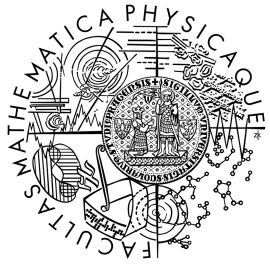
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Beata Turoňová

Rekonstrukce 3D scény ze stereo obrázků

Katedra softwarového inženýrství

Vedoucí bakalářské práce: Mgr. Viliam Holub, Ph.D.
Studijní program: Informatika, Obecná informatika

2008

Ráda bych touto cestou poděkovala Mgr. Viliamu Holubovi, Ph.D. za odborné vedení a cenné rady při tvorbě této práce.

Prohlašuji, že jsem svou bakalářskou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 28.5.2008 Beata Turoňová

Obsah

1 Úvod **6**

2 Projektivní rekonstrukce **8**
2.1 Projektivní prostor a epipolární geometrie 8
2.2 Fundamentální matice 10
2.3 Matice fotoaparátu 13

3 Metrická rekonstrukce **16**
3.1 Esenciální matice 16
3.2 Postupné zdokonalování 18
3.3 Přímá rekonstrukce 20
3.4 Znalost umístění bodů ve 3D 20

4 Nalezení 3D struktury **22**
4.1 Optimální metoda 23
4.2 Lineární metody 25

5 Rekonstrukce povrchu **27**
5.1 Přehled algoritmů 27
5.2 Algoritmus využívající vnitřních vlastností 29

6 Implementace **32**
6.1 Výběr algoritmů 32
6.2 Struktura programu 33
6.3 Vstupní a výstupní data 34
6.4 Výsledné modely 36
6.5 Použité softwarové nástroje a knihovny 37

7 Závěr **38**
7.1 Zhodnocení 38
7.2 Možná vylepšení 39

A Uživatelská dokumentace **40**
A.1 Systémové požadavky 40
A.2 Popis obsahu CD 40
A.3 Pořizování fotografií 41
A.4 Získání shodných bodů 41
A.5 Vstupní data 42
A.6 Výstupní data 44
A.7 Uživatelské rozhraní 45

Literatura **47**

Název práce: Rekonstrukce 3D scény ze stereo obrázků
Autor: Beata Turoňová
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: Mgr. Viliam Holub, Ph.D.
e-mail vedoucího: holub@dsrg.mff.cuni.cz

Abstrakt: V předložené práci se zabýváme problematikou rekonstrukce 3D scény ze stereo fotografií. Nejprve podrobněji rozebereme způsob, jakým lze ze shodných bodů nalézt fundamentální matici a s její pomocí spočítat projektivní souřadnice 3D modelu. Následně se budeme zabývat metodami, které nám umožní vylepšit projektivní model na metrický, přičemž důraz bude kladen zejména na metodu využívající esenciální matici. Dále budeme zkoumat možnosti rekonstrukce povrchu z množiny bodů a podrobněji popíšeme algoritmus založený na vnitřních vlastnostech množiny bodů. Na závěr lehce naznačíme strukturu programu a uvedeme výsledky, kterých dosáhl.

Klíčová slova: 3D rekonstrukce, fundamentální matice, esenciální matice, rekonstrukce povrchu

Title: 3D Scene Reconstruction from Stereo Pictures
Author: Beata Turoňová
Department: Department of Software Engineering
Supervisor: Mgr. Viliam Holub, Ph.D.
Supervisor's e-mail address: holub@dsrg.mff.cuni.cz

Abstract: We study 3D scene reconstruction from stereo pictures. First, we describe methods for computing fundamental matrix from a set of corresponding image points. Using this matrix we can obtain a projective 3D reconstruction. The second part deals with techniques of refining the projective reconstruction to metric one. We focus especially on that one, which uses an essential matrix. Furthermore we discuss the possibilities of surface reconstruction from a given point cloud. The last part includes some details of an implementation and presents its results.

Keywords: 3D reconstruction, fundamental matrix, essential matrix, surface reconstruction

Kapitola 1

Úvod

Rekonstrukce 3D scény z fotografií je oblastí počítačového vidění, které bylo v posledních letech věnováno hodně pozornosti. Přesto zatím nebylo dosaženo dostatečně uspokojivých výsledků.

Existují různé přístupy k rekonstrukci lišíce se v tom, kolik fotografií je k dispozici. Tato práce se bude zabývat případem, kdy vstupní data tvoří dvě fotografie (tzv. stereo).

Převážná část se věnuje teoretické stránce této problematiky, přičemž rozděluje celý proces rekonstrukce na několik kroků. U každého z nich se snaží vysvětlit všechny podstatné souvislosti a popsat nejznámější metody, které s daným krokem souvisí. Uvádí výhody a nevýhody jednotlivých postupů. Podrobně se pak věnuje zejména těm metodám, které byly implementovány v programu.

Závěr práce je věnován implementaci. Naznačuje strukturu programu, odůvodňuje výběr použitých algoritmů a představuje 3D modely vytvořené programem. Shrnuje také poznatky spojené s testováním programu na různých datech a navrhuje jeho možná vylepšení.

Použité značení

Pro lepší orientaci ve významu používaných symbolů zavedeme následující značení.

Bod, který se nachází na první fotografii, budeme značit $\mathbf{x}$. Pokud bude reprezentován homogenními souřadnicemi, bude platit $\mathbf{x} = (x, y, 1)^T$. V případě nehomogenních souřadnic je $\mathbf{x} = (x, y)^T$. Bod na druhé fotografii, který odpovídá bodu $\mathbf{x}$, značíme $\mathbf{x}'$. Vztah těchto dvou bodů budeme také vyjad-

řovat jako $\mathbf{x} \leftrightarrow \mathbf{x}'$.

Bod ve 3D prostoru značíme $\mathbf{X}$ a jeho souřadnice jsou X, Y, Z.

Je-li $\mathbf{a} = (a_1, a_2, a_3)^T$ vektor, pak symbol $[\mathbf{a}]_\times$ označuje jemu odpovídající antisymetrickou matici

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Kapitola 2

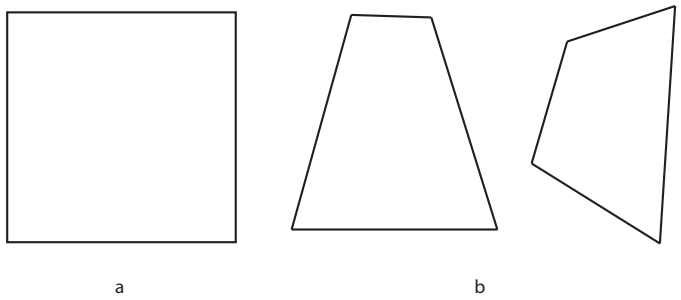
Projektivní rekonstrukce

Vzájemný vztah mezi fotografiemi a na nich zobrazenou scénou hraje klíčovou roli při rekonstrukci 3D modelu. Nejprve se tedy budeme zabývat epipolární geometrií, která tento vztah určuje. Následně se budeme věnovat fundamentální matici, jež je algebraickým vyjádřením epipolární geometrie. Popíšeme vlastnosti této matice a způsoby, jak ji můžeme získat. Dále uvedeme, jak s pomocí fundamentálni matice můžeme nalézt matice fotoaparátu.

2.1 Projektivní prostor a epipolární geometrie

Projektivní prostor

V euklidovském prostoru se každé dvě přímky protínají v právě jednom bodě. Výjimku tvoří rovnoběžky, které se neprotínají nikde. Projektivní prostor P^n je vhodným rozšířením prostoru euklidovského o množinu dalších bodů. Tyto body se nazývají *body v nekonečnu*. Všechny navzájem rovnoběžné přímky se protnou v jednom bodě v nekonečnu. Platí také, že všechny body v nekonečnu tvoří v P^2 *přímku v nekonečnu* a v P^3 *rovinu v nekonečnu*. Pro projektivní prostor je tedy pravdou, že každé dvě přímky se protínají v právě jednom bodě a že každými dvěma body prochází právě jedna přímka. Projektivní prostor nezachovává tvar, délku, vzdálenosti, poměry mezi vzdálenostmi ani úhly. Rozdíl mezi euklidovským a projektivním prostorem znázorňuje obrázek (2.1).



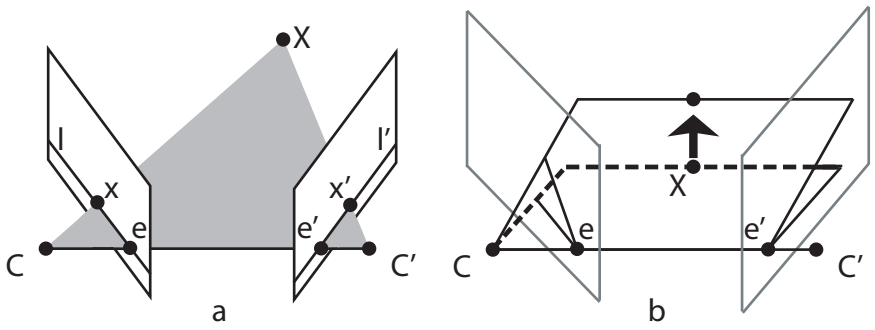
Obrázek 2.1: (a) Zobrazuje čtverec v euklidovském prostoru. (b) Tentýž útvar, ale v projektivním prostoru.

Epipolární geometrie

Epipolární geometrie je vnitřní projektivní geometrie, která se zabývá určením vztahu mezi dvěma fotografiemi téže scény. Jinými slovy jde o geometrii dvou středových promítání. Není závislá na struktuře scény, závisí pouze na vnitřních parametrech fotoaparátů a jejich vzájemné pozici. Označme si $\mathbf{C}$ a $\mathbf{C}'$ středy fotoaparátu a jako obrazové roviny označme roviny, v nichž leží fotografie. Buď $\mathbf{X}$ bod v prostoru a $\mathbf{x}$ a $\mathbf{x}'$ jemu odpovídající body na snímcích. Mezi klíčové pojmy epipolární geometrie patří

- **Epipól** - průsečík přímky procházející $\mathbf{C}$ a $\mathbf{C}'$ s obrazovou rovinou. Označuje se $\mathbf{e}$.
- **Epipolární rovina** - rovina obsahující spojnici $\mathbf{C}$ a $\mathbf{C}'$.
- **Epipolární přímka (epipolára)** - průsečnice epipolární roviny s rovinou obrazovou. Značí se $\mathbf{l}$. Všechny epipolární přímky se protínají v epipólu.

Epipolární geometrie se využívá také k hledání shodných bodů na fotografiích. Body $\mathbf{X}$, $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{C}$ a $\mathbf{C}'$ leží v jedné epipolární rovině π. Je zřejmé, že přímky spojující $\mathbf{X}$ s $\mathbf{x}$ a $\mathbf{X}$ s $\mathbf{x}'$ leží také v rovině π. Známe-li $\mathbf{x}$, $\mathbf{C}$ a $\mathbf{C}'$, můžeme určit π a následně $\mathbf{l}$. Víme, že hledaný bod $\mathbf{x}'$ musí ležet také v rovině π. Zároveň musí ležet v obrazové rovině. Je tedy jasné, že $\mathbf{x}'$ musíme hledat na $\mathbf{l}'$. Situaci znázorňuje obrázek (2.2, a).



Obrázek 2.2: (a) Epipolární geometrie. (b) Epipolární svazek. Inspirováno nákresy [8, str. 240].

S měnícím se $\mathbf{X}$ rotuje epipolární rovina kolem spojnice $\mathbf{C}$ a $\mathbf{C}'$. Množina těchto rovin se nazýva epipolární svazek a můžeme jej vidět na obrázku (2.2, b).

2.2 Fundamentální matice

Fundamentální matice F je algebraickým popisem epipolární geometrie. Jde o matici velikosti 3x3 s hodností 2. Vztah mezi epipóly a fundamentální maticí je dán rovnicemi $F\mathbf{e} = 0$ a $F^T\mathbf{e}' = 0$. Pro epipolární přímky a fundamentální matici pak platí $\mathbf{l} = F^T\mathbf{x}'$ a $\mathbf{l}' = F\mathbf{x}$. Jestliže je $\mathbf{X}$ bod v prostoru a $\mathbf{x} \leftrightarrow \mathbf{x}'$ je dvojice jemu odpovídajících shodných bodů, potom platí vztah

$$\mathbf{x}'^T F\mathbf{x} = 0. \tag{2.1}$$

Z tohoto vztahu je zřejmé, že přestože F nezávísí na struktuře scény, můžeme ji spočítat z pouhé znalosti dostatečného počtu shodných bodů. Způsob, jakým lze fundamentální matici nalézt, byl poprvé popsán Longuet-Higginsem [11] a jednalo se o kalibrovanou matici. Několik metod, jak získat F z nekalibrovaných dat, navrhl Zhang [16].

Buď $\mathbf{x} = (x, y, 1)^T$ a $\mathbf{x}' = (x', y', 1)^T$ dvojice shodných bodů. Pro každý takovýto pár bodů máme jednu lineární rovnici, kde neznámými jsou prvky matice F

$$x'xF_{11} + x'yF_{12} + x'F_{13} + y'xF_{21} + y'yF_{22} + y'F_{23} + xF_{31} + yF_{32} + F_{33} = 0.$$

Vyjádříme-li prvky F jako vektor $\mathbf{f}$, můžeme také psát

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1)\,\mathbf{f} = 0.$$

Pro n shodných bodů máme tedy soustavu n lineárních rovnic o devíti neznámých, což můžeme vyjádřit

$$A\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} = 0. \qquad (2.2)$$

Aby měla soustava (2.2) jedinečné řešení, je třeba znát minimálně 8 dvojic shodných bodů. Existují sice metody, jak vyřešit (2.2) i pro 7 nebo dokonce jen 6 dvojic, avšak v obou případech má A hodnost menší než 8 a řešení není jednoznačné. Proto se zaměříme pouze na případ, kdy známe alepsoň 8 párů schodných bodů. K získání F použijeme *normalizovaný 8-mi bodový algoritmus*.

Nevýhodou klasického 8-mi bodového algoritmu je jeho závislost na souřadnicovém systému, ve kterém jsou reprezentovány vstupní body. Závislost algoritmu na vstupních datech byla dokázána [6] a také bylo ukázáno, že některé souřadnicové systémy dávají výrazně lepší výsledky než jiné. Vhodnou normalizací shodných bodů se dá zajistit, aby body v libovolném souřadnicovém systému byly převedeny do kanonického systému souřadnic, jehož použití dává nejlepší výsledky.

Prvním krokem normalizace je posunutí bodů tak, aby jejich těžiště bylo v počátku souřadnicového systému. Následně se x-ová i y-ová souřadnice každého bodu vynásobí stejným koeficientem, aby průměrná vzdálenost od počátku byla rovna $\sqrt{2}$. Tyto úpravy souřadnic, které provádíme pro každou fotografii zvlášť, lze popsat transformačními maticemi T pro první a T' pro druhou fotografii

$$T = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix}, T' = \begin{bmatrix} s' & 0 & t'_x \\ 0 & s' & t'_y \\ 0 & 0 & 1 \end{bmatrix},$$

kde s je koeficient, kterým násobíme, a t_x,t_y je posunutí ve směru osy x a y. Normalizované body $\hat{\mathbf{x}}_\mathbf{i}$ a $\hat{\mathbf{x}}'_\mathbf{i}$ tedy získáme

$$\hat{\mathbf{x}}_\mathbf{i} = T\mathbf{x}_\mathbf{i},\ \hat{\mathbf{x}}'_\mathbf{i} = T'\mathbf{x}'_\mathbf{i}.$$

Nyní můžeme nalézt $\hat{F}$ jako řešení soustavy lineárních rovnic $\hat{\mathbf{x}}'^T \hat{F} \hat{\mathbf{x}} = 0$. Tuto rovnici si přepíšeme do tvaru $\hat{A}\hat{\mathbf{f}} = 0$. Matice $\hat{A}$ by měla mít hodnost 8. Většinou se však stává, že má hodnost 9. Způsobeno to bývá přítomností šumu na fotografiích, což má za následek nepřesné souřadnice shodných bodů. V takovém případě lze najít řešení metodou nejmenších čtverců. Buď $\hat{A} = UDV^T$ rozklad na vlastní čísla, pak sloupec matice V, který odpovídá nejmenšímu vlastnímu číslu v D, je hledaný vektor $\hat{\mathbf{f}}$. Použitím metody nejmenších čtverců k získání $\hat{\mathbf{f}}$ bude navíc zajištěna minimalizace $\left\|\hat{A}\hat{\mathbf{f}}\right\|$ tak, že $\left\|\hat{\mathbf{f}}\right\| = \hat{\mathbf{f}}^T\hat{\mathbf{f}} = 1$.

Důležitou vlastností fundamentální matice je její singulárnost a hodnost 2. Pouhým vyřešením soustavy rovnic většinou nezískáme matici s touto hodností. Můžeme však získanou $\hat{F}$ nahradit maticí $\tilde{F}$, která již bude mít požadovanou hodnost 2. Matice $\tilde{F}$ navíc minimalizuje Frobeniovu normu $\left\|\hat{F} - \tilde{F}\right\|$ tak, že bude platit $det\tilde{F} = 0$. Matici $\tilde{F}$ nalezneme opět pomocí rozkladu na vlastní čísla. Buď $\tilde{F} = UDV^T$ zmíněný rozklad, kde v D jsou vlastní čísla seřazena v nerostoucím pořadí. Nahradíme-li nejmenší vlastní číslo v D nulou a označíme-li pozměněnou matici $\tilde{D}$, pak $\tilde{F} = U\tilde{D}V^T$. Tuto metodu navrhli Tsai a Huang [15].

Posledním krokem algoritmu je denormalizace nalezené matice $\tilde{F}$. Položíme $F = T'^T\tilde{F}T$. Fundamentální matice F odpovídá shodným bodům $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$.

Normalizovaný 8-mi bodový algoritmus je nejjednodušší metodou, jak získat fundamentální matici z minimálně osmi shodných bodů. Existují další způsoby, z nichž většina v některé své fázi využívá 8-mi bodového algoritmu. Můžeme je rozdělit do dvou skupin - na algoritmy, které minimalizují algebraickou chybu, a na ty, které minimalizují geometrickou vzdálenost na fotografiích. Často doporučovanou metodou je *Gold Standard*, která spadá do druhé skupiny. Tento algoritmus odhaduje fundamentální matici metodou maximální věrohodnosti a je založen na předpokladu, že chyby ve vstupních bodech mají Gaussovu distribuci.

Nejdříve pomocí normalizovaného 8-mi bodového algoritmu nalezneme

první odhad fundamentální matice, pak spočítáme matice fotoaparátů P a P' a 3D souřadnice $\mathbf{X_i}$ (těmto krokům se budeme blíže věnovat později). Mělo by platit, že $\mathbf{x_i} = P\mathbf{X_i}$ a $\mathbf{x'_i} = P'\mathbf{X_i}$. Přítomnost šumu ve fotografiích však způsobuje, že vynásobením 3D souřadnic maticema fotoaparátů nezískáme původní $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$, ale trochu odlišné $\bar{\mathbf{x}}_\mathbf{i} \leftrightarrow \bar{\mathbf{x}}'_\mathbf{i}$. Označíme-li vzdálenost mezi dvěma body $d()$, pak *Gold Standard* se snaží pomocí Lavenberg-Marquardtova algoritmu minimalizovat

$$\sum_i d(\mathbf{x_i}, \bar{\mathbf{x}}_\mathbf{i})^2 + d(\mathbf{x'_i}, \bar{\mathbf{x}}'_\mathbf{i})^2.$$

Tento algoritmus je náročný na implementaci, ale dává jedny z nejlepších výsledků. Ovšem jak Hartley [6] ukázal, i použitím samotného normalizovaného 8-mi bodového algoritmu můžeme získat uspokojivé výsledky.

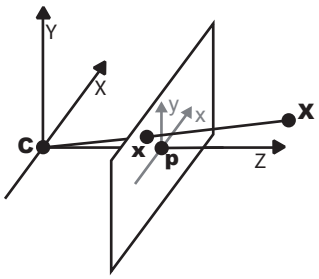
2.3 Matice fotoaparátu

K získání 3D souřadnic $\mathbf{X_i}$ ze znalosti shodných bodů $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$, je potřeba znát vztah mezi body v prostoru a jejich promítnutím do roviny. Právě tento vztah nám popisuje matice fotoaparátu P. Pro lepší pochopení jednotlivých částí P je dobré znát jednoduchý matematický model fotoaparátu.

Mějme optický střed fotoaparátu $\mathbf{C}$ umístěný v počátku soustavy souřadnic. Fotografie, na kterou se bude 3D scéna promítat, leží v obrazové rovině rovnoběžné s rovinou danou osami x a y. Průsečík obrazové roviny s osou z označíme $\mathbf{p}$. Vzdálenost mezi $\mathbf{C}$ a $\mathbf{p}$ se nazývá ohnisková a značí se f. Bod v prostoru $\mathbf{X}$ se na fotografii zobrazí tam, kde obrazovou rovinu protne přímka spojující $\mathbf{X}$ a $\mathbf{C}$. Tuto situaci znázorňuje obrázek (2.3). Z téhož obrázku je snadno vidět, že bod $\mathbf{X} = (X, Y, Z)^T$ se v obrazové rovině zobrazí na bod $\mathbf{x} = (fX/Z, fY/Z)^T$. Jde o mapování z euklidovského prostoru R^3 do euklidovského prostoru R^2. Pokud použijeme homogenní souřadnice získáme

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \qquad (2.3)$$

Je-li tedy prostorový bod $\mathbf{X}$ i jeho obraz v rovině $\mathbf{x}$ dán homogenními souřadnicemi, pak vztah mezi nimi lze popsat pomocí projekční matice fo-



Obrázek 2.3: Matematický model fotoaparátu

toaparátu P o velikosti 3x4 takto

$$\mathbf{x} = P\mathbf{X}. \tag{2.4}$$

Matice P ve tvaru uvedeném v (2.3) odpovídá situaci, kdy souřadnice na fotografii mají počátek soustavy souřadnic v $\mathbf{p}$. Většinou ovšem bývá zvykem, mít počátek v levém horním, nebo dolním rohu fotografie. Je tedy nutné zahrnout do P i příslušný posun o p_x ve směru osy x a o a p_y ve směru osy y. Hodnoty p_x a p_y odpovídají souřadnicím $\mathbf{p}$. Po této úpravě bude P vypadat následovně

$$P = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} I|0 \end{bmatrix} = K \begin{bmatrix} I|0 \end{bmatrix}. \tag{2.5}$$

Matice K se nazývá kalibrační matice a udává vnitřní parametry fotoaparátu. Může se stát, že některé fotoaparáty nemají stejné zvětšení ve směru osy x a ve směru osy y. Jinými slovy pixely nejsou čtvercové. Pokud jsou v takovém případě vstupní souřadnice vztahující se k fotografii dány v pixelech, dochází k nepřesnostem. Aby se tomuto nežádoucímu jevu předešlo, upraví se kalibrační matice

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix},$$

kde $\alpha_x = m_x f$, $\alpha_y = m_y f$, $x_0 = m_x p_x$ a $y_0 = m_y p_y$ (m_x a m_y udávají počet pixelů na jednotku vzdálenosti ve směru osy x a y).

Rovnice (2.5) se vztahuje k situaci, kde bod $\mathbf{X}$ je reprezentován stejným souřadnicovým systémem jako fotoaparát. Ve skutečnosti se však oba souřadnicové systémy liší, protože $\mathbf{X}$ bývá vyjádřen ve světových souřadnicích. Rozdíl mezi oběma systémy je tvořen posunem a rotací. Matice rotace R udává orientaci souřadnicového systému fotoaparátu a $\tilde{C}$ pozici jeho optického středu ve světovém systému souřadnic. R a $\tilde{C}$ jsou tedy vnějšími parametry fotoaparátu. Platí

$$\mathbf{x} = KR \left[\; I| - \tilde{C} \; \right] X.$$

Často se optický střed neudává přesně a namísto $P = KR \left[\; I| - \tilde{C} \; \right]$ se zavádí

$$P = K \left[\; R|\mathbf{t} \; \right],$$

kde $\mathbf{t} = -R\tilde{C}$.

Matice fotoaparátu závisí jak na vnitřních, tak i na vnějších parametrech. Fundamentální matice F nezávisí na volbě souřadnicového systému a při projektivních transformacích 3D prostoru zůstává beze změny. Takže dvě projektivní matice P a P' určují jednoznačně F, ale opačně to neplatí. S ohledem na tuto projektivní nejednoznačnost, můžeme z F určit matice fotoaparátu následovně

$$P = \left[\; I|0 \; \right] \text{ a } P' = \left[\; SF|\mathbf{e}' \; \right],$$

kde S je antisymetrická matice a $\mathbf{e}'$ je epipól na druhé fotografii. Matici S můžeme nahradit $[\mathbf{e}']_\times$, jak bylo navrženo Luongem [12]. Matice fotoaparátu pak budou mít následující tvar

$$P = \left[\; I|0 \; \right]$$

$$P' = \left[\; [\mathbf{e}']_\times F|\mathbf{e}' \; \right].$$

Je více způsobů, jak získat P a P', v závislosti na tom, jaká data máme k dispozici. Kromě postupu uvedeného v této podkapitole, je pro nás ještě důležitá metoda, která matice fotoaparátu získává ze znalosti esenciální matice. Touto metodou se budeme zabývat v (3.1).

Kapitola 3

Metrická rekonstrukce

Člověk je zvyklý na euklidovský prostor, a tak na něj projektivní model působí často velmi nepřirozeně. Mnohem lepší by tedy bylo získat model metrický. K tomu však nestačí pouhá znalost shodných bodů, potřebujeme mít další informace o scéně nebo znát kalibrační data použitého fotoaparátu.

3.1 Esenciální matice

Jednou z metod, jak získat metrickou strukturu, je nalezení esenciální matice E. Jde o speciální druh fundamentální matice, který vyžaduje znalost interních kalibračních dat. Tato matice byla poprvé popsána Longuet-Higginsem [11] (a to dřív než matice fundamentální, kterou můžeme považovat za její zobecnění nevyžadující znalost kalibračních dat).

Následující vlastnosti E a postupy nutné k získání metrického modelu byly popsány Longuet-Higginsem [11], Huangem a Faugerasem [9].

V (2.3) jsme popsali matici fotoaparátu mimo jiné pomocí rovnice

$$P = K \left[\; R|\mathbf{t} \; \right].$$

Vztah mezi promítnutým bodem $\mathbf{x}$ a jeho $\mathbf{X}$ je dán rovnicí $\mathbf{x} = P\mathbf{X}$. Známe-li kalibrační matici K, můžeme tento vztah přepsat následovně

$$\hat{\mathbf{x}} = \left[\; R|\mathbf{t} \; \right] \mathbf{X},$$

kde $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$ je normalizovaný bod $\mathbf{x}$. Obdobně $K^{-1}P = \left[\ R|\mathbf{t}\ \right]$ nazveme normalizovanou maticí fotoaparátu.

Máme-li dvojici takovýchto normalizovaných matic $P = \left[\ I|0\ \right]$ a $P' = \left[\ R|\mathbf{t}\ \right]$, pak jim odpovídající matici nazýváme esenciální a popsat jí můžeme takto

$$E = [\mathbf{t}]_\times R = R\left[\ R^T\mathbf{t}\ \right]_\times .$$

Podobně jako F je i E definována rovnicí

$$\hat{\mathbf{x}}'^T E\hat{\mathbf{x}} = 0, \tag{3.1}$$

kde $\hat{\mathbf{x}} \leftrightarrow \hat{\mathbf{x}'}$ jsou normalizované body odpovídající dvojici $\mathbf{x} \leftrightarrow \mathbf{x}'$. Dosazením nenormalizovaných bodů do (3.1) získáme $\mathbf{x}'^T K'^{-T} E K^{-1}\mathbf{x} = 0$.

Protože pro fundamentální matici F platí $\mathbf{x}'^T F\mathbf{x} = 0$, můžeme vztah mezi F a E vyjádřit

$$E = K'^T F K. \tag{3.2}$$

Existují tedy dva způsoby, jak E získat:

1. Spočítáme normalizovaným 8-mi bodovým algoritmem F a pak použijeme K a (3.2) k získání E.

2. S pomocí K normalizujeme $\mathbf{x} \leftrightarrow \mathbf{x}'$ a soustavu rovnic (3.1) vyřešíme opět metodou nejmenších čtverců.

Matice 3x3 je esenciální právě tehdy, když dvě její vlastní čísla jsou si rovna a třetí je nulové. K docílení této podmínky můžeme opět použít rozklad na vlastní čísla. Buď $E = UDV^T$, kde D má na diagonále vlastní čísla α_1, α_2 a α_3, pro která platí $\alpha_1 \geq \alpha_2 \geq \alpha_3$. Pak $\hat{E}$ (nejbližší matice E ve Frobeniově normě) se rovná součinu $U\hat{D}V^T$. Diagonálními prvky $\hat{D}$ jsou $(\alpha_1 + \alpha_2)/2$, $(\alpha_1 + \alpha_2)/2$, 0.

Máme-li matici E, můžeme z ní získat matice fotoaparátu P a P'. Tyto matice nejsou dány jednoznačně - použitím následující metody získáme čtyři možná řešení, z nichž však jen jedno je vyhovující. Buď $E = UDV^T$ rozklad E na vlastní čísla. Prvky D jsou 1, 1, 0. Jak bylo dokázáno [8, str. 258], existují dvě různé faktorizace $E = SR$

$$S = UZU^T$$

$$R = UWV^T \text{ nebo } R = UW^TV^T,$$

kde

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Je-li $E = U\text{diag}(1, 1, 0)V^T$ a $P = \left[\ I|0\ \right]$, pak existují čtyři možnosti jak získat P':

$$P' = \left[\ UWV^T| + \mathbf{u_3}\ \right], P' = \left[\ UWV^T| - \mathbf{u_3}\ \right],$$

$$P' = \left[\ UW^TV^T| + \mathbf{u_3}\ \right], P' = \left[\ UW^TV^T| - \mathbf{u_3}\ \right].$$

Geometrický význam těchto čtyř řešení je patrný z obrázku (3.1). Správné řešení je tedy to, kdy bod $\mathbf{X}$ leží před oběma fotoaparáty. K zjištění, které P' této situaci odpovídá, je nutné provést test s alespoň jedním bodem $\mathbf{X}$.

Jakmile získáme kalibrační data pro určitý fotoaparát, můžeme je použít k získání metrického modelu ze všech dvojic fotografií pořízených daným fotoaparátem. To je obrovskou výhodou oproti následujícím metodám.

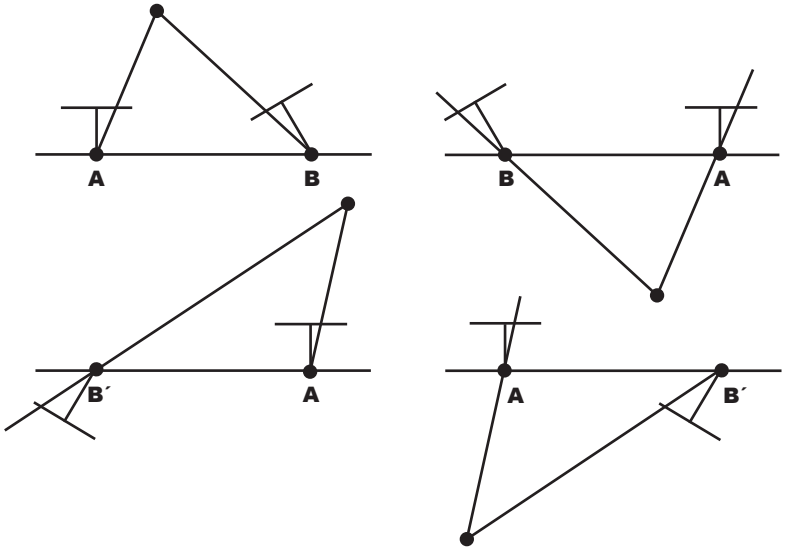
3.2 Postupné zdokonalování

Jednou z metod, jak docílit metrické rekonstrukce, je nejdříve spočítat projektivní model, z něj pak získat afinní a ten ještě zdokonalit na metrický.

Afinní rekonstrukce

Abychom mohli získat z projektivní rekonstrukce, která je dána maticemi fortoaparátu P a P' a množinou 3D bodů $\{\mathbf{X_i}\}$, rekonstrukci afinní, musíme určit rovinu v nekonečnu π. To se nám například podaří, když

- Známe tři body, které leží v rovině v nekonečnu.
- Známe tři dvojice přímek, které jsou ve skutečnosti rovnoběžné.



Obrázek 3.1: Čtyři možná řešení v závislosti na P'. Inspirováno nákresy [8, str. 260].

- Víme, že fotoaparát byl po pořízení prvního snímku pouze posunut, nikoli otočen.

Máme tedy rovinu π a vyjádříme ji jako vektor v souřadnicovém systému projektivní rekonstrukce. Mělo by platit, že π má souřadnice $(0,0,0,1)^T$. Naším cílem je tedy najít projektivní transformaci, která zajistí, aby tomu tak bylo. Jinými slovy hledáme takové H, aby $H^{-1}\pi = (0,0,0,1)^T$. Transformace bude dána

$$H = \begin{bmatrix} I|0 \\ \pi^T \end{bmatrix} .$$

Známe-li H, získáme afinní rekonstrukci následovně

$$P_A = PH^{-1},\ P'_A = P'H^{-1},\ \mathbf{X_{Ai}} = H\mathbf{X_i}.$$

Metrická rekonstrukce

Klíčovým pojmem v této části je *absolutní kuželosečka*. Značit ji budeme ω. Jde o kuželosečku, která se nachází v rovině v nekonečnu $\pi = (0, 0, 0, 1)^T$. Pro body na ω platí

$$\left.\begin{matrix} \mathbf{X_1^2} + \mathbf{X_2^2} + \mathbf{X_3^2} \\ \mathbf{X_4} \end{matrix}\right\} = 0.$$

Řekněme, že se nám podařilo nalézt na jedné z fotografií ω. Matice fotoaparátu odpovídající této fotografii je $P = \left[\begin{array}{c|c} M & \mathbf{m} \end{array}\right]$. Pak afinní rekonstrukce může být vylepšena na metrickou pomocí 3D transformace ve tvaru

$$H = \left[\begin{array}{cc} A^{-1} & \\ & 1 \end{array}\right],$$

kde A získáme Choleského faktorizací z rovnice $AA^T = (M^T \omega M)^{-1}$.

K nalezení ω potřebujeme mít další informace buď o scéně (např. úběžné body a přímky), nebo o fotoaparátech (kalibrační data). Často se k vypočítání ω dostupné informace kombinují.

Metoda postupného vylepšování je nevýhodná především proto, že pro každou scénu, kterou chceme rekonstruovat, musíme znovu hledat π i ω.

3.3 Přímá rekonstrukce

Tato metoda opět využívá znalosti ω. Platí

$$\omega = (KK^T)^{-1}. \tag{3.3}$$

Známe-li ω, můžeme s pomocí Choleského faktorizace získat K. Pokud víme, že obě fotografie byly pořízeny jedním fotoaparátem se stejným nastavením, pak $K = K'$. V opačeném případě, musíme znát i ω', které odpovídá druhé fotografii, a K' získat také faktorizací z (3.3). Když známe obě kalibrační matice, můžeme metrickou rekonstrukci získat pomocí esenciální matice, jak bylo popsáno v (3.1).

3.4 Znalost umístění bodů ve 3D

Další metodou, jak z projektivní rekonstrukce získat přímo rekonstrukci metrickou, je využití znalosti umístění 3D bodů v euklidovském světovém sys-

tému souřadnic. Mezi projektivním modelem a tím, který přesně odpovídá zobrazované scéně, existuje homografie

$$\mathbf{X_{Ei}} = H\mathbf{X_i} \text{ pro i} = 1,\ldots,\text{k}.$$

Mějme projektivní 3D souřadnice $\mathbf{X_i}$ a jim odpovídající body v euklidovském souřadnicovém systému $\mathbf{X_{Ei}}$. Každá dvojice $\mathbf{X_{Ei}} \leftrightarrow \mathbf{X_i}$ dává tři lineárně nezávislé rovnice. Protože matice H má patnáct stupňů volnosti, potřebujeme k jejímu nalezení, aby $k \geq 5$ (a žádné čtyři body neležely v jedné rovině). Pokud spočítáme H, pak metrickou rekonstrukci získáme následovně

$$P_M = PH^{-1},\ P'_M = P'H^{-1},\ \mathbf{X_{Mi}} = H\mathbf{X_i} \text{ pro i} = 1,\ldots,\text{n}.$$

Bohužel, data, která potřebujeme znát k získání metrického modelu, musíme opět získávat pro každou scénu, jejíž rekonstrukci chceme získat. Dalším negativem je fakt, že pro některé typy scén je dost obtížné požadované informace vůbec získat. Naopak pokud jde o rekonstrukci malých snadno změřitelných objektů, může i tato metoda dávat dobré výsledky.

Kapitola 4

Nalezení 3D struktury

Máme dvojice shodných bodů $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$ a jim odpovídající matice fotoaparátu P a P'. Nyní můžeme spočítat 3D souřadnice modelu $\mathbf{X_i}$. Metody, pomocí kterých $\mathbf{X_i}$ získáváme, se nazývají triangulační.

Vhodná triangulační metoda by měla brát ohled na fakt, že vstupní shodné body nebývají přesné (způsobeno to bývá především šumem na fotografiích). To má za následek, že pro většinu dat nelze nalézt bod $\mathbf{X}$, který by splňoval geometrické podmínky

$$\mathbf{x} = P\mathbf{X} \ , \ \mathbf{x}' = P'\mathbf{X}. \tag{4.1}$$

Zrovna tak nebývá splněna epipolární podmínka

$$\mathbf{x'}^T F\mathbf{x} = 0. \tag{4.2}$$

Metoda, kterou budeme rekonstrukci počítat, by také měla být invariantní vůči transformacím toho typu, jakého je rekonstrukce - tzn. pokud počítáme projektivní model, měla by metoda být invariantní vůči projektivním transformacím. Označíme-li způsob, jakým získáme 3D souřadnice τ, pak platí

$$\mathbf{X} = \tau(\mathbf{x}, \mathbf{x}', P, P').$$

Řekneme, že τ je invariantní vůči transformaci H pokud

$$\tau(\mathbf{x}, \mathbf{x}', P, P') = H^{-1}\tau(\mathbf{x}, \mathbf{x}', PH^{-1}, P'H^{-1}).$$

4.1 Optimální metoda

Jestliže $\mathbf{x} \leftrightarrow \mathbf{x}'$ nesplňují (4.2), existují body $\bar{\mathbf{x}} \leftrightarrow \bar{\mathbf{x}}'$, které se nacházejí v jejich blízkosti a pro je splněna epipolární podmínka $\bar{\mathbf{x}}'^T F \bar{\mathbf{x}} = 0$.

Snažíme se najít takové body $\hat{\mathbf{x}}$ a $\hat{\mathbf{x}}'$, aby minimalizovaly následující funkci

$$C(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, \hat{\mathbf{x}})^2 + d(\mathbf{x}', \hat{\mathbf{x}}')^2 \tag{4.3}$$

tak, aby $\hat{\mathbf{x}}'^T F \hat{\mathbf{x}} = 0$. Euklidovskou vzdálenost mezi dvěma body představuje $d(*, *)$.

Minimum můžeme nalézt např. užitím Lavenberg-Marquardtovy metody, nebo pomocí Sampsonovy aproximace. Obě tyto metody využívají iteračních algoritmů. Hartley a Sturm [7] představili neiterační triangluační metodu, která nalezne požadované globální minimum funkce (4.3).

Body $\hat{\mathbf{x}}$ a $\hat{\mathbf{x}}'$, které přesně splňují (4.2), leží na epipolárních přímkách $\mathbf{l}$ a $\mathbf{l}'$. Platí, že každá dvojice bodů ležících na $\mathbf{l}$ a $\mathbf{l}'$ splňuje epipolární podmínku. Označme $\mathbf{x}_\perp$ bod ležící na $\mathbf{l}$, který je nejblíže původnímu $\mathbf{x}$. Obdobně definujme $\mathbf{x}'_\perp$ na $\mathbf{l}'$. Ze všech bodů, jež leží na $\mathbf{l}$ a $\mathbf{l}'$, právě $\mathbf{x}_\perp$ a $\mathbf{x}'_\perp$ minimalizují součet vzdáleností (4.3). Buď tedy $\hat{\mathbf{x}} = \mathbf{x}_\perp$ a $\hat{\mathbf{x}}' = \mathbf{x}'_\perp$.

Problém minimalizace teď můžeme přeformulovat tak, že nyní hledáme takové $\mathbf{l}$ a $\mathbf{l}'$, aby součet

$$d(\mathbf{x}, \mathbf{l})^2 + d(\mathbf{x}', \mathbf{l}')^2 \tag{4.4}$$

byl co nejmenší. Parametrizujeme epipolární přímky na prvním snímku parametrem t. Epipoláru na první fotografii tak můžeme zapsat jako $\mathbf{l}(t)$. Pomocí fundamentální matice F nalezneme odpovídající přímku $\mathbf{l}'(t)$ na druhé fotografii. Snažíme se najít t, které minimalizuje funkci

$$\min_t C = d(\mathbf{x}, \mathbf{l}(t))^2 + d(\mathbf{x}', \mathbf{l}'(t))^2. \tag{4.5}$$

Vhodnou volbou parametru t se z (4.5) stává racionální polynomiální funkce o proměnné t. Problém minimalizace vzdálenosti se tak změní v hledání reálných kořenů polynomu stupně šest.

Za předpokladu, že žádný z bodů $\mathbf{x}$ a $\mathbf{x}'$ není totožný s epipóly $\mathbf{e}$ a $\mathbf{e}'$, můžeme body přesunout do počátku soustavy souřadnic. Nové souřadnice tedy budou $\mathbf{x} = \mathbf{x}' = (0, 0, 1)^T$. Epipóly posuneme tak, aby platilo $\mathbf{e} =$

$(1,0,f)^T$ a $\mathbf{e}' = (1,0,f')^T$. Žádné z těchto posunutí nijak nemění hledání minima funkce (4.5).

Protože platí $F(1,0,f)^T = (1,0,f')F = 0$, bude mít fundamentální matice tvar

$$F = \begin{pmatrix} ff'd & -f'c & -f'd \\ -fb & a & b \\ -fd & c & d \end{pmatrix}.$$

Epipolární přímka $\mathbf{l}(t)$ prochází bodem $(0,t,1)^T$ a epipólem $\mathbf{e}$. Vektor reprezentující tuto přímku získáme vektorovým součinem $(0,t,1)\times(1,0,f) = (tf,1,-t)$. Z toho vzdálenost přímky od počátku, ve kterém se nyní nachází bod $\mathbf{x}$, získáme jako odmocninu z výrazu

$$d(\mathbf{x},\mathbf{l}(t))^2 = \frac{t^2}{1+t^2f^2}.$$

Druhou epipoláru vyjádříme s užitím fundamentální matice takto

$$\mathbf{l}'(t) = F(0,t,1)^T = (-f'(ct+d), at+b, ct+d)^T.$$

Vzdálenost $\mathbf{l}'(t)$ od počátku je dána

$$d(\mathbf{x}',\mathbf{l}'(t))^2 = \frac{(ct+d)^2}{(at+b)^2+f'^2(ct+d)^2}.$$

Hledáme tedy minimum následující funkce

$$s(t) = \frac{t^2}{1+t^2f^2} + \frac{(ct+d)^2}{(at+b)^2+f'^2(ct+d)^2}. \tag{4.6}$$

Jejím zderivováním získáme

$$s'(t) = \frac{2t}{(1+t^2f^2)^2} - \frac{2(ad-bc)(at+b)(ct+d)}{((at+b)^2+f'^2(ct+d)^2)^2}. \tag{4.7}$$

Funkce $s(t)$ nabývá maxima a minima pro $s'(t) = 0$. Upravíme (4.7) na nepodílový tvar a položíme rovno nule

$$g(t) = t((at+b)^2+f'^2(ct+d)^2)^2 - (ad-bc)(1+t^2f^2)^2(at+b)(ct+d) = 0.$$

Polynom $g(t)$ je stupně šest a může mít až šest reálných kořenů, které odpovídají třem minimům a třem maximům funkce $s(t)$. Postupným dosazením všech reálných částí kořenů do (4.6) najdeme absolutní minimum této funkce. Měli bychom také ověřit, zda funkce $s(t)$ nenabývá minima pro $t = \infty$. Poté, co nalezneme minimum, můžeme určit $\mathbf{l}$ a $\mathbf{l}'$ a následně i $\hat{\mathbf{x}}$ a $\hat{\mathbf{x}}'$. Nyní můžeme použít k získání $\hat{\mathbf{X}}$ některou z následujících lineárních metod.

4.2 Lineární metody

Triangulační lineární metody patří k nejjednodušším způsobům, jak spočítat $\mathbf{X}$ odpovídající dvojici bodů $\mathbf{x} \leftrightarrow \mathbf{x}'$. Pro každý bod lze ze vztahů (4.1) získat tři rovnice, z nichž dvě jsou lineárně nezávislé. Z rovnice $\mathbf{x} \times (P\mathbf{X}) = 0$ například získáme tyto tři

$$\begin{aligned} x(p^{3T}\mathbf{X}) - (p^{1T}\mathbf{X}) &= 0 \\ y(p^{3T}\mathbf{X}) - (p^{2T}\mathbf{X}) &= 0 \\ x(p^{2T}\mathbf{X}) - y(p^{1T}\mathbf{X}) &= 0 \end{aligned}$$

kde p^{iT} představuje i-tý řádek matice P. Z každé fotografie vybereme pro jednotlivé body vždy dvě z těchto tří rovnic. Pro jednu dvojici shodných bodů tak budeme mít vždy čtyři rovnice. Z nich sestavíme matici A o velikosti 4x4

$$A = \begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p'^{3T} - p'^{1T} \\ y'p'^{3T} - p'^{2T} \end{bmatrix}.$$

Rovnice (4.1) můžeme zapsat ve tvaru

$$A\mathbf{X} = 0, \tag{4.8}$$

kde neznámou je vektor homogenních 3D souřadnic bodu $\mathbf{X}$.

Soustavu můžeme řešit dvěma způsoby. Tím prvním, je homogenní metoda. Buď $A = UDV^T$ rozklad matice A na vlastní čísla. Pak řešením (4.8) je sloupec matice V, který odpovídá nejmenšímu vlastnímu číslu v D.

Druhá metoda se nazývá nehomogenní. Budeme požadovat, aby $\mathbf{X} = (X, Y, Z, 1)^T$. Tím se z (4.8) stává soustava čtyř rovnic o třech neznámých.

K vyřešení soustavy použijeme metodu nejmenších čtverců. Tato metoda není vhodná, pokud se poslední souřadnice $\mathbf{X}$ blíží nebo dokonce rovná nule (to se může stát u bodů, které leží v rovině v nekonečnu).

Je nutné poznamenat, že lineární algoritmy nejsou projektivně invariantní a ani nezajistí, aby platily geometrické podmínky (4.1). Proto je lepší nejdříve použít jednu z metod zmíněných v (4.1) a lineární algoritmy využít až k vypočtení $\hat{\mathbf{X}}$. Tak se zajistí invariance vůči projektivním i metrickým transformacím a také splnění všech geometrických podmínek.

Kapitola 5

Rekonstrukce povrchu

Známe-li 3D souřadnice bodů, můžeme vytvořit trojúhelníkovou síť, která bude reprezentovat povrch 3D modelu. Nejprve stručně uvedeme možné přístupy k rekonstrukci povrchu a zmíníme několik známých algoritmů. Následně se budeme podrobně zabývat popisem algoritmu, jenž využívá k získání povrchu pouze danou množinu bodů a její vnitřní vlastnosti.

5.1 Přehled algoritmů

Rekonstrukce povrchu z množiny roztroušených bodů představuje problém, který zatím nebyl úspěšně vyřešen. Existuje mnoho algoritmů, které se touto problematikou zabývají. Žádný z nich však nedokáže zaručit, aby za všech okolností výsledná rekonstrukce byla geometricky správná a odpovídala požadovanému povrchu.

Algoritmy můžeme rozdělit např. podle toho, jestli výsledný povrch aproximuje nebo interpoluje vstupní data. Interpolující algoritmy mají přesnější výstup, ale také vysoké naroky na kvalitu vstupních dat. Pro běžné fotografie zatížené šumem nejsou tedy moc vhodné. Aproximující algoritmy sice většinou nemají tak přísné požadavky na vstupní data jako ty interpolující, ale jejich výstupem je pouze odhad původního povrchu.

Další možností, jak se dají algoritmy pro rekonstrukci povrchu roztřídit, je podle způsobu vytváření povrchu. Dělí se pak například podle toho, zda jsou založeny na warpingu, vzdálenostní funkci, dělení prostoru nebo na postupném přidávání trojúhelníků do sítě (tzv. inkrementální algoritmy).

Algoritmy založené na warpingu nejdříve vytvoří síť trojúhelníků, která obalí množinu vstupních bodů, a pak ke všem vrcholům této sítě najde odpovídající body z dané množiny. Trojúhelníková síť se tedy deformuje tak, aby co nejvíce odpovídala hledanému povrchu. Z prací věnovaných warpingovým algoritmům můžeme zmínit například metodu navrženou Millerem [13].

Vzdálenostní funkce popisuje nejkratší vzdálenost bodu od předpokládaného povrchu. Pro uzavřené povrchy je její hodnota buď záporná, nebo kladná podle toho, zda bod leží uvnitř či vně rekonstruovaného objektu. Funkce se počítá pro každý bod vstupní množiny.

K dělení prostoru se nejčastěji používá Delaunayova tetrahedronizace. Z trojúhelníků, které tvoří čtyřstěny, se pak vybírají ty, jež tvoří výsledný povrch. Mezi nejznámější patří *CRUST* algoritmus, který trojúhelníky vybírá pomocí Voronoi diagramu. Tuto metodu poprvé popsala Amenta [2].

Výběr vhodných trojúhelníků může být proveden také pomocí tzv. α-*shapes*. Autorem této metody je Edelsbrunner [5] . Z tetrahedronizace se vyloučí všechy útvary, jejichž koule opsaná má větší poloměr než je parametr α. Nevýhodou algoritmu je, že α je vstupním parametrem a v průběhu výpočtu se nijak nepřizpůsobuje hustotě bodů. Vyžaduje tedy uniformní vzorkování vstupních dat. Byl navržen algoritmus [4], který tento nedostatek odstraňuje.

Všechny metody využívající Delaunayovu tetrahedronizaci jsou při velkém množství náročné na paměť i čas.

Poslední zmíněnou skupinou jsou inkrementální algoritmy. Určitým způsobem si určí první trojúhelník a další k němu pak postupně připojují na základě různých kritérií. Většina inkrementálních algoritmů pracuje i při větším množství dat rychle a bez velkých nároků na paměť.

Do této skupiny patří například tzv. *ball-pivoting algoritmus* (BPA) popsaný Bernardinim [3]. Obdobně jako α-*shapes*, i BPA vyžaduje jako vstupní parametr číslo udávající poloměr koule, která bude používána k získávání nových trojúhelníku. I BPA tedy vyžaduje uniformní vzorkování vtupních dat. Navíc je nutné u každého bodu znát jeho normálu.

Mezi inkrementální metody se také řadí algoritmus, který vybírá nové trojúhelníky pouze na základě vnitřních vlastností jednotlivých vstupních bodů [10]. Budeme se mu podrobně věnovat v následující podkapitole.

5.2 Algoritmus využívající vnitřních vlastností

Algoritmus využívá k postupnému vytváření trojúhelníkové sítě vnitřních vlastností dané množiny bodů. Kromě ní nevyžaduje tato metoda žádná další vstupní data. Je-li vzorkování dostatečně husté, algoritmus pracuje správně a vytvoří povrch, jenž poměrně přesně aproximuje tvar rekonstruovaného objektu. Z těchto dvou důvodů byl algoritmus vybrán pro rekonstrukci povrchu.

Důležité pojmy

Klíčovým pojmem je *sampling uniformity degree* (SUD) jednotlivých bodů. Jedná se o číslo, které odpovídá poměru délek nejdelší a nejkratší hrany přilehlé danému bodu. Protože jde o inkrementální algoritmus budeme SUD počítat z těch přilehlých hran, které v tu chvíli známe.

Oblast vlivu hrany e_{ij} je mnohostěn, který nám udává oblast, v níž budeme hledat body vhodné pro vytvoření nového trojúhelníku.

Hrany budeme dělit na *aktivní*, *vnitřní* a *hraniční*. Každá nově vzniklá hrana je aktivní. Hrana, které přiléhají dva trojúhelníky, je vnitřní. Pokud v oblasti vlivu aktivní hrany není žádný bod, který by mohl s danou hranou vytvořit vhodný trojúhelník, nazveme tuto hranu hraniční. Všechny hrany budou uchovávány ve frontě.

Pokud všechny hrany přilehlé bodu jsou vnitřní, pak daný bod nazveme *pevným*. Bod, který není pevný, je *aktivní*.

Nalezení prvního trojúhelníku

Z množiny bodů vybereme bod s nejvetší souřadnicí z a označíme jej **P**. Ze zbylých bodů vybereme ten, který je bodu **P** nejblíž a nazveme jej **Q**. Sestrojíme válec, jehož osou bude úsečka **l** spojující body **P** a **Q**. Střed **l** bude středem válce a délka **l** bude jeho výškou a průměrem. V případě, že tento válec nebude obsahovat žádné body ze vstupní množiny, budeme rovnoměrně zvětšovat jeho výšku a poloměr, dokud nějaké body obsahovat nebude. Z těchto bodů pak vybereme bod **R** tak, aby součet délek hran **RP** a **RQ** byl co nejmenší. Body **PQR** tvoří první trojúhleník sítě. Jeho hrany označíme za aktivní a vložíme je do fronty.

Spočítáme normálu **PQR**. Pokud je třetí souřadnice vektoru kladná, znamená to, že vektor směřuje ven. Jestliže bude záporná, pak směr vektoru obrátíme.

Oblast vlivu

Z fronty vezmeme první aktivní hranu e_{uv} a spočítáme její oblast vlivu. K tomu potřebujeme znát střed dané hrany $\mathbf{P_c}$, třetí vrchol $\mathbf{P_w}$ trojúhelníku, jehož je hrana součástí, těžiště tohoto trojúhelníku $\mathbf{P_t}$ a jeho normálu $\mathbf{N} = \mathbf{P_w P_u} \times \mathbf{P_w P_v}$. Dále určíme, které ze dvou SUD bodů $\mathbf{P_u}$ a $\mathbf{P_v}$ je větší a vynásobíme ho průměrnou délkou nejkratších hran přilehlých vrcholům $\mathbf{P_u}$ a $\mathbf{P_v}$. Tento součin označíme s. Nyní můžeme určit mnohostěn odpovídající oblasti vlivu hrany e_{uv}.

Každá stěna $\mathbf{S_i}$ je určena bodem $\mathbf{P_i}$, který obsahuje, a normálovým vektorem $\mathbf{N_i}$. Pro horní stěnu $\mathbf{S_1}$ platí $\mathbf{N_1} = \mathbf{N}$ a $\mathbf{P_1} = \mathbf{P_c} + s\mathbf{N}$. Dolní stěna $\mathbf{S_2}$ je dána $\mathbf{N_2} = -\mathbf{N}$ a $\mathbf{P_2} = \mathbf{P_c} - s\mathbf{N}$. Normálový vektor $\mathbf{N_3}$ je roven normalizovanému vektorovému součinu $\mathbf{N} \times \mathbf{P_t P_u}$ a $\mathbf{P_3} = \mathbf{P_u}$. Obdobně $\mathbf{N_4}$ získáme normalizací $\mathbf{N} \times \mathbf{P_t P_v}$ a $\mathbf{P_4} = \mathbf{P_v}$. Normálový vektor $\mathbf{N_5}$ poslední stěny je roven $\mathbf{P_u P_v} \times \mathbf{N}$ a $\mathbf{P_5} = \mathbf{P_c} + s\mathbf{N_5}$.

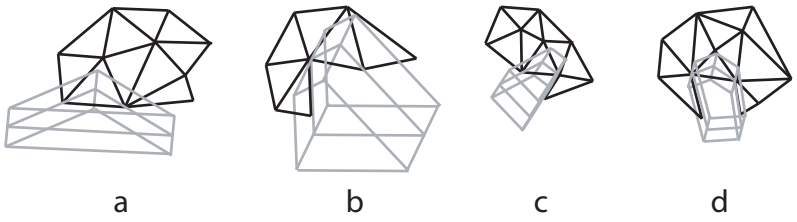
Pokud existují hrany přilehlé $\mathbf{P_u}$, které leží na stejné straně stěny $\mathbf{S_3}$ jako hrana e_{uv}, pak je třeba pro zachování geometrické integrity vytvořit další stěnu $\mathbf{S_6}$. Tuto situaci znázorňuje obrázek (5.1, b). Z množiny přilehlých hran vybereme tu, která svírá s hranou e_{uv} nejmenší úhel a označíme ji $\mathbf{P_u P_{left}}$. Normála stěny $\mathbf{S_6}$ je rovna vektorovému součinu $\mathbf{N} \times \mathbf{P_u P_{left}}$. Bod $\mathbf{P_6}$ by se měl rovnat $\mathbf{P_u}$ [10], ale lepších výsledků algoritmus dosahuje, pokud $\mathbf{P_6} = \mathbf{P_{left}}$. Obdobně, pokud existují hrany přilehlé $\mathbf{P_v}$, které leží na stejné straně stěny $\mathbf{S_4}$ jako hrana e_{uv}, vytváříme stěnu $\mathbf{S_7}$. Výsledný mnohostěn bude tedy mít minimálně pět, maximálně sedm stěn. Všechny možné případy zachycuje obrázek (5.1).

Určení vhodného bodu

Pokud se v mnohostěnu nenachází žádné aktivní body, je hrana e_{uv} označena jako hraniční. V opačném případě, je třeba z aktivních bodů v mnohostěnu vybrat ten nejvhodnější pro tvorbu nového trojúhelníku. K tomu nám poslouží kritérium vážené minimální délky. Buď T_{uvw} již existující trojúhelník přilehlý zkoumané hraně e_{uv}. Bod $\mathbf{P_n}$, který bude s e_{uv} tvořit nový trojúhelník T_{uvn} vybereme tak, aby následující součet byl co nejmenší

$$k_{uv} \left\|\mathbf{P_u} - \mathbf{P_v}\right\|^2 + k_{un} \left\|\mathbf{P_u} - \mathbf{P_n}\right\|^2 + k_{vn} \left\|\mathbf{P_v} - \mathbf{P_n}\right\|^2 . \qquad (5.1)$$

Označíme-li L délku hrany a A obsah trojúhelníku, pak koeficienty k_{uv},



Obrázek 5.1: Černou barvou je znázorněna již existující trojúhelníková síť. Mnohostěn určující oblast vlivu je šedé barvy.

k_{un} a k_{vn} spočítáme takto

$$k_{uv} = (L_{uw}^2 + L_{vw}^2 - L_{uv}^2)/A_{uvw} + (L_{un}^2 + L_{vn}^2 - L_{uv}^2)/A_{uvn},$$

$$k_{un} = k_{vn} = 2(L_{un}^2 + L_{vn}^2 - L_{uv}^2)/A_{uvn}.$$

Máme-li vybraný bod $\mathbf{P_n}$, musíme ještě otestovat, zda nový trojúhelník T_{uvn} nenaruší geometrickou integritu sítě. Otestujeme tedy průsečík tohoto trojúhelníku s trojúhelníky přilehlými ke všem bodům (aktivním i pevným) obsaženým v mnohostěnu. Pokud žádný průsečík neexistuje, nebo je totožný s nějakou již existující aktivní či hraniční hranou v síti, pak je vše v pořádku a můžeme T_{uvn} přidat do sítě. V opačném případě není $\mathbf{P_n}$ vyhovující a musíme ze zbylých aktivních bodů opět vybrat ten, pro který bude součet (5.1) co nejmenší, a znovu ověřit geometrickou integritu. Pokud se nám ze všech aktivních bodů v mnohostěnu nepodaří najít ani jeden, který by integritu sítě nenarušil, označíme hranu e_{uv} za hraniční.

Vybrali jsme bod $\mathbf{P_n}$ splňující všechny požadavky a můžeme tedy vytvořit nový trojúhelník T_{uvn}. Hranu e_{uv} označíme jako vnitřní. Hrany e_{un} nebo e_{vn} značíme jako aktivní nebo jako vnitřní podle toho, zda mají jeden, nebo dva přilehlé trojúhelníky. Aktualizujeme status bodů $\mathbf{P_u}$, $\mathbf{P_v}$ a $\mathbf{P_n}$.

Kapitola 6

Implementace

Tato kapitola se věnuje implementaci programu. Nejdříve zdůvodníme výběr použitých algoritmů a následně naznačíme strukturu programu. Zmíníme požadavky na vstupní data a popíšeme výstupní soubory programu. Na závěr uvedeme poznatky spojené s testováním aplikace na různých datech.

6.1 Výběr algoritmů

U metod popsaných v předchozích kapitolách byly uváděny také jejich výhody a nevýhody. Především na jejich základě byly vybrány algoritmy pro implementaci.

K nalezení fundamentální matice byl použit normalizovaný 8-mi bodový algoritmus. Jak uvádí Hartleym [6], může tento algoritmus dávat i přes svou jednoduchost uspokojivé výsledky.

Z metod popsaných ve třetí kapitole byla vybrána ta, která k výpočtu metrické struktury využívá kalibračních dat a z nich získané esenciální matice. Na rozdíl od jiných uvedených postupů, stačí uživateli pro dané nastavení fotoaparátu získat kalibrační data jen jednou a pak je může používat k výpočtu metrického modelu u všech fotografií pořízených tímto fotoaparátem.

Jak bylo napsáno v (3.1), esenciální matici můžeme získat jak úpravou fundamentální matice pomocí kalibračních dat, tak i normalizací vstupních bodů kalibračními daty. Implementovány byly obě metody a ukázalo se, že lepší výsledky dává metoda první.

Pro výpočet 3D souřadnic byla kvůli odstranění geometrické chyby nejprve použita optimální metoda z (4.1). Z lineárních triangulačních metod

popsaných v (4.2) byla vyzkoušena jak homogenní, tak i nehomogenní metoda. Pro lepší výsledky byla vybrána metoda nehomogenní.

K rekonstrukci povrchu byl vybrán algoritmus využívající vnitřních vlastností (5.2). Tato metoda byla zvolena především pro malé nároky na *vzhled* vstupní množiny bodů a také proto, že kromě dané množiny nevyžadovala jiná data. Oproti algoritmům založeným na dělění prostoru je výrazně rychlejší a méně náročná na paměť.

6.2 Struktura programu

Proces rekonstrukce 3D scény se dá rozdělit na čtyři hlavní části - na zpracování vstupních dat, spočítání 3D souřadnic, nalezení povrchu a vytvoření výstupního modelu. Podle toho byla také rozdělena struktura programu. Každé z těchto částí je věnován jeden `*.cpp` soubor. Navíc obsahuje program ještě dva soubory `*.cpp` - jeden pro realizaci uživatelského rozhraní a druhý pro práci s maticema. Hlavičky jednotlivých funkcí a procedur, popř. makra se nacházejí v odpovídajících `*.h` souborech. Ke spuštění programu je nutný XML soubor `SPhotos3D.glade`, který obsahuje popis vzhledu uživatelského rozhraní.

Popis globálních proměnných, funkcí a procedur se nachází přímo ve zdrojových kódech ve formě komentářů.

`main.cpp`

Stará se především o realizaci uživatelského prostředí. Využívá k tomu knihoven GTK+ 2.0. Samotný návrh vzhledu je obsažen v XML souboru s názvem `SPhotos3D.glade`. Dále ukládá názvy vstupních a výstupních souborů do globálních proměnných a také nastavuje formát vstupních souborů a typ souborů výstupních. Kontroluje, zda uživatel zadal všechny nutné vstupní soubory.

`input.cpp`

Načítá ze vstupního souboru shodné body a informace o rozměrech fotografií. Kontroluje přitom také správnost formátu vstupního souboru. Při načítání shodných bodů eliminuje dvojice bodů, které jsou uvedeny víckrát (některé programy na hledání shodných bodů totiž úvádějí občas některou dvojici dvakrát).

`points.cpp`

Zajišťuje výpočet 3D souřadnic. Obsahuje funkce a procedury nutné k získání jak projektivních, tak metrických souřadnic.

Pro výpočet optimální 3D struktury je nutné najít reálné kořeny polynomu stupně šest. Používá se k tomu Laguerrova metoda a její implementace se nachází v `roots.h`. Tentýž soubor obsahuje také strukturu komplexních čísel a procedury, které jsou pro práci s nimi nutné.

Během výpočtu se často vyžívá rozkladu na vlastní čísla. Všechny potřebné funkce k nalezení rozkladu se nacházejí v souboru `svdcmp.h`.

`surface.cpp`

Zajišťuje nalezení povrchu z dané množiny bodů. V `surface.h` se nachází struktury, které byly pro potřeby algoritmu vytvořeny. Jde zejména o struktury definující bod (*point*), hranu (*edge*), trojúhelník (*triangle*) a stěnu (*face*).

V rámci rekonstrukce povrchu je nutné testovat protínání trojúhelníků. Rychlý a efektivní způsob testování byl popsán a implementován Tomasem Möllerem [14]. Nachází se v `triangle.h`.

`outputs.cpp`

Realizuje namapování textur a vznik všech výstupních souborů.

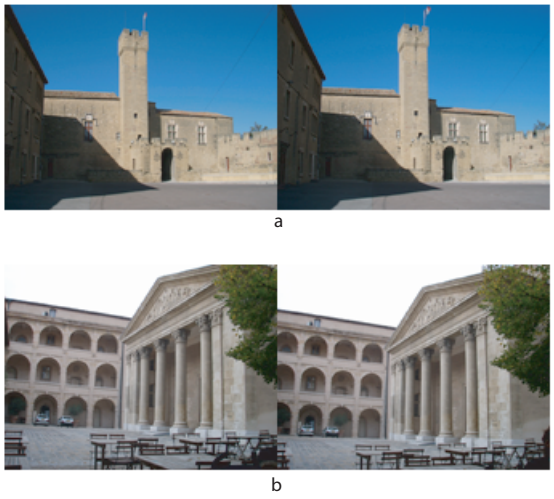
`matrix.cpp`

Soubor obsahuje základní algoritmy pro práci s maticema a s vektory. Matice jsou realizovány dvourozměrným polem, vektory pak jednorozměrným. V maticích jsou také uloženy načtené vstupní body, výsledné 3D souřadnice a vyhledávací tabulky v `surface.cpp`.

Kromě alokace a uvolnění paměti jsou v `matrix.cpp` a `matrix.h` také makra a funkce pro transpozici, násobení, kopírování a tisk matic.

6.3 Vstupní a výstupní data

Vstupní data tvoří dvě fotografie zachycující scénu, kterou chceme rekonstruovat, a soubor se souřadnicema shodných bodů. Předpokládáme, že fotografie byly pořízeny klasickým fotoaparátem s CCD snímačem, popř. webkamerou. Soubor se souřadnicema musí obsahovat minimálně osm dvojic



Obrázek 6.1: (a) Hrad v Arles. (b) Klášter v Marseille.

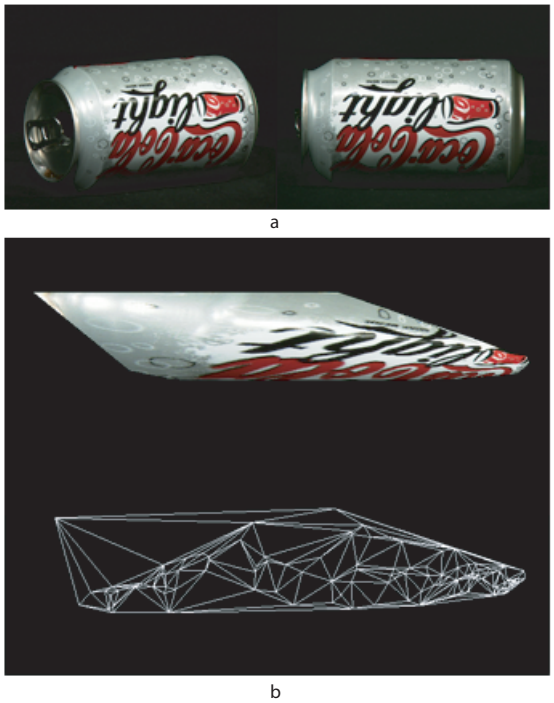
shodných bodů. Tato data jsou povinná a postačují k vytvoření projektivního modelu.

Pokud uživatel bude chtít metrickou strukturu, musí navíc zadat soubor s kalibračními daty.

Podrobným popisem formátů vstupních souborů se zabývá (A.3).

Výsledný 3D model bude uložen ve formátu VRML nebo X3D. VRML i X3D jsou jazyky určené pro popis prostorových scén. Soubory typu VRML, mající většinou příponu wrl, jsou textové, takže je možné je upravovat běžnými textovými editory. Soubory typu X3D mohou být uloženy ve dvou různých formátech - jeden je založen na XML a druhý je velmi podobný jazyku VRML. Program vytváří soubory druhého typu.

Hlavním výstupem programu je (VRML nebo X3D) soubor, který obsahuje 3D model scény. Dále program vytvoří soubor s *bodovým* 3D modelem. Jedná se zobrazení modelu pouze pomocí 3D bodů. Tento soubor je vytvořen ve chvíli, kdy je dokončen výpočet 3D souřadnic modelu, a je opět buď ve formátu VRML nebo X3D. Celý průběh rekonstrukce je zaznamenáván v textovém souboru.



Obrázek 6.2: (a) Vstupní fotografie. (b) Výsledný model a jeho *drátěná* verze.

6.4 Výsledné modely

Prostorovost 3D modelů se špatně zachycuje na obrázcích, proto jich zde moc neuvedeme. Několik ukázek 3D modelů vytvořených programem se však nachází na CD ve složce `Results`.

Program byl testován jak na fotografiích, které zachycují jediný objekt, tak i na běžných fotografiích zobrazujících libovolnou scénu. Prokázalo se, že pro samotný výpočet rekonstrukce není tento rozdíl nijak podstatný.

Naopak kvalita modelu samozřejmě závisí na rovnoměrnosti pokrytí zobrazované scény shodnými body a na přesnosti vstupních dat. Ukázalo se také, že velkou roli hraje úhel mezi jednotlivými fotografiemi. Obrázek (6.1) uvádí dvě dvojice fotografií. U první dvojice se podařilo získat bodový 3D model (`castle_points.wrl`), který správně vystihl tvar zobrazované scény. U druhé dvojice je úhel, který fotografie svírají, jiný. V bodovém modelu

této (`temple_points.wrl`) scény leží skoro všechny body v rovině. V obou případech se přitom jedná o metrický model.

U projetkivních modelů se ukázalo, že nejlepších výsledků se dosáhne, pokud úhel mezi fotografiemi je 30°. Existuje databáze fotografií [23], kde můžeme nalézt i tzv. stereo páry fotografií, které svírají přesně úhel 30°. Na obrázku (6.2) můžeme vidět použité fotografie z databáze a ukázku výsledného modelu. Fotografie pořízené pod jiným úhlem nedávají většinou tak pěkné výsledky.

6.5 Použité softwarové nástroje a knihovny

Program byl vyvíjen pod operačním systémem Windows XP a testován byl na Linuxu.

Pro vývoj aplikace byl použit jak program Microsoft Visual Studio 2005, tak i program Code::Blocks verze 8.02 [21]. Uživatelské rozhraní je realizováno pomocí knihoven GTK+ 2.0 [19]. K navržení vzhledu uživatelského rozhraní pak byla použita aplikace Glade 3.0 [20].

Kapitola 7

Závěr

V této práci jsme prezentovali dostupné metody k získání 3D rekonstrukce scény. Zabývali jsme se také implementací programu a představili jsme výsledky, kterých dosáhl. Na úplný závěr zhodnotíme tyto výsledky a navrhneme možná vylepšení programu.

7.1 Zhodnocení

Program byl testován na různých datech. Byla prokázána závislost jednak na přesnosti vstupních dat, jednak na úhlu, který mezi sebou fotografie svírají. Naopak scéna zobrazená na fotografiích na kvalitu výsledného modelu vliv nemá.

Jak bylo uvedeno v kapitole věnující se rekonstrukci povrchu, zatím neexistuje algoritmus, který by zaručil nalezení topologicky správného povrchu z libovolné množiny bodů. Ani námi zvolený algoritmus využívající vnitřních vlastností bodů nenajde vždy úplně *korektní* povrch. Navíc je na velkých datech velmi pomalý. Dalo by se tedy říci, že rekonstrukce povrchu představuje největší slabinu programu.

Vzhledem k tomu, že se zabýváme rekonstrukcí ze stereo fotografií, dají se pokládat dosažené výsledky po stránce výpočtu 3D souřadnic za uspokojivé. Naopak tvorba povrchu z těchto souřadnic by se dala ještě hodně vylepšit.

7.2 Možná vylepšení

Pro zajištění nejlepších možných výsledků by se měl implementovat algoritmus *Gold Standard* zmíněný v (2.2). Fundamentální matice a 3D souřadnice, které v programu získáme pomocí normalizovaného 8-mi bodového algoritmu a vybraných triangulačních metod, slouží v metodě *Gold Standard* pouze jako výchozí odhad. Ten se postupně pomocí Lavenberg-Marquardtovy metody zdokonaluje tak, aby minimalizoval geometrickou chybu. Implementace *Gold Standard* algoritmu tedy spočívá v dodání zmíněné iterační metody k již existujícím algoritmům.

Dále by se dala značně vylepšit rekonstrukce povrchu. Především by se měla zvýšit rychlost algoritmu. Na velkých datech je algoritmus velmi pomalý, což by vzhledem k výsledkům uváděným jeho autory [10] neměl. Množina vstupních bodů by měla být rozdělena pomocí uniformní mřížky, aby se urychlil výběr bodů, které patří do oblasti vlivu.

Navíc se u některých modelů objevil výskyt trojúhelníků, které by součástí modelu být neměly, a tak narušují jeho celkový vzhled. Autoři této metody doporučují upravit získaný povrch některým z postfiltračních algoritmů [1]. Implementace vhodné postfiltrace by mohla tento nežádoucí jev odstranit.

Dále by mohla být implementována některá z dalších metod umožňujících metrickou rekonstrukci. Uživatel by si pak mohl zvolit, zda chce zadat kalibrační data, nebo údaje o scéně.

Pro uživatele by samozřejmě bylo pohodlnější, kdyby hledání shodných bodů na fotografiích bylo přímo součásti programu. Zrovna tak by součástí programu mohla být funkce pro vypočítání kalibračních dat fotoaparátu.

Příloha A

Uživatelská dokumentace

A.1 Systémové požadavky

Program je multiplatformní a měl by bez problémů fungovat jak na operačním systému Windows, tak na unixových systémech.

U unixových systémů je nutné, aby knihovny GTK+ byly verze 2.0 a vyšší. Jinak nemá program žádné požadavky na přítomnost dalších knihoven či programů.

A.2 Popis obsahu CD

- Složka `Data` obsahuje několik ukázek vstupních souborů, fotografií a také modelů, které z nich byly vytvořeny.
- Složka `Results` obsahuje několik ukázek 3D modelů vytvořených programem.
- Složka `Source` obsahuje zdrojové kódy programu a soubor XML s uživatelským rozhraním `SPhotos3D.glade`.
- Složka `SPhotos3D-unix` obsahuje soubory nutné ke kompilaci programu na unixových systémech.
- Složka `SPhotos3D-win` obsahuje spustitelný soubor `SPhotos3D.exe` a knihovny nutné pro běh programu.
- V souboru `Manual.pdf` naleznete uživatelskou dokumentaci.

- Soubor `Documentation.pdf` obsahuje programátorskou dokumentaci k programu.

A.3 Pořizování fotografií

Fotografie můžete pořídít běžným digitálním fotoaparátem s CCD snímačem, popř. webkamerou. Obě fotografie musí samozřejmě obsahovat stejnou scénu, ovšem každá ji musí zachycovat z jiné pozice. To znamená, že fotoaparát musíte po pořízení prvního snímku buď natočit o určitý ne moc velký úhel (ideálně 30°), nebo posunout, popř. obojí. Pokud to neuděláte a pořídíte obě fotografie ze stejné pozice a pod stejným úhlem a použijete např. pouze zoom, pak rekonstrukce nebude možná. Posunutí i otočení však nesmí být moc velké, jinak by nebylo možné najít dostatečný počet shodných bodů. Několik ukázek vhodných dvojic fotografií můžete nalézt ve složce `Data`. Vždy je lepší při fotografování používat stativ a ukládat snímky s co nejlepším možným rozlišením.

A.4 Získání shodných bodů

Když máte fotografie, je potřeba nalézt shodné body - tedy souřadnice těch bodů, které jsou obsaženy na obou fotografiích. Na přesnosti souřadnic závisí kvalita výsledného modelu a je tedy vhodné použít k jejich nalezení některý s dostupných programů. Jednou z nejlepším a volně dostupných aplikací je autopano-sift [18]. Tento program se zabývá tvorbou panoramat z fotografií, což samozřejmě také vyžaduje znalost shodných bodů. Můžete ho tedy použít k nalezení shodných souřadnic na fotografiích. O formátu, v jakém získáte výstup z této aplikace, se více dozvíte v části věnované vstupním souborům.

Aby bylo možné vypočítat 3D model, je nutné znát alespoň osm dvojic shodných bodů. Pokud program, který k získání souřadnic použijete, najde méně než osm shodných bodů, nebude možné 3D rekonstrukci získat. K tomu nejčastěji dochází proto, že rozdíl mezi scénami na fotografiích je příliš velký. Další možnou příčinou jsou nekvalitní fotografie zatížené šumem.

Maximum shodných bodů není nijak dáno. Platí sice, že čím více shodných bodů máte k dispozici, tím lepší můžete získat 3D model, ale na druhé straně musíte mít na paměti, že rekonstrukce z velkého počtu bodů může

trvat hodně dlouho. Pokud se tedy s ohledem na rychlost používaného počítače rozhodnete omezit počet shodných bodů, je lepší udělat to už při jejich hledání. Program autopano-sift např. umožňuje zadat maximální množství bodů, které chcete najít. Využitím této možnosti zaručíte jednak adekvátní počet bodů, a jednak to, že získané body budou stále zachycovat danou scénu rovnoměrně. Pokud byste totiž měli soubor s velkým množstvím bodů a rozhodli se část z nich odstranit ručně, mohlo by se stát, že by zbylé body zachycovaly jen nějakou část z celé scény.

A.5 Vstupní data

Vstupní data programu tvoří dvě fotografie a soubor se shodnými body. Zatímco na typ fotografií nejsou kladeny žádné zvláštní požadavky, soubor s daty má pevný formát. Jedná se o textový soubor, kde na prvním řádku je číslo udávající počet shodných bodů. Na druhém řádku je uvedena šířka první fotografie a na třetím řádku její výška. Na čtvrtém a pátém řádku je šířka a výška druhé fotografie. Na následujících řádcích se pak nacházejí souřadnice shodných bodů - na jednom řádku jsou vždy čtyři čísla, kde první udává x-ovou a druhé y-ovou souřadnici bodu na první forografii a třetí a čtvrté číslo udává souřadnice odpovídajícího bodu na fotografii druhé. Namísto desetinné čárky musí být v souřadnicích použita tečka. V opačném případě totiž program načte pouze celou část čísla a tím dojde k velkým nepřesnostem během výpočtu. Program načte maximálně takové množství shodných bodů, jaké je uvedeno na prvním řádku.

V případě, že jste k nalezení shodných souřadnic použili autopano-sift, získali jste soubor s příponou pto. Tento soubor má svůj specifický formát. Můžete z něj získat všechny informace pro tvorbu výše popsaného vstupního souboru, ale také můžete jako vstup použít přímo soubor `*.pto`. Přesněji řečeno program umí získat potřebná data z `*.pto` souboru, pokud byl tento soubor vytvořen aplikací autopano-sift verze 2.4. V případě jiných verzí je nutné si nejdříve ověřit, že soubor `*.pto` má stejnou strukturu jako u verze 2.4.

Příklady obou popsaných formátů naleznete ve složce `Data`.

Z fotografií a shodných bodů už lze získat projektivní 3D model scény. Projektivní prostor nezachovává tvar, délku, vzdálenost ani úhly. Proto projektivní model může působit dost nepřirozeně. Lepší je získat metrickou re-

konstrukci scény. K tomu je však nutná znalost kalibračních dat použitého fotoaparátu. Jedná se o data popisující vnitřní vlastnosti fotoaparátu. Bohužel nebývají tato data úváděna v parametrech fotoaparátů. Získat je můžete např. pomocí programu GML Camera Calibration Toolbox [17]. Na přesnosti kalibračních dat závísí kvalita výsledného metrického modelu, proto se vyplatí věnovat jejich získání dostatečnou pozornost.

Pokud jste pro pořízení fotografií použili pouze jeden fotoaparát se stejným nastavením, pak stačí uvést kalibrační data vztahující se k tomuto nastavení. V případě, že jste použili jeden fotoaparát, ale pokaždé s jiným nastavením (jedná se především o funkci zoom), pak je nezbytné uvést kalibrační data pro obě nastavení. Totéž platí v případě, že jste použili dva různé fotoaparáty.

Programy na získání kalibračních dat produkují soubory obsahující různé informace. Pro sestavení souboru s kalibračními daty potřebujete znát pouze následující hodnoty

- Ohniskovou vzdálenost pro směr x a směr y. V ideálním případě by měla být tato dvě čísla totožná. Kvůli různým nepřesnostem se však většinou nepatrně liší. Tyto hodnoty mívají označení *focal length* nebo pouze *fc*. Pozor - nejedná se o tentýž údaj, jaký je uváděn v parametrech fotoaparátu!

- Koeficient zkosení. U většiny fotoaparátů a webkamer by měl být nulový. U méně kvalitní přístrojů však může mít nenulovou hodnotu. Označuje se např. jako *skew coefficient*, *s*, nebo *alpha_c*.

- Souřadnice tzv. *hlavního bodu*. Mohou být označeny jako *principal point*, *pp* nebo *cc*.

Kalibrační data budou uložena v textovém souboru. Na prvním řádku bude číslo 1 nebo 2 podle toho, zda soubor obsahuje jedny, nebo dvoje kalibrační data. Na druhém řádku bude ohnisková vzdálenost ve směru osy x a na třetím ve směru y. Na čtvrtém řádku bude faktor zkosení. Na patém se bude nacházet x-ová souřadnice *hlavního bodu*, na šestém pak y-ová. Bude-li to potřeba, připojte za tyto údaje ve stejném pořadí ještě kalibrační data vztahující se k druhému fotoaparátu (či pouze druhému nastavení). Je důležité zachovávat pořadí dat - kalibrační data, která odpovídají první fotografii, musí být umístěna jako první! Záměna dat by se negativně projevila na výsledném modelu.

I v tomto souboru je nutné u čísel používat tečku namísto desetinné čárky. Vzorové soubory opět najdete ve složce `Data`.

A.6 Výstupní data

Výstupem programu je 3D model uložený ve formátu VRML nebo X3D. Výsledný model můžete upravovat buď pomocí libovolného textového editoru, nebo pomocí modelovacích programů - většina z nich totiž podporuje oba tyto formáty. K prohlížení 3D modelu stačí nainstalovat plug-in do internetového prohlížeče - např. Cortonu [22], která zobrazuje VRML modely (určena je však zejména pro Internet Explorer a na jiných prohlížečích nemusí zobrazovat modely správně).

Kromě modelu 3D scény vytváří program i další výstupní soubory. Řekněme, že program vytvořil požadovaný model a uložil jej do souboru s názvem `model.wrl`. Navíc vytvořil soubory `model_points.wrl` a `log.txt`.

V souboru `model_points.wrl` je uložena *bodová* struktura rekonstruované scény. Jde o model, který je složen pouze z bodů odpovídajících vstupním dvojicím shodných bodů. V podstatě představuje tento soubor jakousi rezervu pro případ, že by program nedoběhl až do konce. Časově nejnáročnější částí rekonstrukce je totiž spojování jednotlivých bodů do trojúhelníku tak, aby vytvořili povrch rekonstruované scény. Pokud by během tohoto výpočtu došlo z nějakého důvodu k přerušení programu, získáte alespoň bodový model, který se vytvoří ještě před zahájením výpočtu povrchu.

Pokud se stane, že po otevření souboru s modelem nic v prohlížeči neuvidíte, je to způsobeno tím, že prohlížeč nezabírá tu část prostoru, ve které se model nachází. Přemísťování v prostoru prohlížeče realizují mimo jiné pomocí tzv. *Viewpoints*. V souboru s modelem je nadefinován *Viewpoint*, ze kterého je 3D model vidět. Stačí si tedy tento *Viewpoint* v prohlížeči zvolit. Některé prohlížeče mají také korekční tlačítko (u Cortony se toto tlačítko jmenuje *Fit*), po jehož zmáčknutí se zobrazí část prostoru s modelem.

Může se také stát, že výsledný model nebude otexturován. Příčiny mohou být dvě. Cesta k zadaným fotografiím je neplatná (zadali jste ji špatně, nebo jste fotografie smazali, či přemístili jinam). Druhou příčinou mohou být speciální požadavky zvoleného prohlížeče na rozlišení textury. Některé prohlížeče mají dané maximální rozlišení textur a nepovolí zobrazení textury s vyšším rozlišením.

Posledním souborem je `log.txt`, v němž je zachycen celý průběh výpočtu.

A.7 Uživatelské rozhraní

Obrázek A.1: Uživatelské rozhraní programu

V části ***Vstup*** si zvolíte vstupní parametry programu. Zadáte soubor se vstupními daty a cestu k oběma fotografiím, k nímž se data vztahují. Je důležité, aby bylo zachováno pořadí fotografií - fotografie, jejíž data jsou v souboru na prvním místě, musí být zadána jako první. Dále si zvolíte formát vstupního souboru. Volba *autopano* odpovídá formátu vytvořeném pomocí programu autopano-sift verze 2.4. Zvolíte-li *vlastní*, musí mít vstupní soubor formát popsaný výše. Pokud budete chtít metrický model, musíte navíc

zadat soubor s kalibračními daty. Pokud nezadáte všechny požadované soubory (tzn. tři pro projektivní, čtyři pro metrický model), nemůžete zahájit rekonstrukci. Dobře si zkontrolujte, zda jste vše zadali správně, protože při špatném formátu souboru program ohlásí chybu a skončí. Museli byste vše zadávat znovu.

V části ***Výstup*** si zvolíte jméno, umístění a typ souboru, do kterého bude uložen výsledný model. Pokud nezadáte žádný název, bude soubor pojmenován `3DModel`. U typu souboru si můžete vybrat mezi *wrl* a *x3d*, ale můžete také označit oba typy - v takovém případě, bude model uložen jak ve VRML, tak i v X3D. Nebude-li při zahájení výpočtu označena žádná z možností, bude výsledný soubor typu VRML.

Název žádného ze souborů (a ani cesta k němu) NESMÍ obsahovat diakritiku!

Tlačítkem *Start* spustíte vlastní rekonstrukci modelu. V okně, které se objeví, můžete sledovat průběh výpočtu. Po úspěšném vytvoření modelu ukončíte tlačítkem *Zavřít* celý program. Pokud by během rekonstrukce došlo k nějaké chybě, objeví se chybové hlášení a program bude ukončen.

Literatura

[1] U. Adamy, J. Giesen, M. John: *Surface Reconstruction Using Umbrella Filters*, Compute Geometry, vol. 21, pp. 63-86, 2002.

[2] N. Amenta, M. Bern, M. Kamvysselis: *A new Voronoi-based surface reconstruction algorithm*, SIGGRAPH, pp. pp. 415 - 421, 1998.

[3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin: *The Ball-Pivoting Algorithm for Surface Reconstruction*, IEEE Transactions on Visualization and Computer Graphics, vol. 5(4), pp. 349-359, 1999.

[4] H. Edelsbrunner: *Weighted alpha shapes*, Journal of Graphics Tools, vol. 2(2), pp. 25-30, 1997.

[5] H. Edelsbrunner, E. P. Mücke: *Three-dimensional alpha shapes*, ACM Trans. Graphics, vol. 13, pp. 43-72, 1994.

[6] R. I. Hartley: *In Defense of the Eight-Point Algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pp. 580-593, June 1997.

[7] R. I. Hartley, P. Sturm: *Triangulation*, Computer Vision and Image Understanding, vol. 68(2), pp. 146–157, November 1997.

[8] R. I. Hartley, A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, Reading, 2003.

[9] T. S. Huang, O. D. Faugeras: *Some Properties of The E-matrix in Two-view Motion Estimation*, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 11, pp. 1310–1312, 1989.

[10] H.-W. Lin, C.-L. Tai, G.-J. Wang: *A mesh reconstruction algorithm driven by intrinsic property of point cloud*, Computer-Aided Design, vol. 36, pp. 1-9, January 2004.

[11] H. C. Longuet-Higgins: *A Computer Algorithm for Reconstructing A Scene from Two Projections*, Nature, vol. 293, pp. 133–135, Septemebr 1981.

[12] Q. T. Luong, T. Viéville: *Canonical Representations for The Geometries of Multiple Projective Views*, Computer Vision and Image Understanding, vol. 64(2), pp. 193–229, September 1996.

[13] J. V. Miller, D. E. Breen, W. E. Lorenzem, R. M. O'Bara, M. J. Wozny: *Geometrically deformed models: A Method for extracting closed geometric models from volume data*, Proc. SIGGRAPH, pp. 217 - 226, 1991.

[14] T. Möller: *A fast triangle-triangle intersection test*, Journal of Graphics Tools, vol. 2(2), pp. 25-30, 1997.

[15] R. Y. Tsai, T. S. Huang: *Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects With Curved Surfaces*, IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 6, pp. 13–27, 1984.

[16] Z. Zhang: *Determining the Epipolar Geometry And Its Uncertainty - a Review*, International Journal of Computer Vision, vol. 27(2), pp. 161–195, March 1998.

[17] `http://graphics.cs.msu.ru/en/research/calibration`

[18] `http://cs.tu-berlin.de/~nowozin/autopano-sift`

[19] `http://www.gtk.org`

[20] `http://glade.gnome.org`

[21] `http://www.codeblocks.org`

[22] `http://www.parallelgraphics.com/products/cortona`

[23] `http://staff.science.uva.nl/~aloi`