozměrech **1 x 1,5 m**. Výsledky můžeme vidět v tabulce 7.4.

Objekt	Skutečná vzdálenost	Odhadovaná vzdálenost
značka Návěst před křižovatkou – výška	1,5 m	1,65 m
značka Návěst před křižovatkou – šířka	1 m	1,06 m
značka Komunikační cíl	1 m	1,09 m
Karosa B 732 – výška	3,16 m	4,03 m
Karosa B 732 – šířka	2,5 m	3,13 m
značka Kulturní cíl	1 m	0,85 m
značka Začátek úseku	0,5 m	0,56 m
Průměrná chyba měření		Směrodatná odchylka
0,287 m		0,302 m

Tabulka 7.4: Scéna 4 – výsledky

Scéna 5

Pátou scénou je ulice *Bráfova třída* v Třebíči. Pro kalibraci scény byl využit Euro billboard o rozměrech **5,1 x 2,4 m** a značka Směrová tabule o šířce **1,1 m**. Výsledky jsou uvedeny v tabulce 7.5

Objekt	Skutečná vzdálenost	Odhadovaná vzdálenost
značka Průběh úseku – výška	0,5 m	0,45 m
značka Přejechod pro chodce – šířka	0,5 m	0,42 m
značka Komunikační cíl	1 m	0,93 m
Euro billboard – výška	2,4 m	2,39 m
značka Hlavní pozem. k. – úhlopříčka	0,7 m	0,68 m
značka Informační tabule – výška	1,5 m	1,45 m
Peugeot 807 – šířka	1,85 m	1,74 m
značka Nemocnice – šířka	0,5 m	0,38 m
značka Zákaz zastavení – průměr	0,7 m	0,8 m
Přejechod pro chodce	1,5 m	1,87 m
Peugeot 206 – výška	1,43 m	1,71 m
Seat Alhambra – šířka	1,81 m	1,57 m
Průměrná chyba měření		Směrodatná odchylka
0,125 m		0,108 m

Tabulka 7.5: Scéna 5 – výsledky

Scéna 6

Šestou demonstrační scénou je část úseku dálnice D1 ve směru na Prahu. Rekonstrukce se zde ukázala jako velmi problematická. Kvůli malé členitosti okolí vozovky nebyl rozpoznán velký počet význačných bodů, které slouží pro rekonstrukci. Ty byly rozpoznávány především na okolních jedoucích vozidlech (obrázek 7.2). Nástroj Bundler proto nedokázal úspěšně určit pozice některých kamer a výsledný point cloud se nedá použít pro odhadování vzdáleností.

7.2 Shrnutí výsledků

Jak je patrné z výsledkových tabulek, přesnost odhadu se v jednotlivých případech měření liší, což je způsobeno především kolísající kvalitou 3D modelu v jednotlivých částech scény. Lepší výsledky byly naměřeny v prostorově menších scénách městského typu, ve kterých díky větší členitosti a různorodosti, bylo dosahováno větší kvality výstupu rekonstrukce.

Odhadování vzdálenosti ze záznamu z jízdy na dálnici se ukázalo jako velmi problematické. Stejně tak tomu bylo i u cest, které neměly členitější okolí, jakou jsou například cesty vedoucí mezi poli. V těchto scénách nebylo většinou měření prakticky možné nebo byly naměřeny velmi zcestné hodnoty. Podobný závěr platí i pro noční scény, ve kterých metody užité v řešení obecně nefungují příliš spolehlivě.

Během testování byl pozorován jev, kdy měření v blízkosti kalibračních objektů dosahovalo lepších výsledků než ve zbytku scény. Avšak tento jev se neprojevil ve všech scénách, které byly měřeny. Také platilo, že využití větších objektů pro kalibraci zvyšuje přesnost odhadovaných vzdáleností. V aplikaci byla implementována možnost kalibrace pomocí více objektů, která se v praxi ukázala jako velmi užitečná. Kalibrace proto nebyla tolik závislá na kvalitě rekonstrukce jednoho objektu a chybu bylo možné potencionálně snížit díky vícenásobné kalibraci.



Obrázek 7.2: Scéna 6 – dálnice

Vliv na výsledky měl i počet snímků, které byly využity. Nedá se říct, že by 3D model téže scény s větším počtem snímků byl o mnoho přesnější při odhadování vzdálenosti, ale obsahoval větší počet bodů, což vedlo k lepším možnostem měření. Větší počet snímků však neúměrně prodlužoval rekonstrukci. Důvodem je, že při zpracování záznamu probíhá porovnávání všech snímků mezi sebou. Dalším důvodem je také to, že nástroj Bundler provádí rekonstrukci scény iterativně, což znamená, že vytvoří vždy část scény, do které se snaží přidávat další body. V praxi se proto ukázalo výhodnější kvůli časové úspoře rozdělovat větší scény na několik menších částí.

Pro dosažení co nejlepších výsledků rekonstrukce, na které je odhadování vzdáleností založeno, je ideální, pokud je scéna snímána z většího rozsahu úhlů pohledu. U záznamu z vozidlové kamery je z principu rozsah těchto úhlů pohledu velmi omezený. I přes všechna další omezení využitých metod lze však získané výsledky označit jako obstojné. Průměrná chyba měření během testování se pohybovala v rozsahu od **0,125 m** do **0,306 m**. Chyba měření byla způsobena nepřesnou 3D rekonstrukcí a také někdy omezenou schopností uživatele kalibrovat scénu. K nepřesnosti také přispěl fakt, že body často dokonale nepasovaly na rozměry, který byly měřeny. V takových případech byly vybrány body, které se alespoň co nejvíce blížily požadované pozici.

Během testování byl odhalen problém s přehráváním videa. Některé záznamy byly přehrávány zpomaleně či zrychleně. Tento jev jsem pozoroval především na testovacích datech. U jiných videí jsem se s tímto nesetkal. Tento problém však nijak neovlivňuje odhadování vzdáleností ve scéně.

7.3 Možná vylepšení

Využitelnost aplikace a přesnost odhadu by se potenciálně mohla zvýšit, pokud by byly využity některá z následujících řešení.

Asi největší překážkou pro měření vzdáleností ve scéně je nedostatek bodů 3D modelu, který je způsoben tím, že Bundler vytváří pouze řídký point cloud. Tento problém by se dal řešit využitím nástroje *PMVS (Patch-based Multi-view Stereo Software)*, který dokáže pomocí výstupu Bundleru vytvořit hustý point cloud. S více body, mezi kterými můžeme odhadovat vzdálenost, by se využitelnost aplikace nepochybně zvýšila.

Řešením, které by částečně mohlo kompenzovat nemožnost měření mimo body point cloudu, by mohlo být proložení vybraných bodů v prostoru rovinou, což by umožňovalo měřit nejen v bodech point cloudu, ale také v této rovině. Toto řešení by mělo využití u objektů s rovnou plochou jako jsou např. vozovka, svislé dopravní značení, billboardy atp. V některých případech by pomohlo proložení bodů i pouhou přímkou, které by poté umožňovalo měření na této přímce.

Dalším možným vylepšením by mohlo být rozpoznávání neostrých snímků, které vznikají z důvodu jízdy automobilu po nerovném povrchu vozovky. Při extrahování snímku z videa by byly poté vybrány pouze ty, které jsou dostatečně ostré. Cílem tohoto vylepšení by bylo opět zvýšit kvalitu výstupu rekonstrukce.

Ke zlepšení přesnosti by určitě přispělo, kdyby byly na vozidle umístěny dvě vozidlové kamery. Snímky z dvojice kamer by potenciálně mohly zlepšit výsledky 3D rekonstrukce a tím i konečného měření. Když se ale zamyslíme nad tímto řešením v praxi, jeví se jako velmi nepravděpodobné, že by si běžný uživatel pořídil a používal dvě kamery jenom z důvodu potenciálního využití záznamu pro odhadování vzdáleností.

Kapitola 8

Závěr

Tématem mé bakalářské práce bylo *Automatické zpracování záznamu z vozidlové kamery*. V práci jsem se rozhodl věnovat možnosti využití tohoto záznamu pro odhadování vzdálenosti ve scéně. Mým cílem bylo vytvořit aplikaci, která bude toto odhadování umožňovat a jako prostředek pro měření bude využívat částečnou 3D rekonstrukci scény.

Výstupem mé práce je aplikace, která tyto cíle splňuje. Za pomoci externích nástrojů vytvoří částečný model scény reprezentovaný mračnem bodů, který vizualizuje ve svém grafickém rozhraní. Po kalibraci scény, kdy uživatel označí ve scéně několik mu známých rozměrů, které se využívají pro přepočítání z prostorových vzdáleností do metrických, je uživatel schopen v této aplikaci měřit vzdálenosti přímo v 3D modelu scény nebo ve snímcích, ve kterých jsou vyznačeny význačné body využité při rekonstrukci.

Dosažené výsledky měření lze vzhledem k podmínkám, které nebyly pro 3D rekonstrukci přímo ideální, označit jako obstojné. Přesnost měření kolísala napříč testovacími scénami, přičemž bylo dosaženo lepších výsledků v prostorově menších scénách městského typu. Nepoužitelné se ukázaly scény s velmi malou členitostí, jakými jsou například dálnice. Jako největší omezení měření se poté jeví relativně malý počet bodů, mezi kterými je možné odhadovat vzdálenosti.

V porovnání s měřením vzdáleností pomocí radaru nebo případně infračerveného senzoru by přesnost odhadu samozřejmě neobstála. Toto ale nebylo cílem práce. Vybavení automobilu zařízením tohoto typu není ani v dnešní době běžné, proto se nabízí využití právě vozidlové kamery.

Dále se také nabízí srovnání s přístupem, který využívají aplikace v kapitole 3.2. U těchto aplikací je ale nutná přímá spolupráce uživatele při tvorbě modelu scény, který je při využití mého řešení vytvořen automaticky, pouze na základě vstupních dat a parametrů zadaných uživatelem. Model, který je vytvořen těmito aplikacemi, se ve výsledku skládá pouze z rovin, a proto postrádá výhody plastičnosti klasického 3D modelu.

Přesnost a možnosti měření by se daly ještě rozšířit pomocí několika vylepšení. Jako jedno z nejzajímavějších řešení, které se nabízí, je využití hustého mračna bodů. Toto mračno by bylo získáno za pomoci nástroje PMVS, který jako vstup využívá výstup z nástroje Bundler. Husté mračno by potom výrazně mohlo rozšířit možnosti aplikace, protože by se zvýšil počet bodů, mezi kterými by bylo možné odhadovat vzdálenosti.

Pro podpoření kvalitnějšího výstupu rekonstrukce by mohlo být v aplikaci implementováno rozpoznávání neostrých snímků, které by se automaticky vyřazovaly z dalšího zpracování.

Literatura

- [1] OpenCV Wiki. [online], [cit. 2012-04-21].
URL <http://opencv.willowgarage.com/wiki/>
- [2] PCL - Point Cloud Library. [online], [cit. 2012-04-21].
URL <http://pointclouds.org/>
- [3] Qt Licensing. [online], [cit. 2012-04-22].
URL <http://qt.nokia.com/products/licensing>
- [4] Building Rome in a Day. [online], [cit. 2012-04-24].
URL <http://grail.cs.washington.edu/rome/>
- [5] Eltrinex CarHD. [online], [cit. 2012-04-30].
URL <http://www.eltrinex.cz/detail.php?pr=15>
- [6] Kamera do auta FULLHDCAM5. [online], [cit. 2012-04-30].
URL <http://www.onboard.cz/kamera-do-auta/FullHD-Kamery/kamera-do-auta-terna-skrinka-fullhdcam5>
- [7] PC Crash. [online], [cit. 2012-04-30].
URL <http://www.pccrash.cz/>
- [8] PC-RECT A Photograph Rectification Program. [online], [cit. 2012-04-30].
URL http://www.pccrash.co.kr/include/filedown.asp?filename=PC-RECT_4.1.pdf
- [9] Virtual CRASH. [online], [cit. 2012-04-30].
URL <http://www.vcrash3.com>
- [10] Camera calibration With OpenCV. [online], [cit. 2012-05-01].
URL http://opencv.itseez.com/trunk/doc/tutorials/calib3d/camera_calibration/camera_calibration.html#cameracalibrationopencv
- [11] Agarwal, S.; Furukawa, Y.; Snavely, N.; et al.: Building Rome in a day. *Commun. ACM*, ročník 54, č. 10, Říjen 2011: s. 105–112, ISSN 0001-0782.
URL <http://doi.acm.org/10.1145/2001269.2001293>
- [12] Furukawa, Y.: Clustering Views for Multi-view Stereo. [online], [cit. 2012-04-24].
URL <http://grail.cs.washington.edu/software/cmvs/>
- [13] Fusiello, A.: Elements of Geometric Computer Vision. [online], [cit. 2012-04-17].
URL http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial.html

- [14] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, druhé vydání, 2003, ISBN 0-0521-54051-8.
- [15] Kršek, P.: Základy počítačové grafiky IZG, Studijní opora. dostupné pro zapsané v předmětu IZG na FIT VUT v Brně.
- [16] Lowe, D.: Demo Software: SIFT Keypoint Detector. [online], [cit. 2012-04-21].
URL <http://www.cs.ubc.ca/~lowe/keypoints/>
- [17] Snavely, N.: Bundler: Structure from Motion (SfM) for Unordered Image Collections. [online], [cit. 2012-04-22].
URL <http://phototour.cs.washington.edu/bundler/>
- [18] Snavely, N.; Seitz, S. M.; Szeliski, R.: Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph.*, ročník 25, č. 3, Červenec 2006: s. 835–846, ISSN 0730-0301.
URL <http://doi.acm.org/10.1145/1141911.1141964>
- [19] Snavely, N.; Seitz, S. M.; Szeliski, R.; aj.: Photo Tourism. [online], 2006 [cit. 2012-04-22].
URL <http://phototour.cs.washington.edu/>
- [20] Velebová, J.: *Fotogrammetrická analýza obrazů*. Diplomová práce, FIT VUT v Brně, Brno, 2011.
- [21] Říha, K.: Epipolární geometrie. [online], 2005-03-21 [cit. 2012-04-14].
URL <http://www.elektrorevue.cz/clanky/05017/index.html>

Příloha A

Obsah DVD

- technická zpráva
- plakát
- zdrojové kódy
 - kalibrační aplikace
 - hlavní aplikace
- binární soubory externích aplikací
 - sift
 - KeyMatchFull
 - bundler
- sdílené knihovny vyžadované externími aplikacemi
- testovací data
 - snímky využité pro kalibraci kamery
 - záznamy z kamery
 - výstupy aplikace – zrekonstruovaná scéna a význačné body

Příloha B

Plakát



Fakulta informačních technologií
Vysokého učení technického v Brně

Automatické zpracování záznamu z vozidlové kamery Bakalářská práce 2012

Práce se zabývá odhadováním vzdáleností v záznamu pořízeném vozidlovou kamerou. Odhad probíhá díky částečnému 3D modelu scény, který je vytvořen pomocí externích nástrojů ze snímků ze záznamu. Tento model se skládá pouze z význačných bodů scény a je reprezentován jejich mračenem.



Aplikace, která byla vytvořena v rámci mé práce dokáže vizualizovat model scény a zobrazovat k němu snímky s body, ze kterých tento model vznikl. Do obou těchto zobrazení lze klikat a tak určovat, mezi kterými body bude vzdálenost odhadována. Zobrazení jsou mezi sebou provázána a určování bodů v jednom z nich se bude projevat i v druhém.



Ke správné funkci aplikace je nutné, aby uživatel určil ve scéně nějaký jemu známý rozměr. Zde se nabízí využití znalostí rozměrů dopravních značek, přechodů pro chodce, billboardů, ostatních automobilů atp.

Autor: Dominik Klos
Vedoucí: Ing. Michal Španěl, Ph.D.