

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

Rekonstrukce objektu a pozice
pozorovatele z 2D pohledů

Abstrakt

Anotace v češtině : Rekonstrukce objektu a pozice pozorovatele z 2D pohledů.

Tato práce popisuje postup rekonstrukce reálného světa z 2D pohledů. Tento postup se skládá s mnoha kroků, počínaje detekcí významných bodů a konče vlastní rekonstrukcí. Zatímco ostatní práce se často zaměřují pouze na část tohoto postupu, my popisujeme tento postup celý.

Tuto práci můžeme logicky rozdělit na tři hlavní části. První je teoretické pozadí, zejména pak projektivní geometrie. Druhá část obsahuje metody řešící jednotlivé kroky rekonstrukčního procesu. Ve třetí části byla většina popsaných metod implementována a testována v několika experimentech.

Ve většině kapitol je zároveň naznačen další postup kterým se daná problematika ubírá a jehož detailní popis přesahuje rozsah této práce.

Klíčová slova: 3D rekonstrukce, detekce bodů, projektivní geometrie, kalibrace kamery, autokalibrace, epipolární geometrie, trifokální tensor, projektivní rekonstrukce, metrická rekonstrukce

Abstract

English annotation: Reconstruction of object and observer position from 2D views.

This diploma thesis describes the process of reconstruction of a real world scene given several images of it. This process consists of many steps from feature detection to self reconstruction. While other papers often describe only a part of this process, we describe it completely.

We can logically divide this work into three main parts. The first is theoretical background, especially projective geometry. The second part contains methods that solve individual steps of reconstruction process. In the third part there are most of the described methods implemented and tested in several experiments.

Majority of chapters indicate further progresses by which given problems are de-tracted and whose detailed description exceeds the range of this thesis.

Keywords: 3D reconstruction, feature detection, projective geometry, camera calibration, autocalibration, epipolar geometry, trifocal tensors, projective reconstruction, metric reconstruction.

Obsah

Abstrakt.....	1
Abstract.....	1
Obsah	2
Prohlášení o samostatnosti.....	4
Kapitola 1 Úvod.....	5
Kapitola 2 Projektivní geometrie.....	7
2.1 Homogenní souřadnice	8
2.2 Geometrické transformace.....	11
2.2.1 Projektivní transformace.....	11
2.2.2 Afinní transformace	13
2.2.3 Metrické transformace	14
2.2.4 Eukleidovské transformace.....	14
2.3 Kuželosečky a kvadriky.....	15
2.3.1 Kuželosečky a duální kuželosečky	15
2.3.2 Kvadriky a duální kvadriky	16
2.3.3 Absolutní kuželosečky a absolutní duální kvadriky	17
2.4 Model perspektivní kamery	18
2.4.1 Základní model perspektivní kamery	18
2.4.2 Vnitřní a vnější parametry kamery	19
2.4.3 Kalibrace kamery, IAC a DIAC	21
2.4.4 Určení parametrů kamery z její matice.....	22
2.4.5 Reálná kamera.....	23
Kapitola 3 Geometrie více pohledů	24
3.1 Epipolární geometrie.....	24
3.1.1 Geometrická reprezentace.....	24
3.1.2 Algebraická reprezentace.....	25
3.1.3 Výpočet fundamentální matice ze známých kamer	26
3.1.4 Výpočet fundamentální matice z korespondencí	26
3.1.5 Měření chyb	28
3.2 Trifokální tensor	29
3.2.1 Geometrická reprezentace.....	29
3.2.2 Algebraická reprezentace.....	30
3.2.3 Výpočet epipolární geometrie a matic kamer z trifokálního tensoru.....	31
3.2.4 Výpočet trifokálního tensoru z matic kamer.....	32
3.2.5 Výpočet trifokálního tensoru z korespondencí	32
3.3 Kvadrifokální a multifokální tensor	33
Kapitola 4 Nalezení korespondencí mezi snímky.....	34
4.1 Detekce významných bodů.....	35
4.1.1 Harrisův operátor	36
4.1.2 FindFP.....	38
4.2 Inicializační korespondence.....	40
4.2.1 Korelace	40
4.2.2 MEM-Pairs.....	41
4.2.3 Určení Afinní transformace	42
4.3 RANSAC	43
4.3.1 R-RANSAC	45

4.3.2 PLUNDER	48
4.4 Alternativní metody k algoritmu RANSAC	50
Kapitola 5 Projektivní rekonstrukce	51
5.1 Výpočet projektivních hloubek	52
5.1.1 Výpočet projektivních hloubek založený na epipolární geometrii	52
5.1.2 Iterační algoritmus výpočtu projektivních hloubek	53
5.2 Metoda faktorizace	53
5.3 Doplnění chybějících bodů	55
Kapitola 6 Metrická rekonstrukce	59
6.1 Metrická rekonstrukce a absolutní kvadrika	59
6.2 Kolik snímků je potřeba?	61
6.3 Normalizační metoda	62
6.3.1 Příklad 1	65
6.3.2 Příklad 2	66
6.3.3 Příklad 3	67
6.3.4 Shrnutí normalizační metody	67
Kapitola 7 Shrnutí algoritmu 3D rekonstrukce	68
7.1 Detekce bodů	68
7.2 Hledání korespondencí	68
7.3 Projektivní rekonstrukce	69
7.4 Metrická rekonstrukce	69
7.5 Vylepšení modelu na základě známých kamer a polohy některých bodů	69
7.6 Rekonstrukce povrchu a textur	70
7.7 Různé druhy vstupu	70
7.8 Nestatická scéna	71
Kapitola 8 Experimentální ověření popsaných metod	72
8.1 Bodové detektory	72
8.2 MEM-Pairs a RANSAC	74
8.3 Faktorizace	77
8.3.1 Syntetická data	77
8.3.2 Reálná data	79
8.4 Normalizace	80
8.4.1 Syntetická data	80
8.4.2 Reálná data	82
Kapitola 9 Implementační část	83
Kapitola 10 Závěr	86
Literatura	87
Příloha A Singular value decomposition (SVD)	90
A.1 Řešení soustavy homogenních rovnic	91
A.2 Pseudinverze a řešení soustavy nehomogenních rovnic	91
A.3 Snížení hodnosti matice	91
A.4 Rozklad $Q=AT$	92
Příloha B Programátorský manuál	93
B.1 GUI	93
B.2 Panely Nástrojů	94
B.3 Algoritmická část	95
Příloha C Uživatelský manuál k programu	96
Příloha D Evidenční list	102

Prohlášení o samostatnosti

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 19.5.2004,
Ladislav Lang

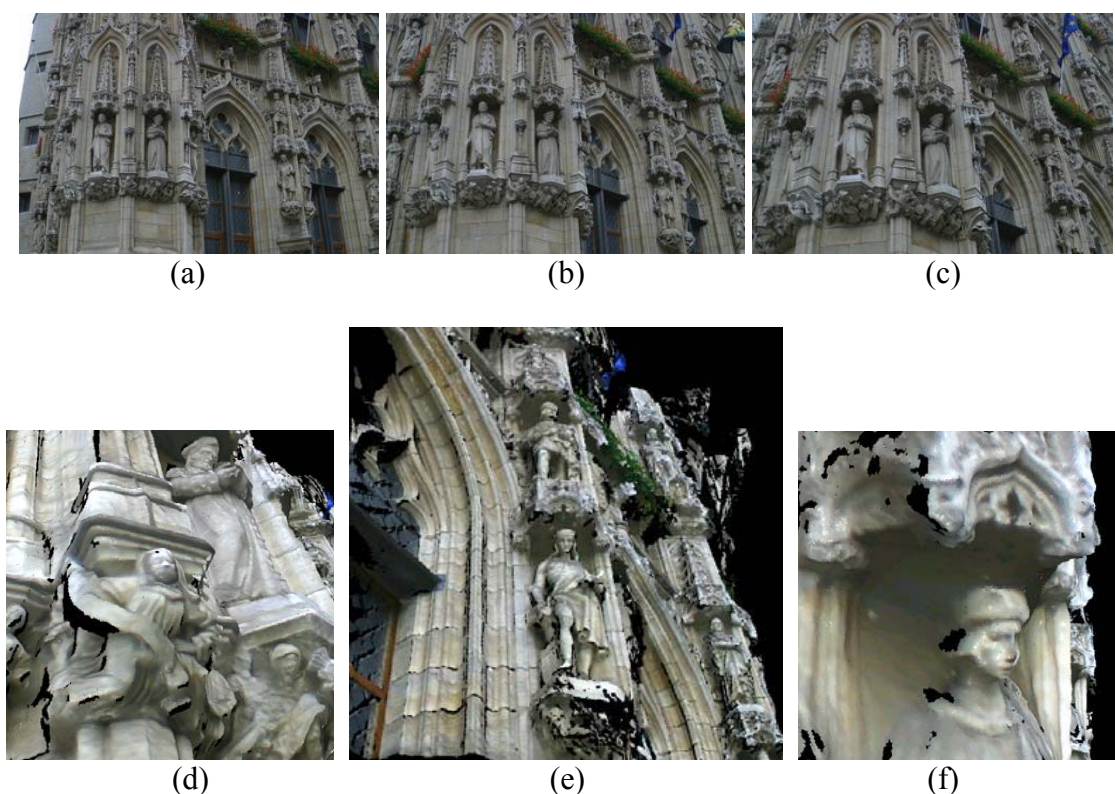
Tento projekt byl podporován z Microsoft Research Ltd. – projekt 2003-178
a MŠMT ČR projekt MSM 235200005.

Kapitola 1

Úvod

Trojrozměrná rekonstrukce z nekalibrovaných dvojrozměrných snímků (dále jen rekonstrukce nebo 3D rekonstrukce) je jednou z hlavních oblastí počítačového vidění, v posledních 10-ti letech prošla tato problematika bouřlivým rozvojem. Cílem 3D rekonstrukce je vytvořit z několika snímků vyfocených z různých pozic, nebo z videosekvence, 3D model pozorované scény.

3D rekonstrukce je alternativou k drahým a obtížně realizovatelným 3D skenerům, zejména při skenování rozměrných objektů (v architektuře, apod.). Existuje celá řada oblastí, ve kterých je možno rekonstrukci uplatnit např.: 3D portréty (obecně 3D fotografie), 3D filmy, rozšířená realita (augmented reality), trojrozměrná kartografie, měření (zejména měření rozměrů) a mnoho dalších.



Obrázky 1.1¹: Příklad 3D rekonstrukce. Obrázky (a,b,c) zastupují množinu originálních snímků, (d,e,f) jsou výsledky rekonstrukce (konkrétně jde o 3-pohledový „stereo“ algoritmus, viz [Strecha-04]). Černé body reprezentují místa, která nejsou na žádném snímku viditelná.

V této práci se budeme zabývat 3D rekonstrukcí s jedním hlavním omezením a to, že pozorovaná scéna bude statická, tj.: nebude se pohybovat, ani nijak měnit. Další omezení jsou dána použitými algoritmy a nejsou nijak významná. Jak ukážeme v kapitole (6) je k přesné rekonstrukci za určitých podmínek potřeba alespoň tři různých pohledů. Je-li však dodána další informace je možno správně rekonstruovat i ze dvou

¹ Zdroj obrázků: [Strecha-04], URL=<http://www.esat.kuleuven.ac.be/~cstrecha/testimages/>

snímků (stereo reconstruction) a dokonce pouze z jednoho snímku (singleview reconstruction). I přes to, že v některých kapitolách popisujeme vztahy mezi dvojicí snímků, budeme na vstupu vyžadovat minimálně tři, optimálně však několik desítek snímků.

Naším cílem je získat ze snímků maximální množství 3D informací o pozorované scéně. Takto zrekonstruovaný model je možno vylepšit využitím dodaných informací o scéně, například informace o symetrii objektů. Je tedy možno rekonstruovat i ty objekty nebo části objektů, které nejsou na žádném ze snímků viditelné. Touto částí rekonstrukce se již zabývat nebudeme, vyžaduje totiž znalosti které jsou vázány na konkrétní problematiku, například při rekonstrukci obličeje můžeme předpokládat jeho symetrii, u rekonstrukce budov můžeme využít znalosti o pravoúhlosti, paralelnosti stěn a podobně.

V kapitole (2, 3) se budeme zabývat teoretickým pozadím, které je pro rekonstrukci nezbytné. Kapitola (2) se zabývá projektivním prostorem, geometrickými transformacemi a v neposlední řadě modelem perspektivní kamery a její kalibrací, což je jedna z podstatných částí rekonstrukčního procesu. kapitola (3) popisuje geometrické vztahy mezi jednotlivými snímky a zároveň nám dává matematický aparát k popisu těchto vztahů.

Proces rekonstrukce můžeme rozdělit do dvou hlavních kroků: Získání korespondencí na množině snímků a následná 3D rekonstrukce založená na těchto korespondencích. Získáním korespondencí se zabývá kapitola (4), která začíná detekcí bodů a končí robustním odstraněním chybných korespondencí založeným na geometrii popsané v kapitole (2). Vlastní 3D rekonstrukci rozdělujeme na projektivní rekonstrukci z korespondencí, kapitola (5) a na přechod od projektivní rekonstrukci k metrické (která je cílem této práce) v kapitole (6).

Shrnutím celého postupu, implementací a otestováním algoritmů se zabývají kapitoly (7-9).

Cílem této práce je snaha shrnout celý rozsáhlý postup 3D rekonstrukce, aniž by byly opomenuty některé důležité fáze. Nejde tedy o to dotáhnout jednotlivé algoritmy k dokonalosti, ale spíše o poskytnutí návodu a souborného pohledu na celou problematiku.

Kapitola 2 Projektivní geometrie

Projektivní geometrie je základním matematickým aparátem potřebným pro 3D rekonstrukci, je tomu tak proto, že lidské oko (a většina běžných fotoaparátů a kamer) vidí okolní svět „změněný“ projektivní transformací. V této kapitole se zaměříme na základní znalosti projektivní geometrie, popíšeme model kamery a teoretické pozadí potřebné ke kalibraci kamery. Užitím těchto znalostí můžeme teoreticky ukázat jak rekonstruovat 3D scénu pouze z 2D snímků.



Obrázek 2.1²: Lidské vidění světa - ukázka perspektivní projekce. Paralelní přímky nejsou v průmětu paralelní a protínají se v „nekonečnu“.

Abychom mohli mluvit o projektivních transformacích musíme nejprve definovat projektivní prostor na kterém tyto transformace provádíme:

Definice 2.1: Přidáme-li do 3D eukleidovského prostoru E^3 *nevlastní rovinu*, označujeme takový prostor jako *rozšířený eukleidovský prostor* a značíme ho $\overline{E^3}$. Podobně přidáním *nevlastní přímky* do E^2 získáme *rozšířenou eukleidovskou rovinu* $\overline{E^2}$.

Definice 2.2: Považujeme-li v prostoru $\overline{E^n}$ *nevlastní a vlastní body* (přímky, roviny,...) za rovnocenné, potom se prostor $\overline{E^n}$ nazývá *projektivní prostor* a značíme ho P^n .

² Zdroj obrázku: URL=<http://mywebpages.comcast.net/pmccullough/cruise/perspective.jpg>

Pojmy nevlastní bod, přímka a nevlastní rovina budou vysvětleny v kapitole (2.1).

Hlavním charakteristickým znakem projektivní geometrie je, že promítneme-li do roviny kružnici, získáme kuželosečku. Kuželosečky v E^2 a kvadriky v E^3 mají velký význam v projektivní geometrii a jak později ukážeme, také v procesu 3D rekonstrukce. Kuželosečkám a kvadrikám je proto věnována kapitola (2.3).

Obsah této kapitoly byl z větší části čerpán z [Hartley-04, Edberg-00], další informace je možno získat v libovolné publikaci zabývající se projektivní geometrií. Detailnější informace o kuželosečkách a kvadrikách je možno nalézt v [Janyška-01].

2.1 Homogenní souřadnice

Jedním z důvodů zavádění homogenních souřadnic, zejména v počítačové grafice, je usnadnění maticových operací. Z našeho hlediska je však významnější, že zavedení homogenních souřadnic postačuje k rozšíření eukleidovského prostoru na prostor projektivní, jak si ukážeme v této kapitole.

Homogenní souřadnice bodu ve 2D

Homogenní souřadnice bodu X jsou v rovině definovány jako uspořádaná trojice (x, y, w) , pro kterou platí, že

$$(x_k, y_k) = \left(\frac{x}{w}, \frac{y}{w} \right) \quad (2.1.1)$$

jsou kartézské souřadnice bodu X . Homogenní souřadnice jsou tedy rozšířením kartézských souřadnic o jednu dimenzi.

Je-li $w \neq 0$, můžeme získat z homogenních souřadnic souřadnice kartézské. V opačném případě, tedy $w = 0$, není možno kartézské souřadnice definovat a hovoříme o tzv. *nevlastních bodech*, tedy bodech ležících v nekonečnu. Nevlastní body bývají často používány k popisu směrových vektorů.

Homogenní souřadnice $X = (x, y, w)$ vynásobené nenulovou skalární hodnotou $\alpha \neq 0$, tedy $\alpha X = (\alpha x, \alpha y, \alpha w)$, popisují stále stejný bod

$$\alpha X = \left(\frac{\alpha x}{\alpha w}, \frac{\alpha y}{\alpha w}, 1 \right) = \left(\frac{x}{w}, \frac{y}{w}, 1 \right) = X \quad (2.1.2)$$

Důsledkem rovnice (2.1.2) je, že zatímco bodu v homogenních souřadnicích odpovídá právě jeden bod v souřadnicích kartézských (s výjimkou nevlastních bodů), tak bodu v kartézských souřadnicích odpovídá celá množina bodů $\alpha X \quad \forall \alpha \neq 0$.

Homogenní souřadnice, jejichž poslední složka w je rovna jedné nazýváme *normalizované homogenní souřadnice*.

Homogenní souřadnice přímky ve 2D

Bod $X = (x, y, w)$ v homogenních souřadnicích leží na přímce L právě tehdy, pokud platí že $ax + by + cw = 0$. Reprezentaci přímky $L = (a, b, c)$ nazýváme *homogenní souřadnice přímky* L .

Užitím maticové notace můžeme obecnou rovnici roviny psát jako:

$$XL^T = LX^T = 0 \quad (2.1.3)$$

Po vynásobení homogenních souřadnic přímky (a, b, c) nenulovou skalární hodnotou $\alpha \neq 0$ reprezentují nově získané souřadnice $\alpha(a, b, c)$ stejnou přímku, tedy obdobně jako u bodu platí $\alpha L = L$.

Jak již naznačuje předchozí odstavec a rovnice (2.1.3), existuje mezi homogenními souřadnicemi bodu a přímky nějaký vztah, tento vztah nazýváme *princip duality* a umožňuje nám bez jakékoli újmy zaměnit bod a přímku: Dva různé body mají společnou právě jednu přímku (2.1.4) ~ Dvě různé přímky se protínají právě v jednom bodě (2.1.5). Tento vztah můžeme maticově popsat jako:

$$X \times X' = L, \quad (2.1.4)$$

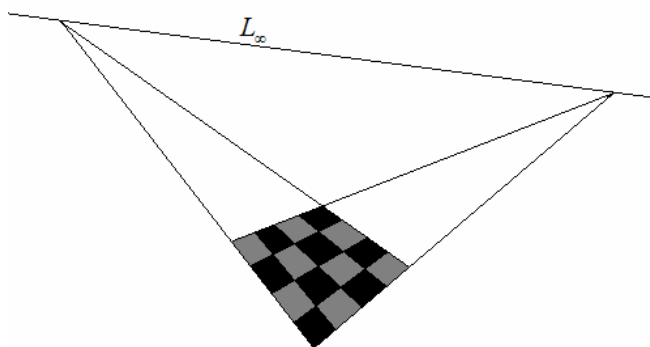
$$L \times L' = X, \quad (2.1.5)$$

kde \times je vektorový součin.

Všechny nevlastní body definují přímku o souřadnicích $L_\infty = (0, 0, 1)$, tato přímka je umístěná v nekonečnu a je nazývána *nevlastní přímka*. Její existence je zajištěna použitím homogenních souřadnic.

Zavedli jsme tedy nevlastní přímku, čímž se nám podle definice (2.1) a (2.2) podařilo rozšířit prostor na projektivní.

Dosadíme-li do rovnice (2.1.5) rovnoběžné přímky získáme nevlastní bod (bod umístěný v nekonečnu). Při použití homogenních souřadnic mají tedy i rovnoběžky společný bod. Narozdíl od eukleidovského prostoru platí v projektivním prostoru princip duality bez jakýchkoli výjimek (v eukleidovském prostoru nemají rovnoběžky společný bod). Princip duality je jedním z hlavních důvodů zavedení homogenních souřadnic.



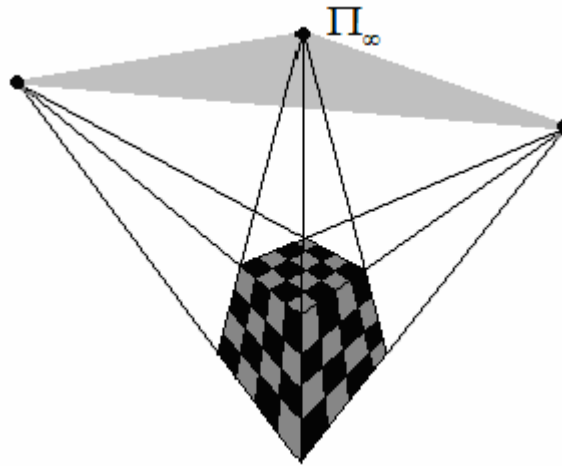
Obrázek 2.1.1: Nevlastní přímka.

Homogenní souřadnice bodu a roviny ve 3D

Podobně jako v rovině definujeme homogenní souřadnice v prostoru jako uspořádanou čtveřici $X = (x, y, z, w)$, pro kterou platí $(x_k, y_k, z_k) = (x/w, y/w, z/w)$. Ekvivalentem k přímce ve 2D je v prostoru rovina, která je definována obecnou rovnicí $ax + by + cz + dw = 0$, to odpovídá homogenním souřadnicím roviny $\Pi = (a, b, c, d)$.

Duální dvojice v prostoru jsou: bod-rovina, přímka-přímka. Věta: „Průsečíkem dvou různých rovin je právě jedna přímka“ sice platí, ale neexistuje pro ni věta duální. Naopak věty: „Tři různé body definují právě jednu rovinu ~ Průsečíkem tří různých rovin je právě jeden bod“ duální jsou.

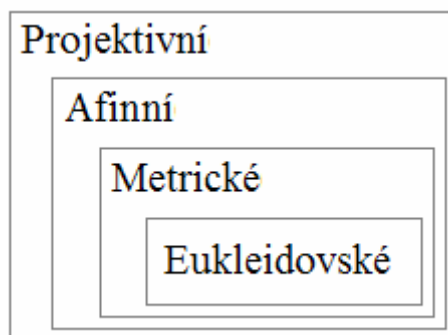
Průsečíkem dvou rovnoběžných rovin je nevlastní přímka, nevlastní přímky všech rovin tvoří nevlastní rovinu o souřadnicích $\Pi_\infty = (0, 0, 0, 1)$. Podobně jako ve 2D, zavedením nevlastní roviny rozšíříme prostor na projektivní.



Obrázek 2.1.2: Nevlastní rovina.

2.2 Geometrické transformace

V této kapitole se budeme zabývat zejména transformacemi v 3D prostoru, transformace v ostatních dimenzích je možno vyjádřit analogicky. Geometrické transformace rozdělíme podle toho, kolik mají invariantů. Nejobecnější skupinou jsou transformace projektivní, veškeré další množiny transformací jsou podmnožinou transformací projektivních, nejméně obecné jsou z hlediska tohoto výkladu transformace eukleidovské.



Obrázek 2.2.3: Hierarchie transformací.

2.2.1 Projektivní transformace

Projektivní třídimenzionální prostor značíme P^3 . Jak již bylo zmíněno v úvodu kapitoly jsou projektivní transformace nejobecnější, obsahují tedy největší množství elementárních transformací (posun, otočení, ...) a zároveň mají nejnížší počet invariantů.

jak jsme si již ukázali v kapitole (2.1) je jedinou podmínkou vytvoření projektivního prostoru zavedení homogenních souřadnic, projektivní transformace pak mohou transformovat vlastní body na nevlastní a naopak.

Projektivní transformaci v P^3 reprezentujeme maticí H_P o rozměrech 4×4 .

$$H_P = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & 1 \end{bmatrix}. \quad (2.2.1)$$

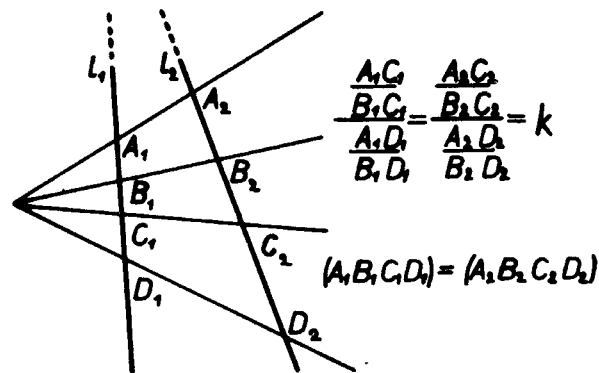
Totéž můžeme zapsat mnohem přehledněji blokovým zápisem:

$$H_P = \begin{bmatrix} A & t \\ v & 1 \end{bmatrix}, \quad (2.2.2)$$

kde A je regulární matice o rozměrech 3×3 , t označuje posun $t = (t_x, t_y, t_z)^T$ a v je obecný tříprvkový vektor.

Protože se význam matice H_p nemění se změnou měřítka, můžeme ji vynásobit libovolnou skalární hodnotou (můžeme psát $h_{44} = 1$). Projektivní transformace má 15 stupňů volnosti.

Nejvýznamnějším invariantem projektivní transformace je tzv. *dvojpoměr čtyř bodů*:

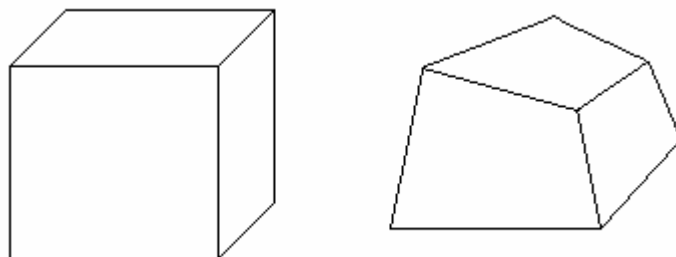


Obrázek 2.2.2: Dvojpoměr čtyř bodů.

Mějme body A, B, C, D ležící na jedné přímce, pak podíl dělicích poměrů

$$k = \frac{\overrightarrow{AC}}{\overrightarrow{BC}} : \frac{\overrightarrow{AD}}{\overrightarrow{BD}} \quad (2.2.3)$$

nazýváme dvojpoměr k bodu D vzhledem k bodu C pro základní body A, B . A zapisujeme $k = (ABCD)$. Záměnou bodu a přímky získáme, díky principu duality, ekvivalentní větu pro přímky.



Obrázek 2.2.3: Projektivně ztransformovaná krychle.

2.2.2 Afinity transformace

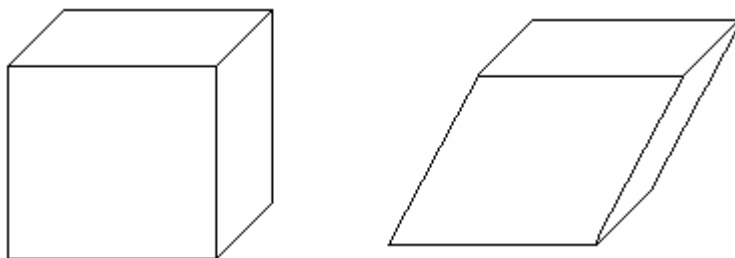
Afinní třídímní prostor značíme A^3 . Afinní transformace jsou podmnožinou transformací projektivních. A^3 má větší množství invariantů a menší množinu transformací. Od P^3 se afinní prostor liší zejména v tom, že v A^3 je pevně daná rovina nevlastní roviny. Rovnice roviny má 3 stupně volnosti, z toho plyne, že Afinní transformace bude mít stupňů volnosti 12.

Afinní transformaci v A^3 reprezentujeme maticí H_A o rozměrech 4×4 . V blokovém zápisu můžeme psát:

$$H_A = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix}, \quad (2.2.4)$$

kde A je regulární matice o rozměrech 3×3 , t označuje posun $t = (t_x, t_y, t_z)^T$ a 0 je nulový tříprvkový vektor.

Invarianty afinních transformací jsou v první řadě všechny invarianty transformací projektivních. Jak již bylo řečeno, je v afinním prostoru invariantem nevlastní rovina, z čehož se dají odvodit další invarianty afinní transformace, např.: zachování rovnoběžnosti.



Obrázek 2.2.4: Afinně ztransformovaná krychle

Nevlastní rovina $\Pi_\infty = [\pi_1^\infty \ \pi_2^\infty \ \pi_3^\infty \ \pi_4^\infty]$ je rovina, kde se protínají rovnoběžné roviny a přímky. Nalezneme-li rovnici nevlastní roviny v projektivním prostoru, můžeme tuto rovinu transformovat do její pozice v prostoru afinním $[0 \ 0 \ 0 \ 1]$, transformaci vyjádříme následovně

$$H_{P \rightarrow A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \pi_1^\infty / \pi_4^\infty & \pi_2^\infty / \pi_4^\infty & \pi_3^\infty / \pi_4^\infty & 1 \end{bmatrix}. \quad (2.2.5)$$

A tímto způsobem převést rekonstrukci z projektivní na afinní.

2.2.3 Metrické transformace

Metrický třídímní prostor značíme M^3 . Metrické transformace jsou podmnožinou transformací Afinních. Metrická transformace definuje objekt nazývaný *absolutní kuželosečka* (z angl.: *absolute conic*), což je kuželosečka umístěná na nevlastní rovině. Je definována symetrickou maticí 3×3 a nemění se změnou měřítka. Absolutní kuželosečka má tedy 5 stupňů volnosti a protože afinní transformace má stupňů volnosti 12, zbývá metrické rekonstrukci 7 stupňů volnosti. Podrobnosti o kuželosečkách se je možno dozvědět v kapitole (2.3).

Metrickou transformaci v M^3 reprezentujeme maticí H_M o rozměrech 4×4 . V blokovém zápisu můžeme psát:

$$H_M = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}, \quad (2.2.6)$$

kde R je rotační ortogonální matice (platí tedy $RR^T = R^T R = I$) o rozměrech 3×3 , t označuje posun $t = (t_x, t_y, t_z)^T$, 0 je nulový tříprvkový vektor a s vyjadřuje globální změnu měřítka.

Protože metrická transformace umožňuje globální změnu měřítka, není vzdálenost mezi dvěma body invariantem, ale poměr vzdáleností již ano. Podobně jako u afinních transformací, získávají i metrické invarianty obecnějších transformací, tedy projektivní a afinní. Invariant kterým se odlišuje metrická transformace od afinní je absolutní kuželosečka. Nalezením absolutní kuželosečky při afinní rekonstrukci můžeme tuto převést na rekonstrukci metrickou.

2.2.4 Eukleidovské transformace

Eukleidovský třídímní prostor značíme E^3 . Jediný rozdíl mezi eukleidovským a metrickým prostorem je, že eukleidovský prostor neumožňuje globální změnu měřítka, jediné povolené transformace jsou posun a otočení.

Metrickou transformaci v M^3 reprezentujeme maticí H_M o rozměrech 4×4 . V blokovém zápisu můžeme psát:

$$H_M = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}, \quad (2.2.7)$$

kde význam jednotlivých prvků je stejný jako u metrické rekonstrukce, rovnice (2.2.6).

Narozdíl od metrického prostoru, je invariantem vzdálenost mezi dvěma body. Abychom mohli transformovat metrickou rekonstrukci na eukleidovskou postačuje zadání nějaké vzdálenosti ve scéně. Ostatní invarianty má eukleidovská transformace společné s metrickou.

2.3 Kuželosečky a kvadriky

Kuželosečky a kvadriky hrají důležitou roli v auto-kalibraci. Jak jsme se již zmínili v kapitole (2.2.3) je tzv. absolutní kuželosečka invariantem metrické rekonstrukce a její znalost nám umožňuje převést problém projektivní rekonstrukce na rekonstrukci metrickou.

Definice 2.3.1: Množinu všech bodů $x \in P^n$, pro které platí, že

$$xQx^T = \sum_{i,j=0}^n q_{ij}x_ix_j = 0 \quad (2.3.1)$$

nazýváme *kvadrika*. symetrická matice Q je kvadratická forma o rozměrech $n \times n$, která tuto kvadriku reprezentuje. V případě, že $n = 2$ se kvadrika nazývá *kuželosečka*. Pod pojmem kvadrika budeme dále rozumět kvadriku v P^3 .

Vynásobíme-li kvadriku Q libovolnou nenulovou skalární hodnotou $\alpha, \alpha \neq 0$, rovnice (2.3.1) stále platí, αQ tedy určuje tutéž kvadriku a můžeme psát $Q = \alpha Q$.

2.3.1 Kuželosečky a duální kuželosečky

Kuželosečky

Kuželosečka Ω v P^2 (elipsa, hyperbola, parabola, ...) je definována jako symetrická matice 3×3 , pro kterou platí:

$$x\Omega x^T = 0 \quad (2.3.2)$$

a která se nemění se změnou měřítka. Vlastní kuželosečka je tvořena množinou bodů v homogenních souřadnicích $x \in P^2$. Kuželosečka má 5 stupňů volnosti, pět bodů v obecné pozici definuje kuželosečku.

Duální kuželosečky

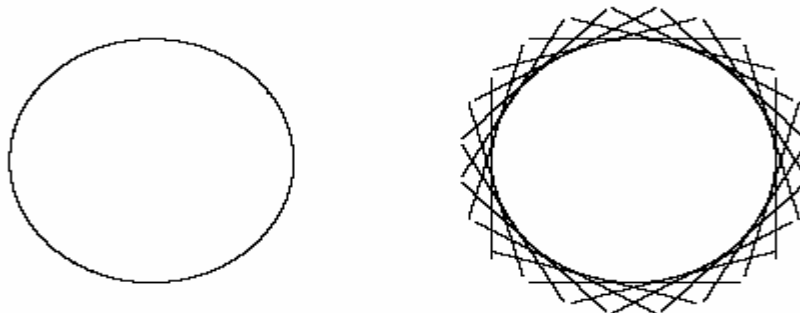
Použijeme-li principu duality a nahradíme body x za přímky l , získáme duální reprezentaci kuželosečky, tzv.: *duální kuželosečku* Ω^* definovanou vztahem:

$$l\Omega^*l^T = 0. \quad (2.3.3)$$

Ω^* reprezentuje opět kuželosečku a platí, že Ω^* je symetrická matice o rozměrech 3×3 nezávislá na změně měřítka. Je-li Ω regulární, můžeme psát následující vztah mezi ní a kuželosečkou duální:

$$\Omega^* \approx \Omega^{-1}. \quad (2.3.4)$$

Ω^* je v tomto případě inverzní matice k matici Ω .



Obrázek 2.3.1: Příklad kuželosečky a kuželosečky k ní duální

Podrobíme-li body transformaci $x' = Hx$, bodová kuželosečka bude transformována jako:

$$\Omega' = H^{-T} \Omega H^{-1}, \quad (2.3.5)$$

pro odpovídající duální kuželosečku platí:

$$\Omega^{*'} = H \Omega^* H^T, \quad (2.3.6)$$

2.3.2 Kvadriky a duální kvadriky

Kvadriky

Obdobě jako kuželosečku Ω v P^2 je kvadrika Q v P^3 (dále jen kvadrika) definována jako symetrická matice 4×4 , nezávislá na změně měřítka, pro kterou platí:

$$xQx^T = 0 \quad (2.3.7)$$

kde $x \in P^3$ je množina bodů v homogenních souřadnicích tvořící kvadriku (elipsoid, hyperboloid, paraboloid, ...). Kvadrika má 9 stupňů volnosti, devět bodů v obecné pozici definuje kvadriku. Průsečíkem kvadriky s libovolnou rovinou je kuželosečka.

Duální kvadriky

Použijeme-li principu duality a nahradíme body x tentokrát za roviny π , získáme duální reprezentaci kvadriky, tzv.: *duální kvadriku* Q^* definovanou vztahem:

$$\pi Q^* \pi^T = 0. \quad (2.3.8)$$

Q^* reprezentuje opět kvadriku a platí, že Q^* je symetrická matice o rozměrech 4×4 nezávislá na změně měřítka. Je-li Q regulární, můžeme psát následující vztah mezi ní a kuželosečkou duální:

$$Q^* \approx Q^{-1}. \quad (2.3.9)$$

Q^* je v tomto případě inverzní matice k matici Q .

Podrobíme-li body transformaci $x' = xH$, bodová kvadrika bude transformována jako:

$$Q' = H^{-T} Q H^{-1}, \quad (2.3.10)$$

pro odpovídající duální kvadriku platí:

$$Q^{*'} = H Q^* H^T, \quad (2.3.11)$$

2.3.3 Absolutní kuželosečky a absolutní duální kvadriky

Absolutní kuželosečky

Pod pojmem absolutní kuželosečka Ω_∞ , rozumíme kuželosečku umístěnou na nevlastní rovině Π_∞ . V metrickém prostoru platí $\Pi_\infty = (0 \ 0 \ 0 \ 1)$ a body $X = (x, y, z, w)$ tvořící Ω_∞ můžeme vyjádřit jako:

$$\left. \begin{array}{l} x^2 + y^2 + z^2 \\ w \end{array} \right\} = 0, \quad (2.3.12)$$

Rovnici (2.3.12) můžeme pro $w = 0$ zapsat ve tvaru:

$$(x \ y \ z) I (x \ y \ z)^T, \quad (2.3.13)$$

pak Ω_∞ koresponduje s kuželosečkou C , s maticí $C = I$. Kuželosečka C je tvořena imaginárními body.

Dokážeme-li nalézt v projektivním prostoru absolutní kuželosečku, můžeme tento prostor transformovat na metrický tak, že se nalezená kuželosečka transformuje do $C = I$. Zejména tohoto faktu pak využíváme při přechodu z projektivní rekonstrukce kapitola (5) k rekonstrukci metrické kapitola (6), jak jsme to již naznačili v kapitole (2.2.3).

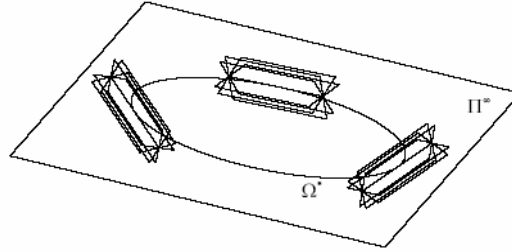
Absolutní duální kvadrika

Protože je absolutní kuželosečka definována v prostoru (P^3) je jejím duálním obrazem degenerovaná kvadrika Q_∞^* , tzv.: *absolutní duální kvadrika*, která je tvořena všemi tečnými rovinami k Ω_∞ .

Algebraicky je Q_∞^* reprezentována maticí o rozměrech 4×4 s hodnotí 3, kterou můžeme v metrickém 3D prostoru zapsat takto:

$$Q_\infty^* = \begin{bmatrix} I & 0^T \\ 0 & 0 \end{bmatrix}. \quad (2.3.14)$$

Absolutní duální kvadrika se nemění s metrickou transformací.



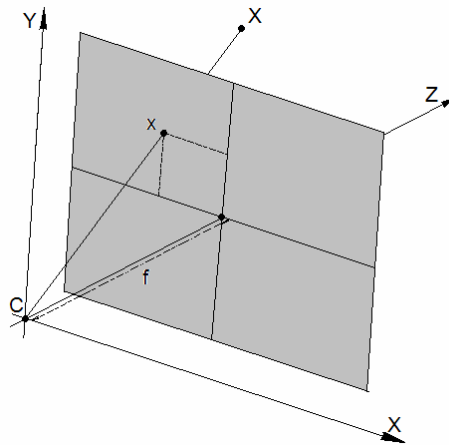
Obrázek 2.3.2: Absolutní duální kvadrika vztažená k rovině v nekonečnu

2.4 Model perspektivní kamery

Nalezení projekčních matic kamer P_i je jedním z hlavních cílů 3D rekonstrukce. Model kamery popisuje, jakým způsobem jsou 3D body kamerou promítnuty do 2D. Jak již napovídá název kapitoly, zaměříme se zejména na model perspektivní kamery, který velmi dobře aproximuje kameru reálnou. V této kapitole nejprve popíšeme model kamery a vlastnosti tohoto modelu, dále se zaměříme na kalibraci kamery, tedy určení parametrů kamery. V poslední části naznačíme jakým způsobem vidí svět reálná kamera nebo lidské oko.

2.4.1 Základní model perspektivní kamery

Základem popisovaného modelu je středové promítání, *optický střed promítání* C umístíme do počátku souřadného systému. Kolmo na osu z leží ve vzdálenosti $z = f$, kde f je *ohnisková vzdálenost*, tzv.: *rovina promítání* (průmětna).



Obrázek 2.4.1: Model perspektivní kamery

Bod (X, Y, Z) je mapována na obrazový bod $(fX/Z, fY/Z, f)$. Vynecháme-li poslední souřadnici, vidíme že

$$(X, Y, Z) \rightarrow (fX/Z, fY/Z) \quad (2.4.1)$$

popisuje mapování z prostorových 3D souřadnic na 2D souřadnice.

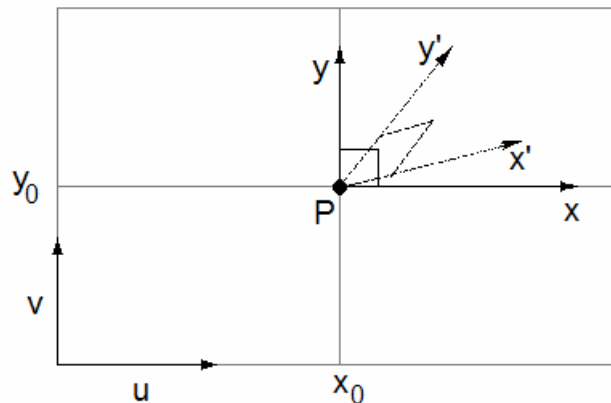
Použijeme-li homogenní souřadnice, můžeme toto v maticové notaci zapsat jako:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.4.2)$$

Osa z se v tomto případě nazývá *hlavní osa kamery* (angl.: *principal ray*). Průsečík hlavní osy kamery a roviny promítání, tedy geometrický střed promítání, se nazývá *hlavní bod*, nebo také *principální bod* (u_0, v_0) (z angl.: *principal point*).

2.4.2 Vnitřní a vnější parametry kamery

Ve výrazu (2.4.1) předpokládáme počátek obrazových souřadnic na průmětně umístěný v principálním bodu, takový souřadnicový systém nazýváme *souřadnicový systém kamery* (x, y). Ve skutečnosti tomu tak nemusí být, v obecném případě je používán tzv. *obrazový* (také *pixelový*) *souřadnicový systém* (u, v). Zatímco bázové vektory obrazového souřadnicového systému jsou na sebe kolmé a stejně dlouhé (čtvercové pixely), u souřadnicového systému kamery toto obecně neplatí



Obrázek 2.4.2: Kamerový a obrazový souřadný systém.

Vztah mezi těmito souřadnými systémy je možno vyjádřit jako

$$\begin{aligned} u &= fx + \beta y + u_0 \\ v &= \alpha y + v_0 \end{aligned} \quad (2.4.3)$$

Zapišeme-li vztah (2.4.3) maticově, získáme kalibrační matici kamery K , které popisují vnitřní parametry kamery:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \underbrace{\begin{bmatrix} f & \beta & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2.4.4)$$

(u_0, v_0) je pozice principálního bodu, α je poměr šířky a výšky pixelů, β vyjadřuje míru zkosení ne-čtvercových pixelů, f je stále ohnisková vzdálenost.

Model kamery se skládá z vnitřních a vnějších parametrů kamery. Vnitřní parametry jsme již popsali a jsou obsaženy v kalibrační matici K . Vnější parametry kamery jsou parametry definující pozici a orientaci kamery a jsou reprezentovány ortogonální maticí $R = [i, j, k]^T$ o rozměrech 3×3 (i, j, k jsou lineárně nezávislé vektory 3×1) reprezentující orientaci kamery a vektorem C určujícím pozici ohniska kamery. Transformační matici kamery P o rozměrech 4×3 z rovnice (2.4.2) pak můžeme rozšířit o vnitřní a vnější parametry takto:

$$P = \mu KR[I|C] = \mu K[R|RC], \quad (2.4.5)$$

kde μ je globální změna měřítka.

Označíme-li $M = KR$ a posun $T = (t_x, t_y, t_z)^T = -RC$, píšeme

$$P = [M|T], \quad (2.4.6)$$

Rozšíříme-li rovnici (2.4.2) o vnitřní a vnější parametry (viz rovnice (2.4.5)) je možno zapsat promítnutí bodu maticí ve tvaru

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \mu \begin{bmatrix} f & \beta & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i & t_x \\ j & t_y \\ k & t_z \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.4.7)$$

f	<i>Ohnisková vzdálenost</i> – vzdálenost mezi ohniskem a průmětnou, změna ohniskové vzdálenosti je nazývána <i>zoom</i>
(u_0, v_0)	<i>Principální bod</i> – průsečík mezi hlavní osou kamery a průmětnou v obrazovém souřadném systému
α	<i>Poměr stran</i> – poměr mezi šířkou a výškou pixelu
β	<i>Zkosení</i> – pixely nemusí být čtvercové, míru zkosení pixelů určuje parametr β . Běžné kamery mají $\beta = 0$

Tabulka 2.4.1: Souhrn vnitřních parametrů kamery.

2.4.3 Kalibrace kamery, IAC a DIAC

Postup nalezení vnitřních parametrů kamery nazýváme *kalibrace kamery*, je-li tento postup automatický používáme termín *autokalibrace*, nebo *samokalibrace* (z angl. self-calibration). V této kapitole ukážeme jakým způsobem souvisí kuželosečky a kvadriky popisované v kapitole (2.3) s kalibrací kamer.

Obraz absolutní kuželosečky ω (z angl.: Image of absolute conic), značíme IAC, je promítnutí absolutní kuželosečky (kuželosečka ležící na rovině v nekonečnu) kamerou P . Obraz absolutní duální kvadriky, tedy duál k IACu značíme DIAC ω^* .

3D body umístěné na nevlastní rovině označíme $X_\infty = \begin{bmatrix} x_\infty & 0 \end{bmatrix}$. Promítneme-li tyto body na průmětnu kamery P dostaneme

$$u^T = PX_\infty^T = K \begin{bmatrix} R|T \\ 0 \end{bmatrix} \begin{pmatrix} x_\infty^T \\ 0 \end{pmatrix} = KRx_\infty^T. \quad (2.4.8)$$

Vidíme, že KR mapuje body z nevlastní roviny do kamery P . Důležitou vlastností rovnice (2.4.8) je, že toto mapování je nezávislé na pozici kamery C . Protože je absolutní kuželosečka umístěná na nevlastní rovině, můžeme aplikovat transformaci z rovnice (2.4.8) a vypočítat tak IAC ω :

$$\omega = (KR)^{-T} \Omega_\infty (KR)^{-1} \quad (2.4.9)$$

V metrickém a eukleidovském prostoru je absolutní kuželosečka určena jednotkovou maticí $I_{(3 \times 3)}$, rovnice (2.4.9) přechází do tvaru

$$\omega = (KR)^{-T} I_{(3 \times 3)} (KR)^{-1} = K^{-T} R R^{-1} K^{-1} = (K K^T)^{-1}, \quad (2.4.10)$$

obdobným postupem můžeme vyjádřit DIAC

$$\omega^* = (KR)I_{(3 \times 3)}(KR)^T = KRR^T K^T = KK^T, \quad (2.4.11)$$

kde $R^T = R^{-1}$. Z tohoto vztahu je jasně patrné, že DIAC je inverzí IACu

$$\omega = (\omega^*)^{-1}. \quad (2.4.12)$$

Rovnice (2.4.10) nám ukazuje, že existuje vztah mezi absolutní kuželosečkou a kalibrační maticí kamery. Podaří-li se nám určit na snímku pořízeném kamerou obraz absolutní kuželosečky ω (nebo ω^*), můžeme Choleského dekompozicí, nebo SVD (viz příloha (A.4)) získat z rovnice (2.4.10) nebo (2.4.11) kalibrační matici K .

To jakým způsobem získáme ω je popsáno v kapitole (6), metrická rekonstrukce.

2.4.4 Určení parametrů kamery z její matice

Mějme matici obecné projektivní kamery P . Naším cílem je nalézt pozici, orientaci a vnitřní parametry kamery P .

Nalezení pozice kamery

Pozice kamery (ohnisko) je bod pro který platí $PC = 0$. Numericky můžeme tuto rovnici vyřešit jako homogenní soustavu např. pomocí SVD (viz příloha (A.1)). Algebraicky můžeme pozici kamery $C = (X, Y, Z, W)^T$ určit jako (viz [Hartley-04]):

$$\begin{aligned} X &= \det([p_2, p_3, p_4]), & Y &= -\det([p_1, p_3, p_4]), \\ Z &= \det([p_1, p_2, p_4]), & W &= -\det([p_1, p_2, p_3]), \end{aligned} \quad (2.4.13)$$

p_1, p_2, p_3 jsou řádky matice P

Nalezení orientace a vnitřních parametrů kamery

Použijeme-li tvar matice kamery z rovnice (2.4.6), tedy $P = [M|T] = [KR|T]$, můžeme snadno rozložit matici M na KR užitím QR dekompozice (viz [Hartley-04]).

$$M_{(3 \times 3)} \xrightarrow{QR} \begin{bmatrix} f & \beta & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix} R_{(3 \times 3)} \quad (2.4.14)$$

2.4.5 Reálná kamera

Perspektivní projekce je idealizovaný matematický model chování reálných kamer, otázkou je, jak dobrý je tento model? Narozdíl od perspektivní projekce, projekce reálné kamery (založené na soustavě čoček) není lineární, podléhá tzv. *radiálnímu zkreslení*. Takže zatímco v perspektivní projekci se přímka v prostoru mapuje na přímku v obraze, u reálné kamery tomu tak není. Radiální zkreslení se projevuje zejména u širokoúhlých čoček (je způsobeno zahnutím čočky), při jejich použití může být radiální zkreslení kritické a vzniká potřeba, tento problém odstranit. Radiální zkreslení je možno zredukovat úpravou výsledného obrazu, existuje celá řada metod, které se zabývají tímto problémem, podrobnosti se je možno dozvědět v [Devernay-95].



Obrázek 2.5.1³: Ukázka radiálního zkreslení.

Model, který by lépe popisoval chování reálné kamery než perspektivní projekce je možno nalézt v publikaci [Kolb-95], ze které je také převzat obrázek (2.5.1).

³ Zdroj obrázku: [Kolb-95]

Kapitola 3

Geometrie více pohledů

Mezi dvojicí, trojicí a více pohledy na scénu existuje vztah, který se pokusíme popsat v této kapitole. Tento vztah je velmi důležitý v kalibraci a 3D rekonstrukci. Velkých pokroků v porozumění těmto vztahům bylo dosaženo v posledních několika letech.

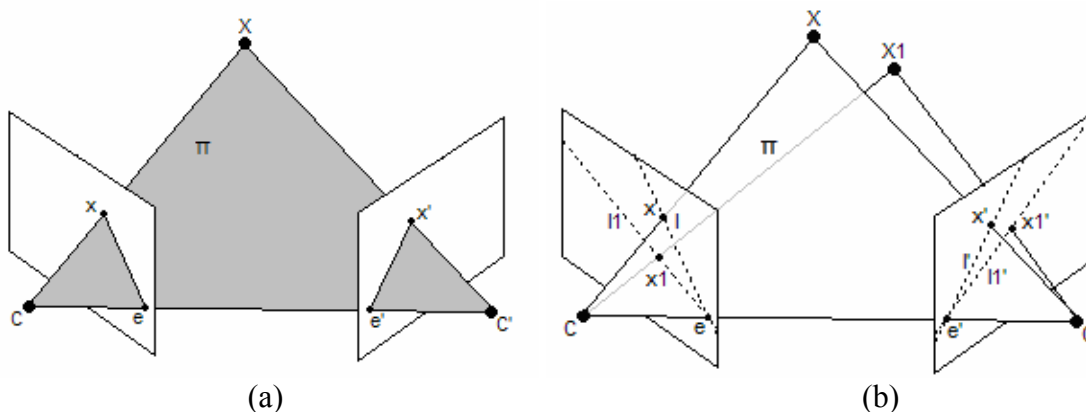
V části (3.1) si popíšeme vztah mezi dvojicí snímků, v (3.2) mezi trojicí a kapitole (3.3) bude věnována popisu vztahu mezi větším počtem snímků. Poslední část bude spíše informativní, protože vztah mezi více než třemi snímky nám již přináší velmi málo nové informace.

3.1 Epipolární geometrie

Epipolární geometrie popisuje základní geometrický vztah mezi dvěma různými perspektivními kamerami, popisuje vnitřní parametry a relativní pozice kamer nezávisle na pozorované scéně [Faugeras-92, Hartley-92, Moons-98, Hartley-04].

Epipolární geometrie závisí pouze na relativní pozici, orientaci a vnitřních parametrech kamer a nezávisí na struktuře scény, popisuje tedy čistě vztah mezi dvojicí kamer.

3.1.1 Geometrická reprezentace



Obrázek 3.1.1: Epipolární geometrie.

Na obrázku (3.1.1) vidíme dvě kamery určené středy promítání C , C' a projekтивními rovinami. Bod v prostoru X společně s C a C' tvoří *epipolární rovinu* Π (angl.: epipolar plane). Protože je *epipolární rovina* kolmá na obě projekтивní roviny kamer, je průmětem *epipolární roviny* přímka l , tzv. *epipolára* (angl.: epipolar line) procházející bodem x a *epipólem* (angl.: epipole) e . Epipól je průmět pozice první kamery do kamery druhé. Což platí analogicky i pro druhou kameru (l' , x' , e').

3.1.2 Algebraická reprezentace

Epipolární geometrie je algebraicky reprezentovaná *fundamentální maticí* F (angl.: fundamental matrix). Je to matice o rozměrech 3×3 s hodnotou 2 definovaná vztahem

$$x'Fx^T = 0, \forall X, \quad (3.1.1)$$

kde X je bod v 3-dimenzionálním prostoru a x, x' jsou homogenní 2D souřadnice průmětů tohoto bodu na první a druhý snímek. Matici F můžeme také označit jako *bifokální tenzor*, toto označení bude dávat větší smysl v kontextu dalších kapitol zabývajících se geometrií více pohledů (více než dvou).

Rovnice

$$l = x'F, l' = Fx^T \quad (3.1.2)$$

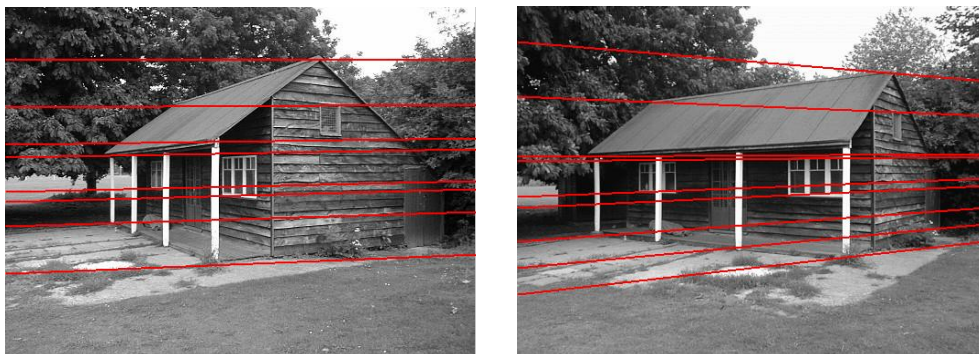
definuje rovnici čáry odpovídající bodu x na druhém obrázku, opět v homogenních souřadnicích, tedy epipoláru l . Epipoláry mají rozsáhlé využití při hledání korespondencí nebo registrace obrazů. Při znalosti epipolární geometrie je možno redukovat hledání korespondence na hledání na přímce (hledáním korespondencí se zabývá kapitola (4)).

Epipóly jsou definovány jako

$$e'F = 0, Fe^T = 0. \quad (3.1.3)$$

Všechny epipoláry se protínají v epipólech.

Epipolární geometrie má 7 stupňů volnosti, čtyři reprezentují souřadnice *epipólů* na každém obrázku $e = [x_e, y_e], e' = [x_{e'}, y_{e'}]$ a zbývající tři projektivitu mezi svazky *epipolár* na obou průmětech.



Obrázek 3.1.2⁴: Epipolární geometrie, čáry reprezentují epipoláry.

⁴ Zdroj obrázku: URL=<http://www.duke.edu/~kmg/cps296/report.html>

3.1.3 Výpočet fundamentální matice ze známých kamer

Určení fundamentální matice, známe-li matice kamer P, P' je poměrně jednoduché, stačí dosadit do následujícího vzorce

$$F = [e']_x P' P^+ \quad (3.1.4)$$

kde e' je epipól, tedy $e' = P'C$ pro $PC = 0$ (C je optický střed kamery P). P^+ je pseudo-inverze matice P (viz příloha (A.2)). Význam symbolu $[a]_x$ je následující: pro tříprvkový vektor $a = (a_1, a_2, a_3)^T \in R^3$, značí $[a]_x$ antisymetrickou matici o rozměrech 3×3 :

$$[a]_x = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}, \quad (3.1.5)$$

která reprezentuje vektorový součin s a , $[a]_x v \approx a \times v$, $\forall v \in R^3$. Pro nenulové hodnoty a má $[a]_x$ hodnost 2.

3.1.4 Výpočet fundamentální matice z korespondencí

V případě, že matice kamer P, P' nejsou známy, je možno získat fundamentální matici ze známých korespondencí.

Lineární 8-bodový algoritmus

8-bodový algoritmus počítá fundamentální matici z osmi nebo více korespondencí. Fundamentální matice je definována rovnicí (3.1.1): $x' F x^T = 0$. Je-li $x' = (u', v', 1)$ a $x = (u, v, 1)$, pak můžeme psát

$$uu'f_{11} + uv'f_{21} + uf_{31} + vu'f_{12} + vv'f_{22} + vf_{32} + u'f_{13} + v'f_{23} + f_{33} = 0. \quad (3.1.6)$$

Řádky soustavy rovnic jsou pak reprezentovány jako vektor $(uu', uv', u, vu', vv', v, u', v', 1)$, z osmi korespondujících bodů můžeme sestavit soustavu lineárních rovnic ve tvaru

$$Af = 0, \quad (3.1.7)$$

kde f je 9-ti prvkový vektor $f = (f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33})$ odpovídající fundamentální matici

$$F = \begin{pmatrix} f_{11} & f_{21} & f_{31} \\ f_{21} & f_{22} & f_{32} \\ f_{31} & f_{32} & f_{33} \end{pmatrix}.$$

Rovnice (3.1.7) má triviální řešení $f = 0$, které nás nezajímá, přidáváme proto podmínku zajišťující nenulové řešení $\|f\|=1$ (nebo $f_{33}=1$). Hodnost matice A musí být 8, protože však má A devět sloupců a počet řádek není shora omezen, nastane vlivem šumu ve většině případů situace kdy je hodnost rovna devíti, neexistuje tedy přesné nenulové řešení rovnice $Af=0$. Hledáme tedy, užitím metody nejmenších čtverců, vektor f , který minimalizuje $\|Af\|$ za podmínky $\|f\|=f \cdot f^T=1$. Řešením tohoto problému můžeme získat například užitím metody SVD, viz příloha (A.1).

Po vyřešení rovnice (3.1.7), složíme vektor f zpět do matice, čímž získáme fundamentální matici F . Takto získaná matice F většinou nemá, opět vlivem šumu, hodnost 2. To se projeví tak, že se epipoláry neprotínají v jednom bodě, hledáme tedy takovou matici F' , která má hodnost 2 a zároveň se minimálně liší od původní matice F . Matici F' je možno nalézt například pomocí SVD (viz příloha (A.3)). Epipoláry matice F' se nyní protínají v epipólu a neprocházejí již zcela přesně zadanými body.

Hlavním problémem osmibodového algoritmu je fakt, že není invariantní vůči posunutí a změně měřítka. Jsou-li například rozměry obrázku 1000×1000 . Pak každý řádek matice A může nabývat hodnot v řádech

$$(10^6 \quad 10^6 \quad 10^3 \quad 10^6 \quad 10^6 \quad 10^3 \quad 10^3 \quad 10^3 \quad 1).$$

Chyba ve vstupních datech a chyba způsobená úpravou matice F na F' se rozloží do všech sloupců rovnoměrně, tedy někde představuje minimální změnu a jinde změnu dosti významnou. Tento problém je možno vyřešit tzv. *normalizací* (viz [Hartley-95]), kde se všechny body před začátkem výpočtu transformují tak, aby jejich těžiště bodů bylo v počátku souřadnic (posun) a aby průměrná eukleidovská vzdálenost bodů od počátku byla $\sqrt{2}$ (změna měřítka).

Nelineární 7-bodový algoritmus

Minimální počet bodů potřebných k sestrojení fundamentální matice je právě sedm. Najdeme tedy taková dvě řešení f_1 a f_2 , aby platilo $\|f_1\|=\|f_2\|=1$ a $Af_1=Af_2=0$, která odpovídají maticím F_1 a F_2 . Pak řešením je každé F pro které platí

$$F = \alpha F_1 + (1 - \alpha) F_2. \quad (3.1.8)$$

Protože matice F musí mít hodnost 2 přidáme nelineární podmínku

$$\det(F) = \det(\alpha F_1 + (1 - \alpha) F_2) = 0. \quad (3.1.9)$$

Získáme tedy kubickou rovnici pro α , která má jedno až tři řešení, ze kterých můžeme dle rovnice (3.1.8) získat stejný počet fundamentálních matic F .

3.1.5 Měření chyb

Přítomnost šumu způsobuje, že rovnice (3.1.1) neplatí přesně, je tedy potřeba určit do jaké míry odpovídá korespondence danému modelu epipolární geometrie. Dokážeme-li správně definovat chybu, můžeme její minimalizací dosáhnout zlepšení výsledků. V této kapitole ukážeme princip určení nejčastěji používaných chybových funkcí.

Algebraická chyba

Algebraická chyba nemá žádný geometrický význam. Algebraickou chybu vyjádříme jako druhou mocninu algebraické vzdálenosti:

$$d_A^2(x, x') = (x'^T Fx)^2. \quad (3.1.10)$$

Určení velikosti algebraické chyby je minimálně výpočetně náročné, a proto velmi rychlé. Můžeme ji tedy použít k počátečnímu odhadu chyby.

Reprojekční chyba

K zadané korespondenci x, x' , která přesně nevyhovuje modelu epipolární geometrie hledáme dvojici \hat{x}, \hat{x}' , která rovnici (3.1.1) splňuje. A zároveň požadujeme, aby součet kvadrátů vzdáleností bodů x od x' a \hat{x} od \hat{x}' byl minimální:

$$\min_{\hat{x}, \hat{x}'} (d(x, \hat{x})^2 + d(x', \hat{x}')^2), \text{ za podmínky že } \hat{x}'^T F\hat{x} = 0,$$

kde $d(*, *)$ je eukleidovská vzdálenost.

Reprojekční chybu pak definujeme následovně:

$$d_R^2(x, x') = d(x, \hat{x})^2 + d(x', \hat{x}')^2. \quad (3.1.11)$$

Vztah mezi reprojekční a algebraickou chybou je podrobně popsán v [Hartley-04]. V [Hartley-94a] je dokázáno, že reprojekční chybu je možno přesně vyjádřit polynomem stupně šest. Protože je výpočet reprojekční chyby výpočetně náročný, používá se k určení chyby její aproximace, tzv. Sampsonova vzdálenost.

Sampsonova vzdálenost

Sampsonova vzdálenost je aproximací reprojekční chyby prvního řádu. Je-li $x = (u, v)$ a $x' = (u', v')$, d_A^2 je algebraická vzdálenost definovaná v (3.1.10) a $r_u^2, r_v^2, r_{u'}^2, r_{v'}^2$ jsou parciální derivace algebraické vzdálenosti:

$$\begin{aligned} r_u^2 &= f_1 x' + f_4 y' + f_7, \\ r_v^2 &= f_1 x + f_2 y + f_3, \\ r_{u'}^2 &= f_2 x' + f_5 y' + f_8, \\ r_{v'}^2 &= f_4 x + f_5 y + f_6, \end{aligned} \quad (3.1.12)$$

$f_1 - f_8$ v rovnici (3.1.12) jsou prvky fundamentální matice F , pak Sampsonovu vzdálenost vyjádříme jako:

$$d_s^2(x, x') = \frac{d_A^2}{r_u^2 + r_v^2 + r_{u'}^2 + r_{v'}^2} \quad (3.1.13)$$

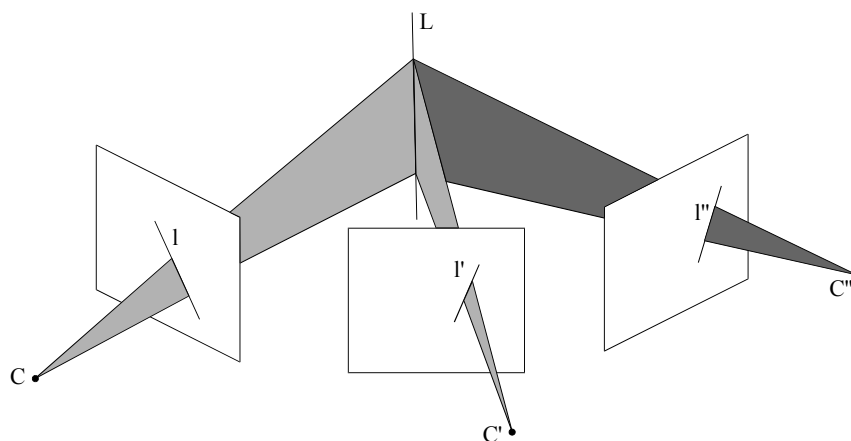
Jak ukazuje např. [Chum-01], je Sampsonova vzdálenost velmi dobrou aproximací reprojekční chyby.

3.2 Trifokální tensor

Pod pojmem *trifokální tensor* [Hartley-94b, Shashua-94, Torr-97, Moons-98, Hartley-04], nebo *trifokální geometrie* rozumíme obdobu epipolární geometrie pro trojici snímků. Pro větší počet snímků je možno formulovat tzv. kvadrifokální, obecně multifokální tensor, jejich určení je však poměrně výpočetně náročné a neposkytují nám již mnoho dalších informací. Narozdíl od Epipolární geometrie dokáže trifokální geometrie pracovat nejen s korespondujícími body, ale také s korespondujícími přímkami. Obdobně jako u epipolární geometrie závisí trifokální tensor pouze na parametrech kamer a jejich vzájemné pozici a nikoli na struktuře scény.

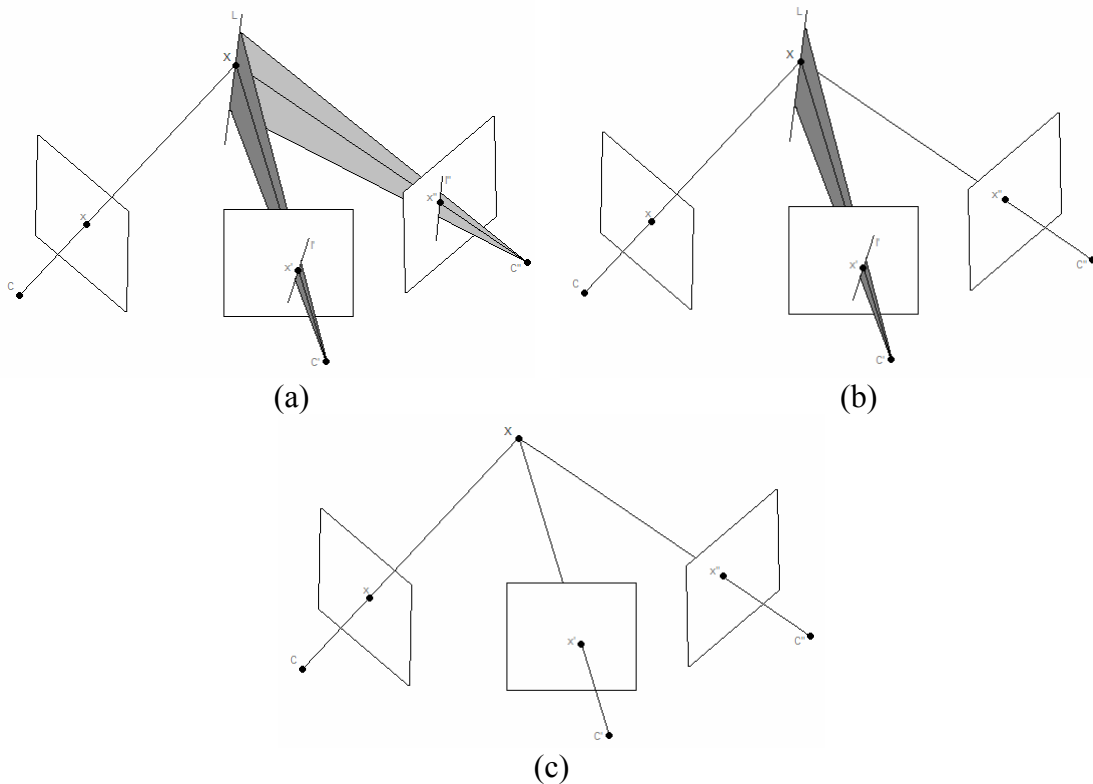
Problematika trifokálních tensorů je velmi rozsáhlá a tato práce ji není schopna celou obsáhnout, našim cílem bude vysvětlení podstaty trifokálních tensorů (3.2.1, 3.2.2), ukážeme jak je možno z trifokálního tensoru získat matice kamer a fundamentální matice (3.2.3) a naopak (3.2.4), zároveň naznačíme jejich výpočet z bodových korespondencí (3.2.5). Pro hlubší studium můžeme doporučit zejména publikaci [Hartley-04].

3.2.1 Geometrická reprezentace



Obrázek 3.2.1: Trifokální geometrie určená trojicí přímek $l \leftrightarrow l' \leftrightarrow l''$.

Na obrázku (3.2.1) vidíme tři kamery určené středy promítání C , C' , C'' a projekčními rovinami. Toto je jedna z možností jakým způsobem je možno geometricky zobrazit vztah mezi třemi kamerami. Tento vztah však můžeme ukázat i na odlišných konfiguracích, než je *přímka-přímka-přímka*, tyto konfigurace jsou zobrazeny na obrázcích (3.2.2).



Obrázky 3.2.2: Různé způsoby reprezentace geometrického vztahu mezi dvěma kamerami. (a) obsahuje trojici průmětů bodu x : $x \leftrightarrow x' \leftrightarrow x''$ a dvě přímky procházející body x' a x'' , tedy $x \leftrightarrow l' \leftrightarrow l''$ znamená korespondenci bod-přímka-přímka. podobně je na (b) určen vztah trojicí *bod-přímka-bod* a na (c) *bod-bod-bod*.

3.2.2 Algebraická reprezentace

Trifokální tensor můžeme popsat sadou tří matic $[T_1, T_2, T_3]$ o rozměrech 3×3 , tento tvar trifokálního tenzoru označujeme jako *trifokální tensor v maticové notaci*. Trifokální tensor má pouze 18 stupňů volnosti, tedy ne každý tensor o rozměrech $3 \times 3 \times 3$ popisuje vztah mezi trojicí snímků.

Jak jsme si již ukázali v kapitole (3.2.1) můžeme trifokální tensor získat z různých konfigurací, jeho definice je tedy také možná více způsoby. Vztahy definující trifokální tenzory shrneme v tabulce (3.2.1).

(i) korespondence *přímka-přímka-přímka*

$$l'[T_1, T_2, T_3]l''^T = l \quad \text{nebo} \quad (l'[T_1, T_2, T_3]l''^T)[l]_x = 0^T \quad (3.2.1)$$

(ii) korespondence *bod-přímka-přímka*

$$l' \left(\sum_i x^i T_i \right)^T l''^T = 0_{(3 \times 1)} \quad \text{pro korespondenci } x \leftrightarrow l' \leftrightarrow l'' \quad (3.2.2)$$

(iii) korespondence *bod-přímka-bod*

$$l' \left(\sum_i x^i T_i \right) [x'']_x = 0_{(3 \times 1)} \quad \text{pro korespondenci } x \leftrightarrow l' \leftrightarrow x'' \quad (3.2.3)$$

(iv) korespondence *bod-bod-přímka*

$$[x']_x \left(\sum_i x^i T_i \right)^T l''^T = 0_{(3 \times 1)} \quad \text{pro korespondenci } x \leftrightarrow x' \leftrightarrow l'' \quad (3.2.4)$$

(v) korespondence *bod-bod-bod*

$$[x']_x \left(\sum_i x^i T_i \right)^T [x'']_x = 0_{3 \times 3} \quad (3.2.5)$$

Tabulka 3.2.1⁵: Shrnutí trifokálních tensorů pro různé konfigurace s užitím maticové notace. Symbol $[a]_x$ reprezentuje vektorový součin a je definován rovnicí (3.1.5). $0_{(3 \times 1)}$ označuje nulový vektor o rozměrech (3×1) .

3.2.3 Výpočet epipolární geometrie a matic kamer z trifokálního tenzoru

Mějme trifokální tensor v maticové notaci $[T_1, T_2, T_3]$,

Nejprve vypočteme souřadnice epipólů e' , e'' . Dále vypočteme vektory u_i , v_i pro které platí $u_i T_i = 0_{(3 \times 1)}$, $T_i v_i = 0$. Epipóly jsou získány řešením soustav

$$\begin{aligned} e' [u_1, u_2, u_3]^T &= 0 \\ e'' [v_1, v_2, v_3]^T &= 0 \end{aligned} \quad (3.2.6)$$

⁵ Zdroj tabulky: [Hartley-04]

pak fundamentální matice F_{21} , F_{31} vypočteme jako

$$\begin{aligned} F_{21} &= [e']_{\times} [T_1, T_2, T_3] e''^T \\ F_{31} &= [e'']_{\times} [T_1^T, T_2^T, T_3^T] e'^T. \end{aligned} \quad (3.2.7)$$

Epipóly e' , e'' normalizujeme na jednotkovou normu, pak matice kamer P' , P'' za podmínky $P = [I|0]$, vypočteme jako

$$\begin{aligned} P' &= \left[[T_1, T_2, T_3] e''^T \mid e' \right] \\ P'' &= \left[(e'' e''^T - I) [T_1^T, T_2^T, T_3^T] e'^T \mid e''^T \right]. \end{aligned} \quad (3.2.8)$$

3.2.4 Výpočet trifokálního tensoru z matic kamer

Je-li první kamera v kanonickém tvaru $P = [I|0]$, další dvě kamery $P' = [a_j^i]$, $P'' = [b_j^i]$, trifokální tensor můžeme vypočítat ze vztahu

$$T_i = a_i b_4^T - a_4 b_i^T, \quad (3.2.9)$$

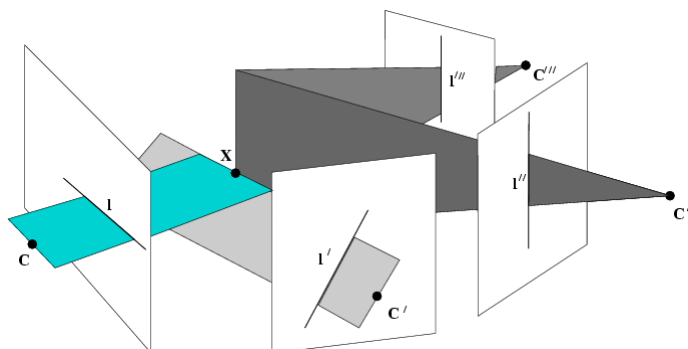
kde a_i je i -tý řádek matice P' a b_i je i -tý řádek matice P'' .

3.2.5 Výpočet trifokálního tensoru z korespondencí

Již jsme ukázali, že trifokální tensor můžeme vypočítat jak z bodových tak i z přímkových korespondencí. Výpočet trifokálního tensoru je podobný tomu jak jsme počítali fundamentální matici. Podle toho které korespondence použijeme k výpočtu trifokálního tensoru vybereme danou definici z tabulky (3.2.1), po roznásobení získáme lineární soustavu rovnic, kde neznámé jsou jednotlivé prvky tensoru. Tuto homogenní soustavu vyřešíme metodou SVD (viz Příloha (A.1)).

Abychom mohli vypočítat trifokální tensor lineárním algoritmem potřebujeme 7 korespondencí (trojic). Minimální počet bodů je 6, podobně jako u epipolární geometrie musíme přidat nelineární podmínku založenou na hodnotě tensoru. To, že k výpočtu trifokálního tensoru postačuje pouze 6 bodů oceníme zejména v kapitole (4.3 RANSAC), kde na počtu nezbytných korespondujících bodů nepřímo (čím méně bodů, tím rychlejší) závisí rychlost algoritmu.

3.3 Kvadrifokální a multifokální tensory



Obrázek 3.3.1⁶: Kvadrifokální tensory

Kvadrifokální tensor popisují vzájemný vztah 4 kamer, multifokální obecně více kamer. Výpočet těchto tensorů je poměrně komplikovaný. V této práci se o těchto tensorech zmiňujeme pouze pro úplnost. Minimálním počtem bodů potřebných k výpočtu kvadrifokálního tensoru (a zároveň všech dalších tensorů) je 6, obdobně jako u trifokálních tensorů, proto nám již kvadrifokální tensor poskytuje již jen o velmi málo informací více než tensor trifokální.

Při popisu tensorů vyšších řádů si již maticová notace nedostačuje a je potřeba zavést tensorovou notaci, příklad tensorové notace pro trifokální tensor:

$$\text{Trifokální tensor v maticové notaci } [T_1, T_2, T_3]: \quad l' \left(\sum_i x^i T_i \right) l''^T = 0$$

$$\text{Trifokální tensor v tensorové notaci } T_i^{jk}: \quad x^i l_j'^T l_k''^T T_i^{jk} = 0$$

Jak již bylo napsáno, nebudeme se podrobně zabývat multifokálními tensorů vyšších řádů (vyšších než 3), pro podrobnější studium této problematiky (odvození a výpočet) doporučujeme nahlédnout do publikace [Hartley-04].

⁶ Zdroj obrázku: [Hartley-04]

Kapitola 4

Nalezení korespondencí mezi snímky

Vzájemné přiřazení korespondujících objektů je nezbytný základ pro další metody 3D rekonstrukce scény s neznámými parametry kamer. Nalezením korespondence rozumíme identifikaci stejného objektu na několika (dvojici, trojici, ...) snímcích. Protože je problém korespondence objektů v obecné rovině poměrně složitý, bývá většinou redukován na hledání společných bodů, úseček nebo jejich kombinace, které je možno jednoduše matematicky popsat a následně využít. Jak jsme se již dozvěděli v kapitolách (3.4, 3.5), epipolární geometrie dokáže využít pouze informace o korespondujících bodech, trifokální tensorů již ze své definice dokáží pracovat jak s body, tak i s úsečkami.

V této práci se budeme zabývat výhradně hledáním společných bodů. Nebude-li uvedeno jinak je následující postup aplikován na dvojici snímků získaných perspektivní projekcí z 3D scény.

Proces hledání korespondujících bodů je možno rozdělit do tří základních kroků, které budou v této kapitole podrobně popsány:

- 1, Detekce bodů
- 2, Získání inicializačních korespondencí
- 3, Finální zpřesnění korespondencí získaných v kroku 2

V prvním kroku se na každém snímku (nezávisle na ostatních snímcích) detekují body u kterých vyžadujeme co největší invarianci vzhledem ke geometrickým transformacím (konkrétně afinní transformace a perspektivní projekce). Mohou to být například: *rohů objektů, průsečíky čar, maxima křivosti obrysů, těžiště objektů*, apod. Nejdůležitější vlastností bodového detektoru je aby nalezené body na prvním snímku byly, v co největší míře, detekovány i na druhém snímku (tzv. *repeatibilita*).

V druhém kroku se naleznou korespondence podle lokálního chování snímku v okolí bodů. Okolí bodu je popsáno několika invariantními (afině případně projektivně invariantními) příznaky, které jsou k bodu přidruženy a na jejichž základě se hledá nejpravděpodobnější kandidát na korespondenci ve druhém snímku. Takto získaná množina potenciálně korespondujících bodů se předává ke zpracování do dalšího kroku. Mezi takto získanými korespondencemi bude jistě velké množství korespondencí přiřazeno chybně, právě tyto chybné korespondence odstraňujeme ve třetím kroku jednou z finálních metod. Je-li jako finální metoda použit algoritmus RANSAC (kapitola (4.3)), stačí ke správnému výsledku, aby přibližně 40% inicializačních korespondencí bylo správných. Právě tato část má největší vliv na automatické určení korespondencí a tím i na výsledek vlastní rekonstrukce.

Třetím krokem je aplikace robustní metody, využívající geometrických vlastností projekce (globální řešení na základě informace o pozici bodů). V této části jsou *inicializační korespondence* rozděleny na množinu správných korespondencí (tzv. *inliers*) a chybných, které jsou vyloučeny (tzv. *outliers*). Vybrané metody finálního odhadu korespondencí jsou popsány v kapitolách (4.3), (4.4), (4.5) s důrazem na již zmiňovaný algoritmus RANSAC.

Mezi druhým a třetím krokem nemusí být obecně pevná hranice a získané výsledky mohou být zpřesňovány, nebo doplňovány opakovaným použitím metod.

Popsaný postup by měl být formálně doplněn ještě jedním krokem, který by na základě známých (v předchozích krocích získaných) korespondencí dokázal určit další korespondence. Čím máme větší množství korespondencí, tím více je možno potlačit šum obsažený v pozicích detekovaných bodů (vzniklý šumem na samotném snímku), což výrazně zlepšuje výsledky dalších kroků 3D rekonstrukce.

4.1 Detekce významných bodů

Detekce bodů je prvním krokem při hledání korespondencí a tedy i prvním krokem vlastní 3D konstrukce. Kvalita získaných bodů je tedy faktorem významně ovlivňujícím celou rekonstrukci. Od detekovaných bodů vyžadujeme, aby body získané z jednoho snímku byly, v co možná největší míře, zastoupeny i na druhém snímku, tuto vlastnost nazýváme *repeatibilita* (opakovatelnost).

V této kapitole se zaměříme zejména na dnes již klasický Harrisův kombinovaný rohový a hranový detektor [Harris-88], který je jednoduše realizovatelný, rychlý a dosahuje velmi dobrých výsledků.

Protože od roku 1988 již uplynula poměrně dlouhá doba, vznikla mezitím celá řada vylepšení základního Harrisova detektoru (např.: [Schmid-98, Montesinos-98]) a několik odlišných metod dosahujících nepatrně lepších výsledků. Za všechny zde budeme prezentovat algoritmus FindFP [Zítová-00], který vykazuje výborné výsledky zejména u rozmazaných (smooth) obrázků, což je možno s výhodou využít u snímků zatížených šumem tak, že obrázky nejprve vyhladíme.

Většinou jsou různé metody založeny na různých principech a nalézají proto odlišné body, s tím že zachovávají vysokou *repeatibilitu*. Toho je možno s výhodou využít tak, že zkombinujeme výstupy několika metod.

4.1.1 Harrisův operátor

Jedním z nejpoužívanějších bodových detektorů je kombinovaný hranový a rohový Harrisův detektor [Harris-88], který dosahuje velmi dobrých výsledků. Harrisův detektor hledá takové body, v jejichž okolí nastává maximální změna jasu, k čemuž využívá tzv. *strukturální matice* (angl.: structural matrix, nebo také second moment matrix).

Popis algoritmu

Vstupy Harrisova detektoru:

- I – obrazová funkce o rozměrech $M \times N$.
- σ_D – rozptyl gaussovského filtru, tzv. *derivační měřítko* (angl.: derivation scale).
- σ_I – rozptyl gaussovského filtru, tzv. *integrační měřítko* (angl.: integration scale).

Prvním krokem algoritmu je výpočet gradientů pro všechny body obrazu, protože je výpočet gradientů náchylný na přítomnost šumu, použijeme nejprve gaussovský filtr na vyhlazení vstupního obrazu.

$$I' = I \otimes G_{\sigma_D}, \quad (4.1.1)$$

kde

$$G_{\sigma_D}(x, y) = \frac{1}{2\pi\sigma_D^2} \exp\left(\frac{-x^2 - y^2}{2\sigma_D^2}\right). \quad (4.1.2)$$

K výpočtu gradientů se použijeme následující aproximaci.

$$\begin{aligned} X &= I' \otimes \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} = \frac{\partial I}{\partial x} \\ Y &= I' \otimes \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^T = \frac{\partial I}{\partial y} \end{aligned} \quad (4.1.3)$$

Sestavení *strukturální matice*:

$$M = G_{\sigma_I} \otimes \begin{bmatrix} X^2 & XY \\ XY & Y^2 \end{bmatrix}. \quad (4.1.4)$$

Symbol \otimes znamená konvoluci.

Vlastní čísla matice M : α, β určují největší a nejmenší změnu jasu v okolí, vlastní vektory potom směr ve kterém je změna jasu největší a nejmenší.

Harris navrhl následující funkci (tzv. *response function*), vyhodnocující vlastní čísla α, β :

$$R = \alpha\beta - k * (\alpha + \beta)^2, \quad (4.1.5)$$

kde $k = 0.04$ je konstanta určená pozorováním.

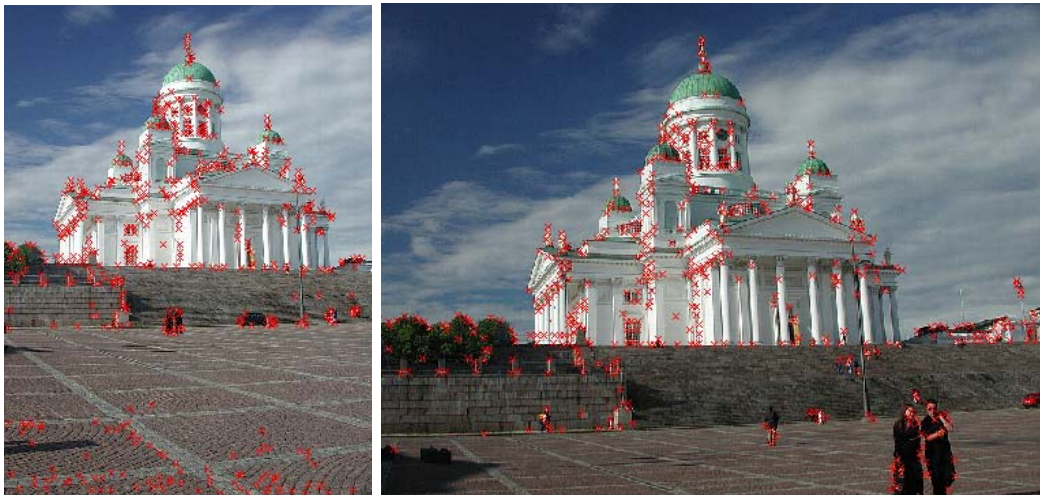
Užití následujících vztahů umožní vyhodnocení funkce bez nutnosti počítat vlastní čísla.

$$T_r(M) = \alpha + \beta = A + B, \quad (4.1.6)$$

$$Det(M) = \alpha\beta = AB - C^2, \quad (4.1.7)$$

$$R = Det - k * T_r^2. \quad (4.1.8)$$

Hodnota R je kladná v rozích objektů, záporná na hranách a malá v jasově konstantních oblastech. Body zájmu tedy získáme jako lokální maxima funkce $R(x,y)$.



Obrázek 4.1.1 Ukázka výstupu Harrisova detektoru

Základní Harrisův detektor pracuje s šedotónovým snímkem, existuje však i barevná (resp. RGB) verze Harrisova detektoru (viz [Montesinos-1998]). Postačuje použít strukturální matici ve tvaru:

$$M = \begin{bmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{bmatrix}, \quad (4.1.9)$$

kde R_x, R_y jsou parciální derivace červené barevné složky (získané obdobně jako X, Y , v rovnici (4.1.3)). Stejným způsobem jsou definovány G_x, G_y, B_x, B_y , pro zelenou, respektive modrou složku.

4.1.2 FindFP

Název tohoto algoritmu vychází z anglického *Find Feature Points*, tedy hledání významných bodů. Tento algoritmus byl prezentován v [Zitová-00]. Podobně jako Harrisův detektor hledá body zejména na rozích objektů, jeho přístup jak tyto body nalézt je však odlišný: algoritmus provádí hranovou detekci a následně vybírá ty body, ve kterých se sbíhají právě dvě hrany svírající ostrý úhel (rohy objektů).

Popis algoritmu

Vstupy algoritmu FindFP:

- I – obrazová funkce o rozměrech $M \times N$.
- N_{FP} – požadovaný počet detekovaných bodů.
- M – poloměr okolí pro výpočet průměrovaných hodnot.
- r – poloměr okolí pro výpočet znaménkových změn.
- d_a – určuje úhel, který může být maximálně svírán dvěma hranami v detekovaném bodě.
- s – minimální vzdálenost mezi dvěma kandidáty na detekovaný bod.
- d_s – maximální dovolená změna křivosti u přímkových kandidátů.
- t – minimální vzdálenost mezi dvěma detekovanými body.

Nejprve zinicilizujeme pole C o rozměrech $M \times N$.

Výpočet funkce J obsahující lokální průměry I

$$J(i, j) = \frac{1}{\pi M^2} \sum_{\Omega_{i,j}, M} I(k, l), \quad (4.1.10)$$

kde $\Omega_{i,j}, M$ je kruhové okolí bodu (i, j) o poloměru M .

Výpočet váhové funkce lokálních změn:

$$W(i, j) = \sum_{\Omega_{i,j}, M} (I(k, l) - J(i, j))^2, \quad (4.1.11)$$

Detekce kandidátů na detekovaný bod: Okolo každého bodu (i, j) vytvoříme kružnici R o poloměru r : $R = \{(i_1, j_1), \dots, (i_k, j_k)\}$, kde $i_1 = i$, $j_1 = j + r$ a další body následují po směru hodinových ručiček. Vypočteme počet znaménkových změn $N_{sc}(i, j)$ v sekvenci:

$$I(i_1, j_1) - J(i, j), \dots, (i_k, j_k) - J(i, j), I(i_1, j_1) - J(i, j) \quad (4.1.12)$$

Je-li $N_{sc}(i, j)$ rovno dvěma, určíme pozici znaménkových změn (i_a, j_a) , (i_b, j_b) a vypočteme úhel daný těmito body $\alpha_{i,j} = \text{angle}((i_a, j_a), (i, j), (i_b, j_b))$.

Je-li $|\alpha_{i,j} - \pi/2| < d_a$ je bod (i, j) kandidátem na detekovaný bod a přiřadíme $C(i, j) = 1$.

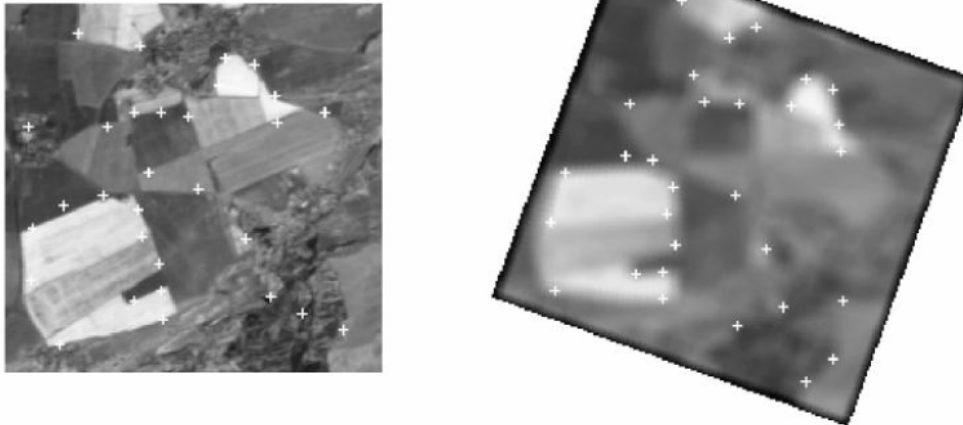
Odstranění nevyhovujících kandidátů: Pro každý bod (i, j) , kde $C(i, j) = 1$ zjistíme, zda v jeho okolí existuje bod (i_f, j_f) takový, že jeho vzdálenost od bodu (i, j) je menší než s . Je-li $N_{sc}(i_f, j_f) = 2$ a zároveň $|\alpha_{i_f, j_f} - \pi| < d_s$, pak $C(i, j) = 0$.

Výběr významných bodů: N_{FP} -krát zopakujeme hledání takového bodu (i_0, j_0) , že

$$(i_0, j_0) = \arg \max_{i,j:C(i,j)=1} W(i, j) \quad (4.1.12)$$

Nastavíme $P_m = (i_0, j_0)$. Pro každý bod (i, j) jehož vzdálenost od (i_0, j_0) je menší než t nastavíme $W(i, j) = 0$.

Výsledkem je sekvence $P_1, \dots, P_{N_{FP}}$ obsahující souřadnice detekovaných významných bodů.



Obrázek 4.1.2⁷: Ukázka výstupu algoritmu FindFP

⁷ Zdroj obrázku: [Zitová-00]

4.2 Inicializační korespondence

V této kapitole prezentujeme rychlou a za určitých okolností velmi účinnou metodu získání odhadu korespondencí ((4.2.2) MEM-Pairs). Tato metoda však není tím, čeho bychom chtěli dosáhnout, jejím hlavním problémem je že není invariantní vzhledem k afinní transformaci. Řešení tohoto problému naznačíme v části (4.2.3).

4.2.1 Korelace

Korelační metoda je jednou z nejpoužívanějších metod na zjištění podobnosti dvou okolí. Na okolí detekovaných bodů pohlíží jako na n -výskytů náhodné veličiny. Nejprve uspořádá čtvercové okolí od vektoru v . To nakolik si dvojice okolí odpovídá poté vyjádříme tzv. korelačním koeficientem, neboli kosinem úhlu mezi těmito vektory:

$$r = C(v_i, v_j) = \frac{(v_i - \bar{v}_i)(v_j - \bar{v}_j)^T}{\|v_i - \bar{v}_i\| \|v_j - \bar{v}_j\|}, \quad (4.2.1)$$

kde \bar{v}_i a \bar{v}_j jsou střední hodnoty vektorů v_i, v_j .

Protože hodnota korelačního koeficientu představuje kosinus, bude se její hodnota pohybovat na intervalu $\langle -1, 1 \rangle$. jednička znamená, že vektory jsou shodné (nulový úhel) a čím menší číslo, tím rozdílnější okolí jsou. Jako korespondující dvojici tedy vybereme ty body jejichž korelační koeficient se bude nejvíce blížit k jedné.

Korelační metoda je jednoduchá, rychlá a invariantní vzhledem ke změně jasu, jejím hlavním problémem je, že s výjimkou posunu není invariantní vůči afinním transformacím.

4.2.2 MEM-Pairs

Metoda MEM-Pairs byla převzata z [Lundberg-00], a je použita ve vlastní implementaci. Zkratka MEM znamená Maximum Entropy Matching, tato metoda je tedy založená na hledání bodů jejichž okolí se vyznačuje stejnou velikostí entropie.

Princip metody je velmi jednoduchý:

1. Nejprve vygeneruje množinu N dvojic (Pairs) náhodných bodů v okolí zkoumaného bodu. N volíme v řádu několika stovek.

Následující postup provedeme pro každý detekovaný bod, se stejnou množinou dvojic, ty jsou tedy náhodně vygenerované, ale pro všechny body společné:

2. Vytvoří bitový vektor o velikosti N , tzv. bitset.
3. Pro každou dvojici bodů porovná jasovou hodnotu obrazu v prvním bodě s jasovou hodnotou obrazu v bodě druhém je-li větší
 - a. příslušnou pozici v bitsetu označíme 1
 - b. jinak 0

Máme-li určeny bitsety všech bodů, přistoupíme ke kroku 4:

4. K bodu na prvním obrázku přiřadíme ten bod na druhém obrázku, jehož množina příznaků (bitset) se liší v minimálním počtu „bitů“.

Protože nejsou porovnávány absolutní hodnoty jasu, ale pouze je-li v jednom bodě větší-či menší než ve druhém, je tato metoda invariantní vzhledem k jasovému posunu.

Tato metoda je alternativou ke korelačním metodám, má tedy stejné problémy s afinními transformacemi jako korelační metody, její hlavní výhodou je, že je přibližně 3 krát rychlejší než nejrychlejší korelační metoda, při stejných výsledcích. Oproti korelační metodě nepracuje s celým okolím, ale pouze s několika (N) náhodně vybranými body tohoto okolí.

4.2.3 Určení Afinní transformace

Tato metoda je doplňková k metodám popisující okolí detekovaných bodů. Využívá strukturální matice, jak jsme ji popsali u Harrisova detektoru. Její základní princip je nalézt takové okolí bodu, které je invariantní vzhledem k afinní transformaci.



Obrázek 4.2.1⁸: Ukázka afinně invariantních okolí bodů zájmu, okolí jsou reprezentována elipsami.

Podrobnosti k metodám založených na hledání afinní transformace, nebo na hledání afinně invariantního okolí je možno získat například z publikací [Lindeberg-97, Baumberg-00, Mikołajczyk-02].

Ve chvíli, kdy se nám podaří určit afinní transformaci, mezi okolími bodů, můžeme tato okolí popsat diferenciálními invarianty, momentovými invarianty dokonce můžeme použít i korelaci (zde nahrazenou metodou MEM-Pairs), tak že okolí na druhém obrázku příslušně transformujeme.

⁸ Zdroj obrázku: [Mikołajczyk-02]

4.3 RANSAC

Algoritmus RANSAC (RANdom SAmple Consensus) je algoritmus sloužící k robustnímu odhadu modelu. Byl představen Fischlerem a Bollesem [Fischler-81], kteří ho použili v oblasti automatické kartografie. RANSAC je robustní ve smyslu dobré odolnosti vzhledem k velkému množství chyb (tzv. *outlierů*) ve vstupních datech. Hartley a Zisserman [Hartley-04] upravili původní algoritmus pro řešení problému hledání korespondencí.

Běžné algoritmy (např. metoda nejmenších čtverců) se snaží odhadnout model na základě všech vstupních dat, takto získané řešení je znehodnoceno velkým množstvím chyb. Oproti tomu RANSAC odhaduje model z minimálního množství dat (2 body pro přímkou, 7 pro epipolární geometrii), které je pro odhad modelu potřebné, tato data jsou z původní množiny získána náhodným výběrem (náhodný vzorek – random sample).

Algoritmus RANSAC je velmi jednoduchý:

Mějme množinu vstupních dat D o velikosti M . Model je možno jednoznačně určit z N datových prvků, N tedy odpovídá počtu stupňů volnosti modelu ($N = 2$ pro přímkou, $N = 7$ pro epipolární geometrii).

1. Z množiny dat je vybrán náhodný vzorek o velikosti N .
2. Z vybraného vzorku jsou určeny parametry modelu \vec{x} (u epipolární geometrie je to fundamentální matice).
3. Zjistíme kolik prvků z D nevyhovuje modelu s parametry \vec{x} , s námi definovanou tolerancí, tento počet označíme o' .
4. Je-li o' dostatečně malé, přijmeme řešení a ukončíme výpočet.
5. Kroky 1-4 opakujeme L -krát.
6. Pokud jsme se dostali až sem výpočet selhal.

L je maximální povolený počet iterací a jeho překročení obvykle znamená, že data obsahují příliš velké množství chybných dat (chybně přiřazených korespondencí).

Používáme-li RANSAC k odhadu epipolární geometrie, pak k určení příslušnosti bodu k vybranému modelu (krok 3) používáme chybové vyjádření popsané v kapitole (3.1.5).

Otázkou je jaká chyba o' je dostatečně malá?

Abychom mohli zodpovědět tuto otázku musíme nejprve určit jaká je pravděpodobnost, že vybereme vzorek nezatížený chybami. Označíme množství chyb v datech jako o (outliery). Poměrné množství chybných dat ve vstupních datech, často označované jako *kontaminace* $c = o/M$. A poměrné množství správných dat (inlierů) $p = 1 - o/M$.

Pravděpodobnost, že vybereme prvek odpovídající modelu je tedy $p = 1 - c$ a pravděpodobnost, že takových prvků vybereme N je:

$$p_N = \left(1 - \frac{o}{M}\right) * \left(1 - \frac{o}{M-1}\right) * \dots * \left(1 - \frac{o}{M-N+1}\right), \quad (4.3.1)$$

protože platí, že $M \gg N$ můžeme psát:

$$p_N \approx \left(1 - \frac{o}{M}\right)^N = (1 - c)^N \quad (4.3.2)$$

Zopakujeme-li výběr L -krát, je pravděpodobnost, že alespoň jeden vybraný vzorek nebude kontaminovaný (nebude obsahovat chybná data) je:

$$p_N^L = 1 - \left(1 - (1 - c)^N\right)^L \quad (4.3.3)$$

Položíme-li podmínku, že hledané řešení musí mít, alespoň 95% pravděpodobnost, že je správné, tedy $p_N^L \geq 0.95$, můžeme z rovnice (4.3.3) získat nezbytný počet kroků v závislosti na kontaminaci:

$$L \leq \frac{\log(1 - 0.95)}{\log(1 - (1 - c)^N)}, \quad (4.3.4)$$

a nejhorší možnou kontaminaci v závislosti na počtu kroků

$$c \leq 1 - \sqrt[L]{1 - \sqrt[L]{(1 - 0.95)}}, \quad (4.3.5)$$

stále platí požadavek, že výsledek je nejméně na 95% správný.

dimenze N	Kontaminace dat v %							
	10	20	30	40	50	60	70	80
2	2	3	5	7	11	18	32	74
3	3	5	8	13	23	46	110	373
4	3	6	11	22	47	116	369	1871
5	4	8	17	38	95	292	1232	9361
6	5	11	25	64	191	731	4109	46808
7	5	13	35	106	382	1827	13697	234041
8	6	17	51	177	766	4570	45659	1170207

Tabulka 4.3.1: Závislost nezbytného počtu kroků algoritmu, při požadované spolehlivosti $p_N^L \geq 0.95$, na kontaminaci c a počtu stupňů volnosti N .

Nyní si představme, že se nacházíme v L' -tém kroku algoritmu. Zatím nejlepší odhad modelu do tohoto kroku dosáhl kontaminace c' (tedy nejnižší dosud dosažená kontaminace), c' je tedy horním odhadem c . Abychom zjistili, je-li námi dosažený výsledek dostatečně dobrý stačí dosadit L', c' do nerovnice (4.3.4) nebo (4.3.5). Dosa-díme-li do (4.3.3) zjistíme jak daleko jsme od požadovaného cíle, tedy jak pravděpodobné je, že jsme již dosáhli správného výsledku.

Tímto jsme zodpověděli na otázku jak poznáme, že máme výpočet ukončit.



Obrázek 4.3.1: Obrázek demonstrující odstranění outlierů metodou RANSAC, odhadující model epipolární geometrie.

Protože podstatou RANSACu je kombinatorické hledání je tento algoritmus velmi časově náročný. V kapitolách (4.3.1, 4.3.2) popíšeme metody, kterými se dá základní algoritmus výrazně urychlit.

4.3.1 R-RANSAC

Urychlení RANSACu popisované v této části je zaměřeno na třetí krok algoritmu, tak jak byl popsán v kapitole (4.3). Tedy zjištění kolik bodů ze vstupní datové množiny vyhovuje vybranému modelu. Vycházíme z jednoduché myšlenky: Místo abychom testovali celou množinu dat velikosti M , otestujeme pouze vzorek velikosti $d \ll M$ (pre-test). A teprve v případě, že tento test dává šanci na dobrý výsledek, provedeme test se zbývajících daty.

Tato myšlenka a její důsledky pro algoritmus RANSAC byla prezentována v [Chum-02] pod názvem R-RANSAC (Randomized-RANSAC), podobný postup je použit v [Lundberg-00] u algoritmu MEM-Pairs (kapitola (4.2.2)), při porovnávání bitsetů.

U takto upraveného základního algoritmu mohou nastat dvě chybové situace:

1. Pre-test akceptuje chybné řešení.
 - Nekritická chyba, důsledkem je pouze to, že je dokončen celý test s negativním výsledkem.
2. Je zamítnuto správné řešení.
 - Kritická chyba, pravděpodobnost nalezení lepšího řešení, než které máme klesá exponenciálně, ztráta takového řešení tedy není žádoucí.

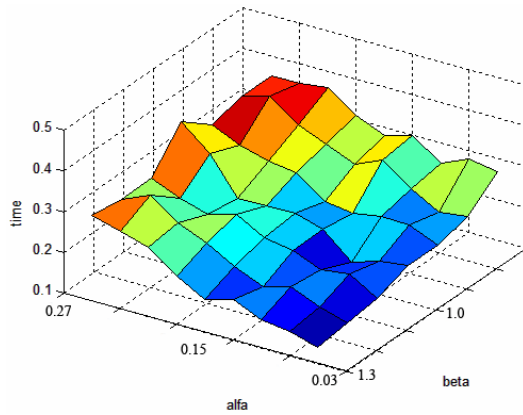
Otázkou je jak velké d máme zvolit, aby bylo dosaženo maximálního urychlení, při zachování stejných výsledků jakých dosahuje původní algoritmus? Odpověď na tuto otázku závisí na dvou hlavních faktorech: jak velký je vzorek d (v % vzhledem k M) a jakým způsobem je vyhodnocen počáteční test. V následujícím experimentu se pokusíme tuto otázku zodpovědět.

Experiment

Cílem tohoto experimentu bude zjistit závislost doby výpočtu na parametrech α , β a nalezení optimálního nastavení těchto parametrů, kde $\alpha = d / M$ je relativní velikost vzorku, $\beta c'$ je maximální povolená kontaminace v datovém vzorku, tedy $\beta = 1.1$ znamená, že pre-test akceptuje řešení, je-li chyba vzorku maximálně o 10% větší než chyba dosavadního nejlepšího řešení (c'). Vzhledem k charakteru algoritmu (založen na náhodném výběru) byl experiment 30x opakován a výsledky byly zprůměrovány.

β	α								
	0.03	0.06	0.09	0.12	0.15	0.18	0.21	0.24	0.27
0.7	0.27	0.28	0.25	0.30	0.28	0.31	0.36	0.35	0.35
0.8	0.21	0.20	0.24	0.29	0.26	0.29	0.32	0.37	0.35
0.9	0.19	0.22	0.22	0.21	0.22	0.27	0.28	0.30	0.33
1.0	0.19	0.18	0.16	0.22	0.24	0.24	0.24	0.33	0.39
1.1	0.16	0.20	0.19	0.22	0.23	0.27	0.25	0.27	0.30
1.2	0.15	0.17	0.20	0.18	0.21	0.25	0.28	0.33	0.30
1.3	0.14	0.16	0.17	0.20	0.18	0.22	0.27	0.29	0.31

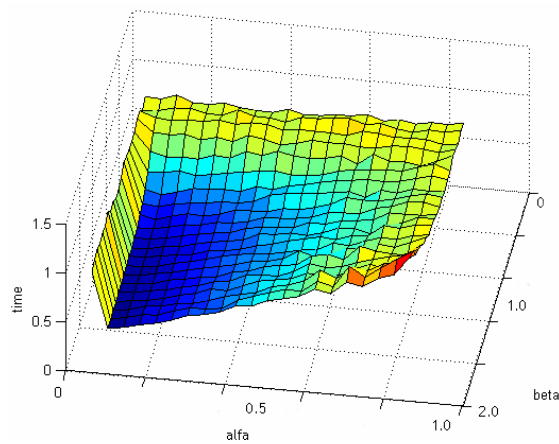
Tabulka 4.3.2: Experiment byl proveden s reálnými daty: 170 korespondencí z nich 40 chybných. Doba výpočtu originálního RANSACu byla v tomto případě průměrně 1 sekunda. Měříme závislost doby výpočtu na parametrech α a β .



Obrázek 4.3.2: Grafické znázornění výsledku experimentu, hodnot z tabulky (4.3.2).

Výsledkem tohoto experimentu je, že čím menší je hodnota α a větší β tím rychlejší je výpočet algoritmu. Podmínkou ovšem je aby datová množina obsahovala dostatečné množství bodů, je-li například $\alpha=0.01$, znamená to že pre-testu se účastní pouze 1% z datové množiny, obsahuje-li datová množina 100 bodů je to právě jeden bod a výsledkem je, že algoritmus se celkově zpomalí.

Důkaz tvrzení z předchozího odstavce můžeme pozorovat na obrázku (4.3.3), kde byl proveden stejný experiment pouze na větším rozsahu parametrů α , β .



Obrázek 4.3.3: Opakování experimentu s větším rozsahem hodnot vstupních parametrů.

4.3.2 PLUNDER

PLUNDER je zobecnění RANSACu pro odhad více modelů, byl prezentován v [Torr-98], jeho dalším vylepšením se dále zabývá [Chum-01].

Odhad epipolární geometrie, tak jak byla popsána v kapitole (3.1), je pro některé případy příliš obecný. Dochází-li například pouze k posunu kamery, vystačíme si s menším počtem stupňů volnosti a tím výrazně urychlíme hledání korespondencí (viz tabulka (4.3.1)). Ve většině případů však neznáme explicitně model, který by nejlépe vyhovoval vstupním datům, je proto potřeba vybrat nejlepší model, k tomu slouží právě PLUNDER.

Shrnutí modelů, přicházejících v úvahu pro dvojici pohledů je na následující straně v tabulce (4.3.3).

Algoritmus PLUNDER nejprve provede několik kroků RANSACu se všemi modely a poté vybere ten model, jehož výsledky jsou nejlepší a v odhadu tohoto modelu bude pokračovat. Velice důležitý je způsob jakým vybereme nejlepší model, resp. kolik kroků RANSACu pro který model učiníme, protože jednotlivé modely dosahují různé úspěšnosti za různou dobu v závislosti na počtu stupňů volnosti.

U počátečního testu všech modelů tedy zvolíme podmínku, aby výsledek byl správný minimálně na 5% (například), tedy aby po dosazení do rovnice (4.3.3) byla hodnota $p_N^L \geq 0.05$. Dosahuje-li několik modelů podobných výsledků, pokračujeme pouze s těmito modely nebo volíme ten s nejmenším počtem stupňů volnosti.

Podobně jako u R-RANSACu jsme zde popsali zejména hlavní princip, na kterém je toto urychlení založeno, a ne již konkrétní metodiku ohodnocení modelů, která byla použita v [Torr-98] nebo [Chum-01].

Dimenze 3 – transformace kamery

Fundamentální matice

7 stupňů volnosti, 7 korespondencí pro 1-3 řešení, 8 pro jedinečné řešení

$$x'Fx^T = 0, \text{ kde } F = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \quad (4.3.6)$$

Fundamentální matice pro afinní kameru

4 stupně volnosti, 4 korespondence

$$x'F_Ax^T = 0, \text{ kde } F_A = \begin{bmatrix} 0 & 0 & a_1 \\ 0 & 0 & a_2 \\ a_3 & a_4 & a_5 \end{bmatrix} \quad (4.3.7)$$

Fundamentální matice pro posunutí kamery

2 stupně volnosti, 2 korespondence

$$x'F_Tx^T = 0, \text{ kde } F_T = \begin{bmatrix} 0 & g_3 & -g_2 \\ -g_3 & 0 & g_1 \\ g_2 & -g_1 & 0 \end{bmatrix} \quad (4.3.8)$$

Dimenze 2 – transformace obrazu

Projektivní transformace, 8 stupňů volnosti, 4 korespondence

$$x' = xH, \text{ kde } H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (4.3.9)$$

Afinní transformace, 6 stupňů volnosti, 3 korespondence

$$x' = xK, \text{ kde } K = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ 0 & 0 & k_7 \end{bmatrix} \quad (4.3.10)$$

Posunutí obrazu, 2 stupně volnosti, 1 korespondence

$$x' = xL, \text{ kde } L = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3.11)$$

Tabulka 4.3.3⁹: Modely, které je možno použít pro dva pohledy U každého modelu je uveden počet stupňů volnosti a minimální možný počet korespondencí nutný k určení modelu.

⁹ Zdroj tabulky: [Torr-98]

4.4 Alternativní metody k algoritmu RANSAC

Algoritmus RANSAC není jediným řešením problému odhadu modelu s mnoha outliery v datech. K řešení tohoto problému se přímo nabízí použití genetických algoritmů. Proč dobré řešení zavrhnout jen pro to, že není nejlepší? Lepším přístupem je pokusit se takové řešení vylepšit drobnou změnou (mutace), nebo kombinací s jinými dobrými řešeními (křížení). Odhadem epipolární geometrie užitím genetických algoritmů se zabývá [Chai-98].

[Denton-02] přehledně shrnuje řadu algoritmů pro určení korespondencí a zároveň prezentuje vlastní algoritmus, tzv. *Local Search*, který v některých případech dosahuje lepších výsledků než RANSAC a který si jistě zaslouží hlubší zkoumání.

Kapitola 5

Projektivní rekonstrukce

Naším cílem je nalézt matice kamer a 3D souřadnice bodů z nekalibrovaných perspektivních snímků. Metody popsané v této kapitole dokáží využít libovolné množství snímků, čím je tento počet větší, tím více je potlačen šum ve vstupních datech. Vstupem projektivní rekonstrukce jsou bodové korespondence, tedy zobrazení m 3D bodů na n snímcích. Bodové korespondence můžeme zadávat ručně nebo získat automaticky tak, jak jsme popsali v kapitole (4). Protože nemáme žádné další informace o pozorované scéně je možná pouze projektivní rekonstrukce (rekonstrukce platí až na nějakou neznámou projektivní transformaci, viz kapitola (2.2.1)), úpravou projektivní rekonstrukce na metrickou se zabývá kapitola (6).

X_j jsou neznámé homogenní souřadnice 3D bodů, P_i jsou neznámé matice kamer o rozměrech 4×3 a x_{ij} jsou homogenní souřadnice 2D průmětů bodu X_j maticí P_i , kde $i = 1, \dots, n$ označuje jednotlivé snímky (kamery) a $j = 1, \dots, m$ jednotlivé body.

Projekce bodu X_j kamerou P_i platí až na nějaký skalární násobek λ_{ij} , tento násobek nazýváme *projektivní hloubka* a platí, že

$$\lambda_{ij} x_{ij}^T = P_i X_j^T. \quad (5.1)$$

Jsou-li souřadnice x_{ij} a X_j normalizované (poslední homogenní souřadnice je rovna jedné) a P_i jsou normalizované v tom smyslu, že norma posledního sloupce je rovna jedné, pak projektivní hloubky reprezentují skutečné optické hloubky, tj. kolmé vzdálenosti bodů od ohniskových rovin kamer.

Promítnutí všech bodů všemi kamerami můžeme maticově zapsat takto:

$$\begin{bmatrix} \lambda_{11} x_{11}^T & \lambda_{12} x_{12}^T & \cdots & \lambda_{1m} x_{1m}^T \\ \lambda_{21} x_{21}^T & \lambda_{22} x_{22}^T & \cdots & \lambda_{2m} x_{2m}^T \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n1} x_{n1}^T & \lambda_{n2} x_{n2}^T & \cdots & \lambda_{nm} x_{nm}^T \end{bmatrix}_{(m \times 3n)} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix}_{(4 \times 3n)} \begin{bmatrix} X_1^T & X_2^T & \cdots & X_m^T \end{bmatrix}_{(m \times 4)}, \quad (5.2)$$

$$W_{(m \times 3n)} = P_{(4 \times 3n)} X_{(m \times 4)}. \quad (5.3)$$

Projektivní rekonstrukci je možno rozdělit do dvou částí: Určení projektivních hloubek a následné získání matic kamer a souřadnic 3D bodů z matice W .

V kapitole (5.1) ukážeme dvě různé metody na výpočet projektivních hloubek, v (5.2) představíme metodu která získá z matice W faktorizací (rozkladem založeným na SVD (viz příloha (A))) matice kamer P a souřadnice promítaných bodů X .

Jak jsme již ukázali v kapitole (3.2) trifokální tensor, je možno při znalosti vztahu mezi snímky získat polohu i těch bodů, které nejsou na všech snímcích viditelné,

tedy matice W je neúplná – obsahuje prázdná místa. V kapitole (5.3) se zabýváme doplněním chybějících bodů v matici W založeným na informaci, že hodnota matice W je rovna čtyřem.

5.1 Výpočet projektivních hloubek

V této kapitole ukážeme dva algoritmy výpočtu projektivních hloubek. První je založený na epipolární geometrii a vyznačuje se konstantní složitostí. Druhý algoritmus je iterační a velmi rychle konverguje.

Abychom zajistili numerickou stabilitu, je před vlastním výpočtem je potřeba znormalizovat souřadnice bodů na každém snímku. A to tak, že jejich těžiště posuneme do nuly a změním měřítko aby průměrná vzdálenost od počátku byla $\sqrt{2}$, tuto úpravu jsme již popisovali v kapitole (3.1.4).

5.1.1 Výpočet projektivních hloubek založený na epipolární geometrii

Vycházíme z výrazu popisujícího vztah mezi projektivními hloubkami jednotlivých kamer:

$$(Fx^T_j)\lambda_j = (e \times q^T_i)\lambda_i, \quad (5.1.1)$$

kde F je fundamentální matice, popisující vztah mezi i -tou a j -tou kamerou, e je epipól a \times má význam vektorového násobení. Detailní odvození tohoto vztahu je možno nalézt v [Sturm-96, Martinec-00].

Řešením rovnice (5.1.1) ve smyslu nejmenších čtverců pro λ_i vzhledem k λ_j získáme vztah

$$\lambda_i = \frac{(e \times x_i) \cdot (Fx^T_j)}{\|e \times x_i\|^2} \quad (5.1.1)$$

Nejprve nastavíme projektivní hloubky první kamery na 1. Poté vypočteme epipolární geometrii (fundamentální matici a epipóly) mezi první a druhou kamerou a dosazením do rovnice (5.1.1) určíme projektivní hloubky druhé kamery. Tento postup opakujeme pro každou další kameru. Popsaný postup je rychlý, ale není příliš robustní, lepším přístupem je počítat projektivní hloubky mezi snímky, které jsou si nejbližší (scéna je zobrazovaná z nejbližší pozice). Například pro video-sekvenci snímků to budou vždy po sobě jdoucí snímky.

5.1.2 Iterační algoritmus výpočtu projektivních hloubek

Nejprve nastavíme projektivní hloubky (každý třetí řádek) matice W na jedna. Protože matice W by měla mít hodnotu 4, jak ukazují rovnice (5.2, 5.3) snížíme hodnotu matice metodou SVD (viz kapitola (A.3)) na 4. Použití metody SVD je nezbytné, nejde pouze o snížení hodnoty, jak posléze ukážeme u faktorizační metody (5.2).

Matici se sníženou hodnotou označíme \tilde{W} . Nové projektivní hloubky vypočteme z rovnice:

$$\lambda_{ij}^{(k+1)} = \lambda_{ij}^{(k)} \frac{\tilde{W}_{ij}^{(k)T} W_{ij}^{(k)}}{W_{ij}^{(k)T} W_{ij}^{(k)}} \quad (5.1.2)$$

Po výpočtu nových projektivních hloubek upravíme matici W takto: $W_i^{(k+1)} = x_{ij} \lambda_{ij}^{(k+1)}$, opět vypočteme \tilde{W} a postup opakujeme, dokud se matice W výrazněji mění.

5.2 Metoda faktorizace

Tato metoda, použitá v [Sturm-96], určí z matice W (obsahující vypočtené projektivní hloubky) kamery a 3D souřadnice bodů splňující podmínku danou rovnicemi (5.2, 5.3).

Nejprve provedeme rozklad matice W metodou SVD (viz příloha (A)), tedy

$$W = USV^T. \quad (5.2.1)$$

Z rovnic (5.2, 5.3) vyplývá, že matice W by měla mít hodnotu 4, přítomnost šumu však způsobuje, že tomu tak není. Hodnotu snížíme tak, že diagonální matici $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ upravíme na tvar $S = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \dots, 0)$. Po této úpravě platí

$$W_{(m \times 3n)} \approx \hat{W}_{(m \times 3n)} = U_{(3n \times 3n)} S_{(m \times 3n)} V_{(m \times m)}^T = U_{(4 \times 3n)} S_{(4 \times 4)} V_{(m \times 4)}^T. \quad (5.2.2)$$

Libovolný rozklad matice $S = S'S''$ nám umožní psát

$$\hat{W} = \underbrace{US'}_{\hat{U}} \underbrace{S''V^T}_{\hat{V}} = \hat{U}_{(4 \times 3n)} \hat{V}_{(m \times 4)}^T. \quad (5.2.3)$$

Maticí \hat{U} můžeme reprezentovat jako n (3×4) projekčních matic \hat{P}_i a \hat{V} jako seznam m 4-prvkových vektorů \hat{Q}_j (3D body)

$$\hat{W} = \hat{U}\hat{V} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} [Q_1 \quad Q_2 \quad \cdots \quad Q_m]. \quad (5.2.4)$$

Rozklad matice $S = S'S''$ je libovolný, většinou volíme $S' = S, S'' = I$ nebo $S' = S'' = \text{diag}(\sigma_1^{1/2}, \sigma_2^{1/2}, \sigma_3^{1/2}, \sigma_4^{1/2})$.

Podíváme-li se nyní na iterační algoritmus výpočtu projektivních hloubek (kapitola (5.1.2)) v kontextu této kapitoly. Můžeme vidět, že matice \tilde{W} vznikne vlastně opětým průmětem 3D bodů kamerami tak, jak by byly získány faktorizací, tedy opětným složením do matice \tilde{W} s hodnotou 4. Iterační algoritmus je tedy v podstatě založen na opakování faktorizační metody.

Vynásobení trojice řádků (kamera), nebo sloupce (bod) matice W nějakou nenulovou skalární hodnotou nemá na rekonstruované kamery a 3D body vliv. Před vlastní faktorizací je tedy dobré, z důvodu numerické stabilizace, přenásobit matici W tak aby všechny sloupce a řádky byly přibližně stejně významné. Toho můžeme jednoduše docílit následujícím iteračním postupem:

1. Každý sloupec l vynásobíme tak, aby $\sum_{r=1}^{3n} (w_{lr})^2 = 1$.
2. Každou trojici řádek ($3k-2, 3k-1, 3k$) vynásobíme tak, aby $\sum_{l=1}^m \sum_{i=3k-2}^{3k} (w_{li})^2$.
3. Dokud se W významně mění opakujeme kroky 1 a 2.

Tento postup je potřebný zejména počítáme-li projektivní hloubky z epipolární geometrie.

5.3 Doplnění chybějících bodů

Při větším počtu snímků není možno získat bodové korespondence na všech snímcích, bod viditelný na některých snímcích nemusí být viditelný na jiných. Tato situace způsobuje, že matice W obsahuje prázdná místa a není možno ji faktorizovat. Algoritmus, který zde popíšeme je založen na Jacobsově algoritmu [Jacobs-97] pracujícím s ortografickou kamerou a v [Martinec-00] upraven pro perspektivní kameru.

Jacobsův algoritmus dokáže doplnit libovolnou matici na požadovanou hodnotu. Nejprve ukážeme příklad jak je možno doplnit prázdná místa v matici nízké hodnoty.

Příklad 5.3.1¹⁰: Necht' má naše matice M hodnotu 1 a políčka v ní chybí na místech s otazníky:

$$M = \begin{bmatrix} 2 & 6 \\ ? & 3 \\ ? & 3 \end{bmatrix}$$

Protože má matice M hodnotu 1 (determinant všech podmatic velikosti 2×2 je roven nule), musí být jednotlivé sloupce matice lineárně závislé. V tomto případě musí být první sloupec násobkem druhého, jediným přípustným řešením je:

$$M = \begin{bmatrix} 2 & 6 \\ 1 & 3 \\ 1 & 3 \end{bmatrix}.$$

Podstatou Jacobsova algoritmu je hledání báze prostoru generující matici W . Tato báze má dimenzi 4 a generuje lineární projektivní vektorový prostor, který označíme L . Získáme-li tuto bázi, je již doplnění chybějících bodů triviální.

Výpočet prostoru L

Kdybychom znali čtveřici lineárně nezávislých sloupců, měli bychom rovnou bázi prostoru L . Jednotlivé sloupce W však nejsou kompletní i přesto obsahují užitečnou informaci.

Z matice W vybereme množinu čtveřic sloupců. L pak leží v lineárním obalu (dále LO) těchto čtveřic a tedy i v LO průniku těchto čtveřic. Zmenšíme-li tento průnik na 4, L bude určen přesně. LO čtveřice vektorů vytvoříme tak, že u každého vektoru nastavíme hodnotu chybějících bodů na 0 a zároveň ke každému vektoru přidáme tentýž vektor s hodnotou chybějících bodů různou od nuly (např. 1). Lineární obal LO se pak skládá z osmi vektorů a generuje vektory s libovolnou hodnotou na místě chybějících bodů. Aby bylo možno vybranou čtveřici akceptovat, je nutno aby mezi nimi bylo minimálně dimenze $d + 1$ (tedy 5) řádek bez chybějících bodů.

Kvůli šumu se průnik lineárních obalů brzy vyprázdní, řešením je tedy přechod k doplňku (komplementu). Průnik tedy můžeme nahradit doplňkem ke sjednocení doplňků LO .

¹⁰ Zdroj příkladu: [Martinec-00]

Nejprve tedy náhodně vybereme skupinu z čtveřic sloupců a vytvoříme jejich lineární obaly M_i . Doplněk určíme pomocí rozkladu SVD (viz příloha (A)) $M_i = USV^T$, doplněk M'_i získáme jako poslední 4 sloupce matice U .

Sjednocení doplňků vyjádříme jako matici $M' = [M'_1 M'_2 \dots M'_z]$ a její komplement opět metodou SVD $M' = USV^T$, doplněk průniku doplňků M je pak tvořen posledními 4-mi sloupci matice U .

M je pak bází prostoru L .

Příklad 5.3.2: Protože úprava matice hodnosti 4 by byla nepřehledná, předvedeme metodu na matici hodnosti 2, mějme tedy matici W o hodnosti 2:

$$W = \begin{bmatrix} 1 & 4 & 4 & 4 \\ 7 & 5 & 2 & 2 \\ 2 & ? & ? & ? \\ ? & 5 & 5 & 5 \end{bmatrix},$$

nejprve vybereme několik dvojic sloupců a vytvoříme jejich lineární obaly, použijeme-li dvojic sloupců 14, 23 a 24, pak jejich lineární obaly jsou:

$$M_{14} = \begin{bmatrix} 1 & 1 & 4 & 4 \\ 7 & 7 & 2 & 2 \\ 2 & 2 & 0 & 1 \\ 0 & 1 & 5 & 5 \end{bmatrix}, M_{23} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 5 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 5 & 5 & 5 & 5 \end{bmatrix}, M_{24} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 5 & 5 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 5 & 5 & 5 & 5 \end{bmatrix}.$$

Numericky získané doplňky těchto obalů s dimenzí 2 (užitím SVD) jsou:

$$M'_{14} = \begin{bmatrix} 0.03 & -0.79 \\ 0.27 & 0.08 \\ -0.96 & -0.02 \\ -0.04 & 0.60 \end{bmatrix}, M'_{23} = \begin{bmatrix} 0.05 & -0.78 \\ 0.01 & 0.00 \\ -1.00 & 0.00 \\ 0.07 & 0.62 \end{bmatrix}, M'_{24} = \begin{bmatrix} 0.05 & -0.78 \\ -0.01 & 0.00 \\ -1.00 & 0.00 \\ 0.06 & 0.63 \end{bmatrix}.$$

Průnik těchto doplňků lineárních obalů můžeme psát jako:

$$M' = [M'_{14} M'_{23} M'_{24}] = \begin{bmatrix} 0.03 & -0.79 & 0.05 & -0.78 & 0.05 & -0.78 \\ 0.27 & 0.08 & 0.01 & 0.00 & -0.01 & 0.00 \\ -0.96 & -0.02 & -1.00 & 0.00 & -1.00 & 0.00 \\ -0.04 & 0.60 & 0.07 & 0.62 & 0.06 & 0.63 \end{bmatrix}.$$

Nakonec lineární bázi prostoru L získáme jako doplněk M' (opět na základě SVD), tedy:

$$M = \begin{bmatrix} 0.17 & 0.59 \\ -0.95 & 0.31 \\ -0.07 & 0.08 \\ 0.26 & 0.74 \end{bmatrix}.$$

Doplnění matice W

Maticovou reprezentaci prostoru L označíme M (jak jsme ho právě vypočetli), W_i i -tý sloupec matice W , p indexy nechybějících bodů, W_i^p sloupec tvořený pouze známými souřadnicemi bodů sloupce W_i , podobně M^p je podmatice matice M tvořená pouze řádky nechybějících bodů. Pak sloupec doplněné matice \tilde{W} určíme jako:

$$\tilde{W}_i \approx M \left((M^p)^+ W_i^p \right), \quad (5.3.1)$$

kde $(M^p)^+$ je pseudoinverze (viz příloha (A.2)) k M^p .

Příklad 5.3.3: Použijeme-li rovnici (5.3.1) na matici W z příklady (5.3.2) a na bázi M spočtenou tamtéž, získáme novou matici, která má doplněné body a jen minimálně se liší od původní matice W :

$$\tilde{W} = \begin{bmatrix} 1.11 & 4.05 & 3.99 & 3.99 \\ 7.11 & 5.00 & 2.00 & 2.00 \\ 0.71 & 0.77 & 0.51 & 0.51 \\ 1.14 & 4.96 & 5.00 & 5.00 \end{bmatrix}.$$

Rozdíl mezi \tilde{W} a původní maticí W (nejvíce patrný na pozici (1,3)) je způsoben zejména tím, že výsledek je spočten numericky. V případě, že by původní matice nemohla mít hodnotu 4 (v tomto příkladu 2), způsobilo by to další chybu. Tato chyba je však srovnatelná s chybou faktorizační metody (5.2), která se již při faktorizaci neprojevuje, protože matice \tilde{W} má již požadovanou hodnotu. Výsledek je také možno zlepšit zavedením většího počtu čtveřic (v tomto příkladu dvojic) sloupců.

Algoritmus doplnění chybějících bodů získaných projektivní projekcí

Vlastní algoritmus je možno rozdělit do dvou kroků:

1. Nejprve je nutno vypočítat projektivní hloubky u bodů, jejichž souřadnice jsou známy. K výpočtu projektivních hloubek není možno použít iterační algoritmus, protože matice W není úplná.
2. Jacobsův algoritmus pak doplňuje chybějící body včetně měřítek.

Tento proces můžeme opakovat dokud nezkonverguje, tj. již se nedoplňují žádné další body.

Další informace týkající se problematiky doplňování chybějících bodů je možno nalézt v práci [Martinec-00]. Popsaný postup je čistě algebraické doplnění na základě informace o hodnotě, tento problém je však možno řešit i jinak: dokážeme-li spočítat vztah mezi trojicí kamer (epipolární geometrie, trifokální tensor), můžeme body doplňovat na jejich základě.

Kapitola 6

Metrická rekonstrukce

V kapitole (5) jsme již naznačili, že výsledky projektivní rekonstrukce jsou správné až na nějakou lineární projektivní transformaci H . Tato transformace, také nazývaná *Projective Distortion Matrix* (PDM), ovlivňuje kamery a body získané projektivní rekonstrukcí takto:

$$W = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX, \quad (6.1)$$

kde $P = \hat{P}H$ a $X = H^{-1}\hat{X}$. Jak vidíme rovnice (6.1) platí pro libovolnou regulární matici H , naším cílem je nalézt takovou H , která transformuje kamery a body do metrického prostoru.

V obecném případě není možno transformaci H určit a končíme projektivní rekonstrukcí, která rozhodně není uspokojivým výsledkem¹¹. Naším úkolem je tedy nalézt nějaké vnitřní omezení, tj. informace o pozorované scéně nebo o kamerách, které nám umožní upravit projektivní rekonstrukci na metrickou.

Jednou z možností je zadat přesné homogenní souřadnice pěti bodů ze kterých je možno H vypočítat. Naším hlavním předpokladem je však, že nemáme žádné informace o pozorované scéně, zaměříme se tedy na omezení týkající se kamer.

V kapitole (6.1) nejprve ukážeme spojitost mezi PDM a absolutní duální kvadrikou, v (6.2) určíme kolik snímků je potřeba při určitých omezeních (z hlediska kamer). A konečně v části (6.3) budeme prezentovat 3 varianty algoritmu, který převádí projektivní rekonstrukci na metrickou na základě různých konfigurací kamery.

6.1 Metrická rekonstrukce a absolutní kvadrika

Většina auto-kalibračních metod počítá vnitřní parametry kamer ze vztahu

$$\omega^* = KK^T, \quad (6.1.1)$$

který jsme odvodili v kapitole (2.4). Je-li $\omega^*_{(3 \times 3)}$ známa, můžeme vnitřní parametry kamery snadno získat Choleského dekompozicí, nebo SVD (viz příloha (A.4)). Podle definice je duální absolutní kuželosečka ω^* projekcí absolutní kvadriky Ω , tak že

$$\omega^* = P\Omega P^T. \quad (6.1.2)$$

Nyní ukážeme, že určení PDM je ekvivalentní výpočtu absolutní kvadriky (viz [Sainz-02, Hartley-04]).

¹¹ Jak již víme z kapitoly (2) projektivní geometrie, promítnutím kružnice v projektivním prostoru je obecně kuželosečka – výsledek projektivní rekonstrukce tedy většinou neodpovídá realitě, k tomu je potřeba rekonstrukce metrická.

Vyjádřeme PDM ve tvaru

$$H = \begin{bmatrix} H & b_1^T \\ h_1 & 1 \end{bmatrix} \quad (6.1.3)$$

Bod odpovídající počátku souřadnic metrického prostoru vypočítáme jako $H(0,0,0,1)^T$, což je $(b_1,1)^T$, pak b_1 je souřadnice odpovídající počátku. Bez újmy na obecnosti, můžeme položit $b_1 = (0,0,0)$ (projektivní a metrický prostor sdílejí společný prostor).

Každá matice P_i , o rozměrech (3×4) , získaná projektivní rekonstrukcí může být rozložena na

$$P_i H = \mu_i K_i [R_i | T_i], \quad i = 1, \dots, m, \quad (6.1.4)$$

kde jednotlivé parametry tohoto rozkladu (matice kamery) byly popsány v kapitole (2.4.2).

Rozdělíme-li matici kamery na $P_i = [\tilde{P}_i, p_i]$, kde \tilde{P}_i jsou první tři sloupce, p_i čtvrtý, pak výraz (6.1.4) můžeme zapsat dvojicí rovnic:

$$[\tilde{P}_i, p_i] \begin{bmatrix} H_1 \\ h_1 \end{bmatrix} = \mu_i K_i R_i, \quad (6.1.5)$$

$$[\tilde{P}_i, p_i] \begin{bmatrix} 0^{T(1 \times 3)} \\ 1 \end{bmatrix} = \mu_i K_i T_i. \quad (6.1.6)$$

Dosadíme-li do vztahu (6.1.1) vztah (6.1.5), můžeme psát

$$\omega^* = K K^T = K R R^T K^T = \frac{1}{\mu^2} P_i \begin{bmatrix} H_1 \\ h_1 \end{bmatrix} [H_1^T, h_1^T] P_i^T = P_i \Omega^* P_i^T, \quad (6.1.7)$$

čímž jsme našli vztah mezi PDM a duální absolutní kvadrikou Ω^* .

6.2 Kolik snímků je potřeba?

Omezení týkající se vnitřních parametrů kamer můžeme rozdělit do dvou hlavních skupin:

1. Některé parametry kamer jsou konstantní, tj. pro získání jednotlivých snímků bylo použito kamer se shodnými parametry (případně jedné kamery).
2. Některé parametry kamer jsou známé (právě tento případ využijeme v normalizační metodě, kapitola (6.3)).

Pravděpodobně nejvýhodnějšími omezeními je předpokládat parametr kamer $\beta = 0$, tedy pixely jsou čtvercové, což platí u většiny běžných kamer a fotoaparátů. Další „výhodné“ omezení je umístit principální bod do počátku souřadného systému, opět běžné kamery mají tento bod umístěn ve středu obrázku (principální bod se nám většinou posouvá v případě, že děláme výřez z vyfotografovaného obrázku). Používáme-li k získávání snímků stejný fotoaparát, můžeme tohoto faktu využít a považovat některé jeho parametry za konstantní.

V následující tabulce (6.2.1) si shrneme většinu běžných omezení, které můžeme při metrické rekonstrukci využít a minimální počet obrázků, který je s daným omezením k metrické rekonstrukci potřebný.

	$f =$	$k =$	$m =$	omezení
konstantní K	5	0	3	$\omega_{ij} / \omega_{33} = \omega_{ij}^1 / \omega_{33}^1$
známý principální bod $(u_0, v_0)^T$	0	2	4	$\omega_{13} = \omega_{23} = 0$
konstantní α a β	2	0	5	$\omega_{11} = \alpha \omega_{22} = 0$
$\beta = 0$	0	1	8	$\omega_{12} \omega_{33} = \omega_{13} / \omega_{23}$
známý $(u_0, v_0)^T$ a $\beta = 0$	0	3	3	$\omega_{12} = 0$ $\omega_{13} = \omega_{23} = 0$

Tabulka 6.2.1: Nezbytný počet snímků (kamer) nutný k metrické rekonstrukci, při různých omezeních parametrů kamery. f je počet neměnných parametrů, k je počet známých parametrů a m je počet snímků (kamer). Jednotlivé vnitřní parametry kamer jsou popsány v kapitole (4.2.2).

Vztah mezi počtem snímků m , počtem neměnných parametrů f a počtem známých parametrů k můžeme vyjádřit jako

$$mk + (m - 1)f \geq 8, \quad (6.2.1)$$

důkaz je možno nalézt v [Hartley-04].

6.3 Normalizační metoda

V této kapitole popíšeme metodu převodu projektivní rekonstrukce na metrickou, tento proces nazýváme *normalizace* a byl publikován v [Han-00], ze které je také čerpán obsah této kapitoly. Vstupem algoritmu jsou projektivní matice kamer a souřadnice bodů získané projektivní rekonstrukci, kapitola (5).

Základní podmínkou normalizační metody je nulové zkosení $\beta = 0$ (pixely jsou čtvercové). Algoritmus je prezentován ve třech verzích: kapitoly (6.3.1, 6.3.2, 6.3.3), podle toho, jaká další omezení klademe na parametry kamer:

Případ 1: Neznámé jsou pouze ohniskové vzdálenosti.

Případ 2: Ohniskové vzdálenosti a principální bod jsou neznámé a pozice principálních bodů je konstantní.

Případ 3: Ohniskové vzdálenosti, principální body i poměr stran pixelu jsou proměnné a neznámé.

První lineární algoritmus pracuje v situacích, kdy je použita jedna kamera, a jediné mění se pouze ohnisková vzdálenost (zoom). Druhý lineární algoritmus je vhodný pro situace, kde ohnisková vzdálenost je měněna minimálně a principální bod je velmi blízko tomu, aby byl konstantní (například sekvence leteckých snímků). Třetí algoritmus je bilineární a může být použit v případě, že jsou snímky pořízeny více kamer a snímek po snímku se mění ohniskové vzdálenosti, principální body a poměry stran pixelů.

Protože je větší část algoritmu společná pro všechny části, bude v kapitolách (6.3.1, 6.3.2, 6.3.3) uvedena pouze ta část výpočtu ve kterých se liší.

Předpokládáme projekční matice ve tvaru

$$P_i = K_i [R_i | T_i], \quad (6.3.1)$$

kde

$$K_i = \begin{bmatrix} f_i & 0 & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{bmatrix}, \quad R_i = \begin{bmatrix} i_i \\ j_i \\ k_i \end{bmatrix}, \quad T_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}.$$

Spojíme-li rovnici (6.3.1) pro všechna $i = 1, \dots, n$, dostaneme

$$P = [M | T], \quad (6.3.2)$$

kde

$$M = [m_{x1}, m_{y1}, m_{z1}, \dots, m_{xn}, m_{yn}, m_{zn}]^T$$

$$T = [T_{x1}, T_{y1}, T_{z1}, \dots, T_{xn}, T_{yn}, T_{zn}]^T,$$

a

$$\begin{aligned}
m_{xi} &= \mu_i f_i i_i + \mu_i u_{0i} k_i \\
m_{yi} &= \mu_i \alpha_i f_i j_i + \mu_i v_{0i} k_i \\
m_{zi} &= \mu_i k_i \\
T_{xi} &= \mu_i f_i t_{xi} + \mu_i u_{0i} t_{zi} \\
T_{yi} &= \mu_i \alpha_i f_i t_{xi} + \mu_i v_{0i} t_{zi} \\
T_{zi} &= \mu_i t_{zi}
\end{aligned} \tag{6.3.3}$$

matici reprezentující 3D body vyjádříme jako

$$X = \begin{bmatrix} S \\ 1 \end{bmatrix}, \tag{6.3.4}$$

kde $S = [s_1, \dots, s_m]$ a $s_j = (x_j, y_j, z_j)^T$, $x_j = (w_j s_j^T, w_j)^T$.

Počátek souřadného systému položíme do těžiště bodů

$$\sum_{j=1}^m w_j s_j = 0, \tag{6.3.5}$$

a dostaneme

$$\sum_{j=1}^m \lambda_{ij} u_{ij} = \sum_{j=1}^m (m_{xi} w_j s_j + w_j T_{xi}) = m_{xi} \sum_{j=1}^m w_j s_j + T_{xi} \sum_{j=1}^m w_j = T_{xi} \sum_{j=1}^m w_j, \tag{6.3.6}$$

obdobně

$$\sum_{j=1}^m \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^m w_j, \quad \sum_{j=1}^m \lambda_{ij} = T_{zi} \sum_{j=1}^m w_j. \tag{6.3.7}$$

$(x_j, y_j, z_j, w_j)^T$ jsou homogenní souřadnice 3D bodů, $(u_{ij}, v_{ij}, \lambda_{ij})^T$ jsou 2D průměry těchto bodů, opět v homogenních souřadnicích.

Definujeme-li 4×4 projektivní transformaci H jako

$$H = [A_{(4 \times 3)} | B_{(4 \times 1)}], \tag{6.3.8}$$

pak vztah $P = \hat{P}H$, můžeme rozepsat jako

$$[M|T] = \hat{P}[A|B] \tag{6.3.9}$$

a dostáváme

$$T_{xi} = \hat{P}_{xi} B, \quad T_{yi} = \hat{P}_{yi} B, \quad T_{zi} = \hat{P}_{zi} B. \tag{6.3.10}$$

Z rovnic (6.3.6) a (6.3.7) víme, že

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^m \lambda_{ij} u_{ij}}{\sum_{j=1}^m \lambda_{ij}}, \quad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^m \lambda_{ij} v_{ij}}{\sum_{j=1}^m \lambda_{ij}}. \quad (6.3.11)$$

Z rovnic (6.3.10) a (6.3.11) sestavíme soustavu $2n$ lineárních rovnic o 4 neznámých prvcích matice B . Soustavu vypočteme metodou nejmenších čtverců (použijeme SVD (viz příloha (A.1))).

Protože m_{xi} , m_{yi} a m_{zi} jsou součty rotačních vektorů (pouze se změněným měřítkem), můžeme z rovnic (6.3.2) formulovat následující omezení:

$$\begin{aligned} |m_{xi}|^2 &= \mu_i^2 f_i^2 + \mu_i^2 u_{0i}^2 \\ |m_{yi}|^2 &= \mu_i^2 \alpha_i^2 f_i^2 + \mu_i^2 v_{0i}^2 \\ |m_{zi}|^2 &= \mu_i^2 \\ m_{xi} m_{xi} &= \mu_i^2 u_{0i} v_{0i} \\ m_{xi} m_{zi} &= \mu_i^2 u_{0i} \\ m_{yi} m_{zi} &= \mu_i^2 v_{0i} \end{aligned} \quad (6.3.12)$$

Na základě tří rozdílných konfigurací vnitřních parametrů kamer, převedeme tato omezení na MM^T (jednotlivé případy jsou popsány v kapitolách 6.3.1, 6.3.2 a 6.3.3). Protože platí

$$MM^T = \hat{P}AA^T\hat{P}^T, \quad (6.3.13)$$

můžeme metodou nejmenších čtverců vyřešit tuto soustavu pro 10 neznámých prvků symetrické matice $Q = AA^T$ o rozměrech 4×4 . Tento rozklad provedeme Choleského dekompozicí nebo SVD (viz příloha (A.4)).

Je-li A nalezena, můžeme sestavit projektivní transformaci $H = [A|B]$ a vypočítat metrické matice kamer $P = \hat{P}H$ a 3D souřadnice bodů $X = H^{-1}\hat{X}$. Zároveň můžeme vypočítat vnitřní a vnější parametry kamer, viz tabulka (6.4.1).

Vnitřní parametry	
měřítka:	$\mu_i = m_{zi} \quad (6.3.14)$
principální bod:	$u_{0i} = \frac{m_{xi}m_{zi}}{\mu_i^2}, v_{0i} = \frac{m_{yi}m_{zi}}{\mu_i^2} \quad (6.3.15)$
ohnisková vzdálenost	$f_i = \frac{\sqrt{ m_{xi} ^2 - \mu_i^2 u_{0i}^2}}{\mu_i} \quad (6.3.16)$
poměr výšky a šířky pixelu	$\alpha_i = \frac{\sqrt{ m_{yi} ^2 - \mu_i^2 v_{0i}^2}}{\mu_i f_i} \quad (6.3.17)$
Vnější parametry	
$ \begin{aligned} k_i &= \frac{m_{zi}}{\mu_i}, & i_i &= \frac{m_{xi} - \mu_i^2 u_{0i}^2 k_i}{\mu_i f_i}, & j_i &= \frac{m_{yi} - \mu_i^2 v_{0i}^2 k_i}{\mu_i \alpha_i f_i} \\ t_{zi} &= \frac{T_{zi}}{\mu_i}, & t_{xi} &= \frac{T_{xi} - \mu_i u_{0i} t_{zi}}{\mu_i f_i}, & t_{yi} &= \frac{T_{yi} - \mu_i v_{0i} t_{zi}}{\mu_i \alpha_i f_i} \end{aligned} \quad (6.3.18) $	

Tabulka 6.3.1: Určení vnitřních a vnějších parametrů kamer

Pro všechny případy, které nyní popíšeme, stále platí podmínka $\beta = 0$.

6.3.1 Příklad 1

Předpokládáme, že ohniskové vzdálenosti jsou jediný neznámý vnitřní parametr, pak

$$u_{0i} = v_{0i} = 0, \alpha_i = 1 \quad (6.3.19)$$

Dosazením do rovnic (6.3.12), získáme tato omezení matice Q :

$$\begin{aligned}
|m_{xi}|^2 &= |m_{yi}|^2 \\
m_{xi}m_{yi} &= m_{xi}m_{zi} = m_{yi}m_{zi} = 0
\end{aligned} \quad (6.3.20)$$

Přidáme ještě jednu rovnici, kterou nastavíme měřítko první kamery na 1, tedy $\mu_1 = 1$:

$$|m_{z1}|^2 = 1 \quad (6.3.21)$$

Celkem tedy máme $4n + 1$ lineárních rovnic o 10-ti neznámých prvcích matice Q , kterou snadno vyřešíme.

6.3.2 Příklad 2

Předpokládáme, že ohniskové vzdálenosti jsou neznámé a principální bod je konstantní, pak

$$u_{0i} = u_0, v_{0i} = v_0, \alpha_i = 1 \quad (6.3.22)$$

Dosazením do rovnic (6.3.12), získáme tato omezení matice Q :

$$\begin{aligned} \frac{m_{xi}m_{yi}}{m_{xi}m_{zi}} &= \frac{m_{yi}m_{zi}}{m_{zi}m_{zi}} \\ \left(|m_{xi}|^2 - |m_{yi}|^2 \right) (m_{zi}m_{zi}) &= (m_{zi}m_{zi})^2 - (m_{yi}m_{zi})^2 \end{aligned} \quad (6.3.23)$$

a

$$\begin{aligned} \frac{m_{zi}m_{zi}}{|m_{xi}|^2 - |m_{yi}|^2} &= \frac{m_{zj}m_{zj}}{|m_{xj}|^2 - |m_{yj}|^2}, \quad \frac{|m_{xi}|^2 - |m_{yi}|^2}{m_{xi}m_{yi}} = \frac{|m_{xj}|^2 - |m_{yj}|^2}{m_{xj}m_{yj}} \\ \frac{m_{xi}m_{yi}}{m_{xi}m_{zi}} &= \frac{m_{xj}m_{yj}}{m_{xj}m_{zj}}, \quad \frac{m_{xi}m_{zi}}{m_{yi}m_{zi}} = \frac{m_{xj}m_{zj}}{m_{yj}m_{zj}}, \quad \frac{m_{yi}m_{zi}}{m_{zi}m_{zi}} = \frac{m_{yj}m_{zj}}{m_{zj}m_{zj}}, \end{aligned} \quad (6.3.24)$$

kde $j = i + 1$, je-li $i \neq n$, $j = 1$ je-li $i = n$. Opět přidáme rovnici, kterou nastavíme měřítko první kamery na 1, tedy $\mu_1 = 1$:

$$|m_{z1}|^4 = 1 \quad (6.3.25)$$

Toto jsou o lineární rovnice obsahující neznámé prvky matice $Q_1 = qq^T$, kde q je vektor o rozměrech 10×1 sestavený z 10-ti neznámých prvků matice Q . Získáme tedy $7n + 1$ lineárních rovnic o 55-ti neznámých prvcích matice Q_1 .

Poté co vypočteme Q_1 , q získáme tak, že snížíme hodnotu matice Q_1 na 1 (viz příloha (A.3)). 10 prvků q poté uspořádáme do matice Q , jejíž rozkladem můžeme získat A .

6.3.3 Příklad 3

Předpokládáme, že ohniskové vzdálenosti, principální body a poměr stran pixelů (dále jen poměr stran) jsou neznámé a proměnlivé, omezení daná rovnicemi (6.3.12) použijeme jako bilineární rovnice pro ohniskové vzdálenosti a principální body plus poměr stran. Začneme s hrubým odhadem principálního bodu a poměru stran u první kamery (α_1) a vložíme je do lineárních omezení prvků matice Q :

$$\begin{aligned} m_{xi}m_{yi} &= u_{0i}v_{0i}m_{zi}m_{zi} \\ m_{xi}m_{zi} &= u_{0i}m_{zi}m_{zi} \\ m_{yi}m_{zi} &= v_{0i}m_{zi}m_{zi} \end{aligned} \quad (6.3.26)$$

Dále přidáme další dvě rovnice pro $\mu_1 = 1$:

$$\begin{aligned} \alpha_1^2(m_{x1}^2 - u_{01}^2) &= |m_{y1}|^2 - v_{01}^2\alpha_1^2(m_{x1}^2 - u_{01}^2) = |m_{y1}|^2 - v_{01}^2 \\ |m_{z1}|^2 &= 1 \end{aligned} \quad (6.3.27)$$

Poté co získáme matici H , vypočteme kamery a 3D body ($P = \hat{P}H$ a $X = H^{-1}\hat{X}$), vypočteme z rovnic (6.3.15, 6.3.16, 6.3.17) vnitřní parametry kamer.

Vezmeme nově vypočtené principální body a poměr stran první kamery a opět je použijeme k výpočtu matice H , postup opakujeme, dokud vnitřní parametry kamer nekonvergují.

6.3.4 Shrnutí normalizační metody

1. Nejprve provedeme projektivní rekonstrukci, matici W dekomponujeme na \hat{P} a \hat{Q} .
2. Sečteme řádky W a vypočteme poměry mezi nimi podle rovnice (6.3.11).
3. Sestavíme $2n$ lineárních rovnic o 4 neznámých prvků matice B založených na poměrech z kroku 2 a vypočteme B .
4. Sestavíme soustavu lineárních rovnic o 10-ti neznámých prvcích symetrické matice Q a vypočteme Q .
5. Rozložíme Q , abychom získali A : $Q = AA^T$.
6. Z matic A a B vytvoříme projektivní transformaci $H = [A|B]$.
7. Vypočteme souřadnice bodů $Q = H^{-1}\hat{Q}$ a kamer $P = \hat{P}H$.
8. Z rovnic uspořádaných v tabulce (6.3.1) vypočteme vnitřní parametry kamer.

Kapitola 7

Shrnutí algoritmu 3D rekonstrukce

V této kapitole shrneme celý algoritmus 3D rekonstrukce, tak jak by měl vypadat aby ho bylo možno prakticky použít. Tento postup je založen na teorii popsané v předchozích kapitolách. Zároveň ukazuje věci, jejichž podrobné vysvětlení se do této práce nevešlo a možnosti dalšího vývoje v oblasti 3D rekonstrukce.

7.1 Detekce bodů

Jak jsme již ukázali v kapitole (4) detekce bodů je prvním a základním krokem celého procesu, není však jedinou možností nezanedbatelné je použití úseček, nebo jiných složitějších objektů. Body a úsečky mají však velkou výhodu v jednoduchosti matematického popisu a ideálním stavem je rekonstrukce provedená kombinací bodů a úseček. Hlavní požadavek na detekované body (úsečky) je aby byly zastoupeny na obou obrázcích a aby se vyznačovaly nějakou výraznou vlastností, která by umožňovala snadné nalezení korespondencí.

7.2 Hledání korespondencí

Několik základních metod použitých k hledání korespondencí jsme popsali v kapitole (4). Metody je možno rozdělit podle toho jakým způsobem vyhodnocujeme jejich vlastnosti, jednou z možností je popsat detekované elementy na základě vlastností snímku v jejich lokálním okolí, druhý přístup vychází z globálního hlediska a hledá geometrický vztah mezi dvěma množinami bodů. Oba tyto přístupy často kombinujeme.

Podáří-li se nám odhadnout model epipolární geometrií nebo trifokálními tenzory (kapitola (3)), můžeme vyloučit body, které tomuto modelu nevyhovují, zároveň můžeme hledat nové korespondence založené na omezení dané odhadnutým modelem. Použijeme-li odhad více modelů (PLUNDER, kapitola (4.3.2)) nejen že výrazně urychlíme výpočet, ale zároveň můžeme detekovat degenerované scény, tj. scény, které vyhovují více různým modelům (tato situace vzniká například pohybuje li se ve scéně nějaký objekt, nebo pokud se vyskytuje opakující se vzor).

Zejména při větším počtu snímků (ale i u trojice) nastává situace, že některé body jejichž korespondence jsme získali nejsou viditelné, nebo nebyly detekovány na dalších snímcích. Tento problém, tedy doplnění souřadnic průmětů chybějících bodů na některých snímcích, můžeme řešit dvěma způsoby. První způsob je doplnění bodů na základě dvojice epipolárních geometrií, nebo trifokálního tensoru, známe-li průmět bodu na dvou snímcích a vztah mezi třemi snímky, můžeme jednoduše určit průmět na třetím snímku. Tento postup je velmi náchylný na kvalitu odhadu epipolární geometrie (trifokálního tensoru). Druhou možností, kterou jsem popsali v kapitole (5.3) je uspořádat body do matice a tuto matici doplnit na základ znalosti její hodnoty (v našem případě je hodnota rovna 4). Kombinace těchto způsobů by měla dosahovat lepších výsledků než jednotlivé metody, zejména v případě, že doplnění bodů na základě hodnoty nemusí doplnit všechny body.

7.3 Projektivní rekonstrukce

Projektivní rekonstrukce faktorizační metodou dosahuje při kvalitním vstupu velmi dobrých výsledků. Její největší slabinou je výpočet projektivních hloubek, zejména používáme-li algoritmus založený na epipolární geometrii (jako obvykle je možno použít trifokální, nebo multifokální tensor) vyžaduje, aby výpočet projektivních hloubek probíhal mezi dvojicemi snímků, které jsou velmi podobné (tento problém je řešen v části (7.7)).

projektivní rekonstrukce se snaží vyhovět všem zadaným bodům. I přes to můžeme na základě výsledku projektivní rekonstrukce odhalit chybně určené korespondence, je-li korespondencí dostatečné množství a chyba není příliš mnoho.

7.4 Metrická rekonstrukce

Podobně jako projektivní rekonstrukce i metrická dosahuje velmi dobrých výsledků jsou-li dobře a ve velkém množství určeny korespondence. Právě u metrické rekonstrukce můžeme výsledky vylepšit, známe-li parametry pozorované scény, nebo kalibrační matice kamer. Metrická rekonstrukce je možná i když nemáme žádné informace s výjimkou samotných snímků a předpokladu nulového zkosení pixelů (tuto podmínku běžné kamery a fotoaparáty splňují). Jak je ukázáno v kapitole (6.2) k úspěšné rekonstrukci bez dalších informací je potřeba minimálně 8 snímků.

Ve chvíli, kdy máme scénu (resp. množinu bodů) a kamery zrekonstruovány, můžeme přistoupit k dalšímu získávání bodů a vylepšení dosažených výsledků. Tato vylepšení jsou shrnuta v kapitole (7.5).

7.5 Vylepšení modelu na základě známých kamer a polohy některých bodů

Metrickou rekonstrukcí celý proces nekončí, ve chvíli kdy známe matice kamer můžeme přistoupit k takzvané triangularizaci¹², což je zpětné promítnutí bodů kamerou do scény. Tento postup je 3D alternativou k získání nových korespondencí založené na epipolární geometrii, tedy body jsou mapovány na přímku. Na základě takto získaných korespondencí můžeme opět přistoupit k projektivní a metrické rekonstrukci a opakováním takového postupu vylepšovat výsledek.

Další možností je vybrat ty body v prostoru které jsou blízko a spojit je do trojúhelníku. Nové body hledat uvnitř průmětu tohoto trojúhelníku jednotlivými kamerami. Ve většině případů bude rekonstruovaný bod ležet v blízkosti původních bodů. Opakováním tohoto postupu se zpočátku velmi hrubý model začne zjemňovat.

¹² Pod pojmem triangularizace je obvykle myšleno vytvoření trojúhelníkové sítě z množiny bodů, v tomto případě jde „trojici“ kamera-kamer-bod.

7.6 Rekonstrukce povrchu a textur

Rekonstrukce povrchu

Postupem popsaným v kapitole (7.5) získáme velké množství bodů v prostoru, dalším krokem je z těchto bodů vytvořit povrch. Jednou z možností je vytvořit z bodů trojúhelníkovou síť nezávisle na originálních snímcích, tím se ovšem ochuzujeme o cennou informaci. Lepším přístupem je kombinovat trojúhelníkovou síť získanou ve 3D s trojúhelníkovými sítěmi získanými ve 2D průmětech. Ideálním případem je, když trojúhelníky mezi zrekonstruovanými body na originálních snímcích jsou vyplněny konstantní barvou, pokud tomu tak není pokoušíme se je dále dělit tak jak jsme to ukázali v kapitole (7.5).

Rekonstrukce textur

Na první pohled může být otexturování povrchu triviální, postačuje například zprůměrovat barvy jednotlivých zobrazení jednotlivých bodů, nebo použít barvu ze snímku na kterém je konkrétní část povrchu nejvíce rovnoběžná s rovinou kamery. Nesmíme však zapomínat, že stejný bod má na různých snímcích jinou barvu i za normálních podmínek. V případě že jsou snímky pořízeny v rozdílných časech a při rozdílném osvětlení je rozdíl ještě výraznější. V obecném případě je tedy problém věrnosti barev zrekonstruovaného modelu jen velmi obtížně řešitelný, jsou-li však snímky získány při stejném osvětlení jsou výsledky uspokojivé.

7.7 Různé druhy vstupu

Rozptýlené snímky

Vstupem rekonstrukčního postupu může být v podstatě libovolná množina snímků, narážíme však na problém, jak poznat který snímek se kterým sousedí. Pro potřeby rekonstrukce potřebujeme rozdělit množinu snímků na dvojice nebo trojice, tvořené velmi podobnými snímky.

Můžeme se pokusit nalézt společné body na kombinacích snímků každý s každým, náročnost takového postupu je však příliš velká. Řešením je nějakým způsobem nalézt snímky, které mají k sobě nejbližší. Snímek můžeme například popsat sadou momentových nebo diferenciálních invariantů, tento postup však nemá dobré výsledky, zejména vlivem okolí pozorovaného objektu. Velmi dobré řešení tohoto problému nám nabízí publikace [Schaffalitzky-02], ze které jsou obrázky 7.7.1.

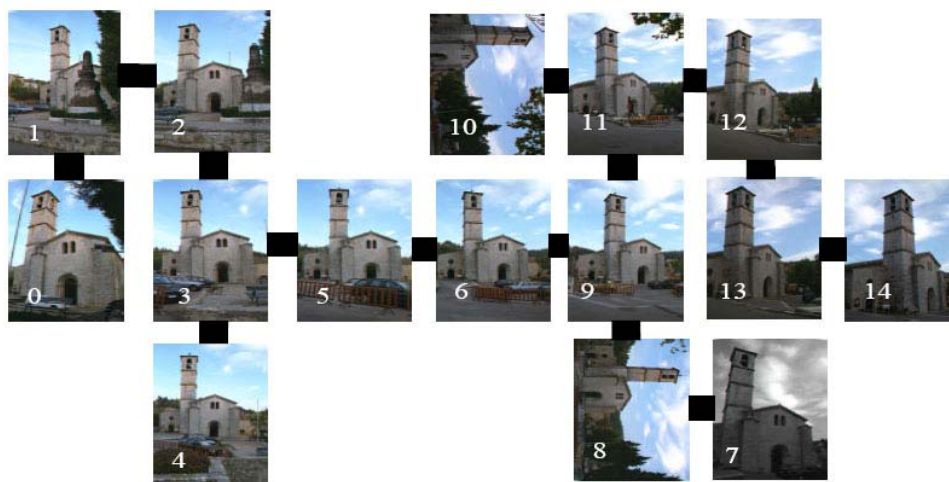
Vidosekvence

Další velmi důležitou možností vstupu je vidosekvence. Ve vidosekvenci víme, že snímky které jsou za sebou jsou snímány z přibližně stejných úhlů. Můžeme tedy použít skupiny (dvojice, trojice, ...) po sobě jdoucích snímků, které se mohou překrývat. Další výhodou vidosekvence je, že můžeme lépe detekovat korespondence, případně predikovat pozici bodů na dalším snímku (k tomu slouží například Kalmanův filtr).

K vidosekvenci, která není vytvořena přímo za účelem rekonstrukce, nebo nemá k rekonstrukci předpoklady je například film. U rekonstrukce z filmu nastává několik situací, které je potřeba ošetřit, musíme například detekovat střihy (po střihu se většinou mění scéna). Další velmi důležitou věcí je filtrovat motion-blur, který je přítomný na každém snímku kde došlo k rychlejšímu pohybu kamery, nebo pozorovaného objektu (Na odstranění motion-bluru se obvykle používá Wienerův filtr).



(a)



(b)

Obrázky 7.7.1¹³: Ukázka uspořádání rozptýlených snímků

7.8 Nestatická scéna

Již v úvodu této práce jsme uvedli, že jedinou omezující podmínkou je, aby scéna byla statická. Co však v případě, že scéna statická není, taková situace způsobuje, že odhad modelu scény není jednoznačný nebo vyhodnotíme pohybující se objekty jako outliery. Řešení prvního případu je velmi obtížné a naší snahou je převést ho na případ druhý, tedy rozdělit obraz na více částí a rekonstruovat právě jednu (a body z druhé části považovat za outliery). Vezměme jako příklad jedoucí auto, buď budeme rekonstruovat samotné auto, nebo jeho okolí (silnici, domy, atd.). Ve chvíli, kdy zrekonstruujeme jednu (statickou) část scény můžeme přistoupit k rekonstrukci zbývajících částí, které budou nyní snadno rozpoznatelné.

¹³ Zdroj obrázků: [Schaffalitzky-2002]

Kapitola 8

Experimentální ověření popsaných metod

V této kapitole budeme odděleně ověřovat vlastnosti jednotlivých metod na syntetických příkladech. Zároveň prověříme postup jako celek na reálných datech. V experimentech se zaměříme zejména na robustnost algoritmů vůči šumu a přítomnosti outlierů ve vstupních datech, případně na funkčnost samotných algoritmů. Všechny výsledky jsou získány zprůměrováním několika pokusů.

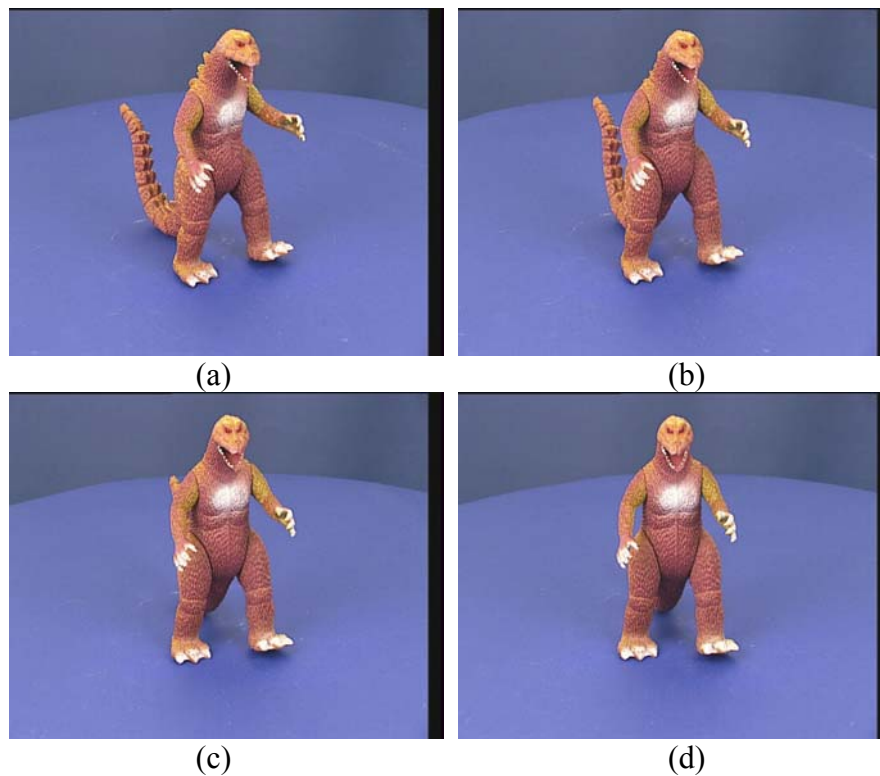
8.1 Bodové detektory

Porovnání Harrisova detektoru (kapitola (4.1.1)) s algoritmem FindFP (kapitola (4.1.2)). Hlavní sledovanou vlastností je repeatibilita. Oba algoritmy vyzkoušíme na třech sadách obrázků, které pokrývají nejčastější případy:

1. Pohled na objekt z různých pozic.
2. Otočení obrazu.
3. Změna měřítka.

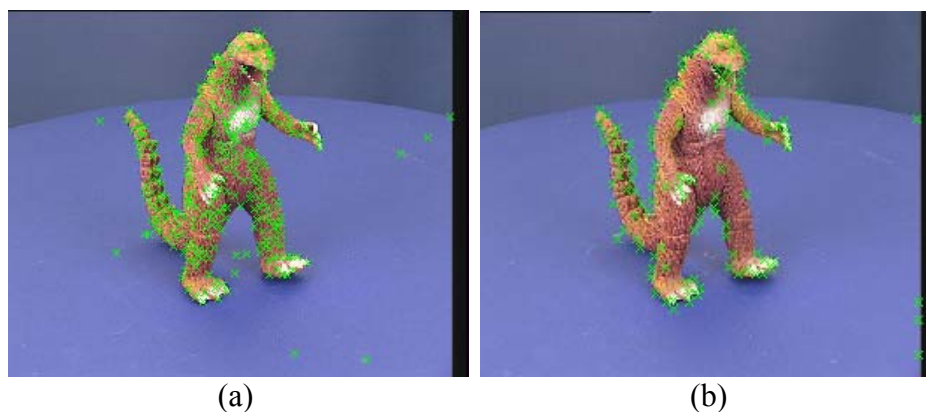
Add 1,

Jako vstupní obrázky byly použity fotografie dinosaura otočené vždy o 10° :



Obrázky 8.1.1¹⁴: Scéna na obrázcích je otáčena po 10-ti stupních.

¹⁴ Zdroj obrázků: URL= <http://www.robots.ox.ac.uk/~vgg/data1.html>



Obrázky 8.1.2: Významné body detekované Harrisovým detektorem (a) a metodou FindFP (b). Při použití Harrisova detektoru jsou na snímku body rovnoměrněji rozloženy, což vyplývá z charakteru algoritmu.

Repeabilitu testujeme na dvojicích snímků 1-2, 1-3, 1-4, ... , výsledky jsou zobrazeny v následující tabulce:

r [%]	α [°]					
	10	20	30	40	50	60
Harrisův detektor	66%	66%	62%	51%	39%	35%
FindFP	68%	55%	59%	58%	44%	37%

Tabulka 8.1.1: Závislost repeability r [%] na úhlu pohledu α [°], pro obě sledované metody. Počítají se pouze ty body, které jsou viditelné na obou snímcích. Ty body které jsou viditelné pouze na jednom snímku výsledky pochopitelně zhoršují.

Obě metody dosahují velice podobných výsledků, nesmíme ovšem zapomínat, že FindFP byl navržen pro rozmazané snímky, u kterých jsou jeho výsledky lepší (viz [Zitová-00]). Zajímavým důsledkem změny pozorovacího úhlu (pozice kamery) je, že se snižuje přesnost lokalizace bodů, tedy i když je detekován stejný bod scény, jeho souřadnice nejsou přesné.

Add 2,

Opět byl použit obrázek (8.1.1a), který byl postupně rotován podle středu po 10-ti stupních, výsledek experimentu shrnuje následující tabulka:

r [%]	β [°]			
	10	20	30	40
Harrisův detektor	86%	82%	85%	78%
FindFP	57%	60%	65%	65%

Tabulka 8.1.2: Závislost repeability r [%] na úhlu otočení obrázku β [°], pro obě sledované metody.

V tomto případě dosahuje Harrisův detektor podstatně lepších výsledků než FindFP.

Add 3,

Pro změnu měřítka byl použit opět obrázek (8.1.1a), měřítko bylo stejně jako rotace měněno ve 2D. Výsledek experimentu shrnuje následující tabulka

r [%]	s					
	0.6	0.8	0.9	1.1	1.2	1.4
Harrisův detektor	42%	68%	81%	78%	74%	60%
FindFP	40%	69%	75%	77%	65%	59%

Tabulka 8.1.3: Závislost repeatability r [%] na změně měřítka s , pro obě sledované metody.

Tento experiment má zajímavý důsledek: Chceme-li zvýšit robustnost vzhledem ke změně měřítka, můžeme obrázek nepatrně zmenšit (nebo zvětšit), opět detekujeme body na zmenšeném obrázku, body transformujeme do souřadnic původního obrázku (vrátíme změnu měřítka) a použijeme pouze ty body, které se opakují.

Velmi důležité je nastavení parametrů jednotlivých metod (zejména u algoritmu FindFP), které může významně ovlivnit kvalitu (repeatabilitu) a množství detekovaných bodů. Důležitým ukazatelem je rychlost algoritmů. Metoda FindFP byla přibližně 10-krát pomalejší, než Harrisův detektor, tento výsledek je však pouze přibližný. Je tomu tak proto, že algoritmy nebyly nijak optimalizovány a v některých případech nebyly implementovány zcela efektivně.

Zajímavým poznatkem, který uplatníme zejména v kapitole (8.2), je že spojením výstupu obou algoritmů se zvýší počet kvalitních bodů, což může zajistit lepší výsledky navazujících metod.

8.2 MEM-Pairs a RANSAC

Algoritmus RANSAC (kapitola (4.3)), přesto že pracuje s náhodou, je velmi dobře předvídatelný a velmi silně závislý na kvalitě vstupu (inicializační korespondence). Proto bude v této kapitole testován společně s algoritmem pro hledání inicializačních korespondencí MEM-Pairs (kapitola (4.2.2)). Výsledky tohoto experimentu můžeme vztáhnout přímo na algoritmus MEM-Pairs, zejména proto, že vliv RANSACu je možno snadno odhadnout (viz tabulka (4.3.1)).

Inicializační korespondence získané metodou MEM-Pairs jsou největší slabinou celého postupu. Je tomu tak proto, že korelace (stejně tak MEM-Pairs) není afinně invariantní. Řešení problému afinní invariance jsme naznačili v kapitole (4.2.3). I přes to, se při testování této metody zaměříme na zjištění vlastností algoritmu MEM-Pairs, při různých transformacích scény a obrazu. Zajímají nás stejné případy jako u bodových detektorů, tedy:

1. Pohled na objekt z různých pozic.
2. Otočení obrazu.
3. Změna měřítka

Sledovanou veličinou je zejména počet správných korespondencí. Vstupem algoritmu jsou body získané kombinací výstupu obou metod testovaných v kapitole (8.1) a odpovídající obrázky. Ukončovací podmínka u algoritmu RANSAC je stanovena tak, aby pravděpodobnost správného řešení byla větší než 99%. Z hlediska urychlení výpočtu byla použita modifikace RANSACU popsaná v kapitole (4.3.1), tedy R-RANSAC.

Add 1

Výsledky tohoto experimentu jsou shrnuty v následujících tabulkách:

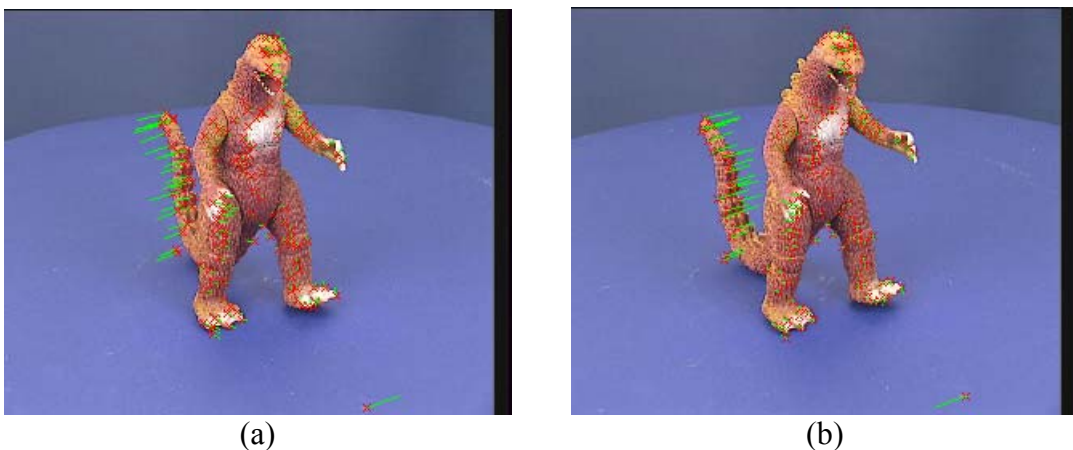
	α [°]					
	10		20		30	
Počet detekovaných bodů	425	414	425	351	425	342
Počet korespondencí MEM-Pairs	215		116		103	
Počet korespondencí RANSAC	148		33		-	

Tabulka 8.2.1: Závislost počtu korespondencí na úhlu pohledu α [°]. Počet detekovaných bodů se u jednotlivých snímků liší, uvádíme tedy dvě hodnoty. Prázdné pole znamená, že algoritmus nenalezl výsledek.

	α [°]		
	10	20	30
Čas výpočtu MEM-Pairs [s]	8.98	7.843	7.5
Počet iterací RANSAC	452	31724	-
Čas výpočtu RANSAC [s]	0.82	83.89	-

Tabulka 8.2.2: Závislost doby výpočtu a počtu iterací na úhlu pohledu α [°].

V tomto experimentu se jasně projeví nedostatky metody MEM-Pairs a ukazuje se potřeba afinně invariantního algoritmu pro hledání inicializačních korespondencí. Tato metoda je použitelná pouze pro velmi blízké snímky. Ukázkou výstupu RANSACu můžeme vidět na obrázcích (8.2.1).



Obrázky 8.2.1: Nalezené korespondence získané kombinací metod MEM-Pairs a RANSAC pro snímky získané z různých pohledů.

Add 2

Výsledky tohoto experimentu jsou shrnuty v následujících tabulkách:

	β [°]					
	10		20		30	
Počet detekovaných bodů	425	423	425	418	425	421
Počet korespondencí MEM-Pairs	194		132		116	
Počet korespondencí RANSAC	152		51		-	

Tabulka 8.2.3: Závislost počtu korespondencí na otočení β [°].

	β [°]		
	10	20	30
Čas výpočtu MEM-Pairs [s]	9.03	8.88	9
Počet iterací RANSAC	24	3581	-
Čas výpočtu RANSAC [s]	0.22	10.73	-

Tabulka 8.2.4: Závislost doby výpočtu a počtu iterací na otočení β [°].

Add 3

Výsledky tohoto experimentu jsou shrnuty v následujících tabulkách:

	$s = 0.6$		0.8		0.9	
	Počet detekovaných bodů	425	114	425	303	425
Počet korespondencí MEM-Pairs	64		146		232	
Počet korespondencí RANSAC	24		104		116	

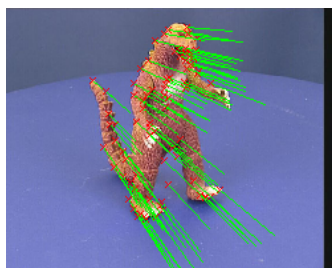
	$s = 1.1$		1.2		1.4	
	Počet detekovaných bodů	425	445	425	482	425
Počet korespondencí MEM-Pairs	274		196		142	
Počet korespondencí RANSAC	240		141		46	

Tabulka 8.2.3: Závislost počtu korespondencí na změně měřítka s .

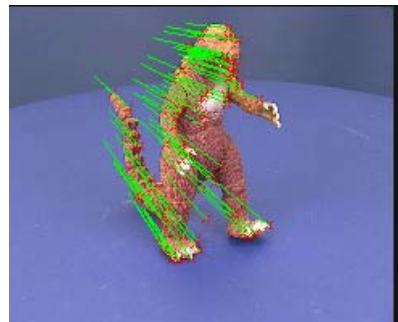
	$s = 0.6$			0.8			0.9		
	Čas výpočtu MEM-Pairs [s]	3.66			7.00			7.73	
Počet iterací RANSAC	6799			48			5		
Čas výpočtu RANSAC [s]	14.94			0.42			0.08		

	$s = 1.1$			1.2			1.4		
	Čas výpočtu MEM-Pairs [s]	9.5			9.91			11.33	
Počet iterací RANSAC	10			44			13606		
Čas výpočtu RANSAC [s]	0.312			0.25			42.42		

Tabulka 8.2.4: Závislost doby výpočtu a počtu iterací na změně měřítka s .



(a)



(b)

Obrázky 8.2.2: Nalezené korespondence získané kombinací metod MEM-Pairs a RANSAC pro snímky s různým měřítkem, (a) je o 20% menší než (b) .

Tento experiment ukazuje, že alespoň ke změně měřítka je MEM-Pairs částečně invariantní.

Urychlení algoritmu RANSAC

Jaký vliv má použití R-RANSACu na rychlost, oproti původnímu RANSACu, jsme již ukázali v kapitole (4.3.1). Urychlení RANSACu algoritmem PLUNDER je do velké míry závislé na použitých obrázcích a urychlení je zřejmé, z tabulky (4.3.1), která ukazuje nezbytný počet kroků RANSACu, při daném modelu, resp. modelu s daným počtem stupňů volnosti.

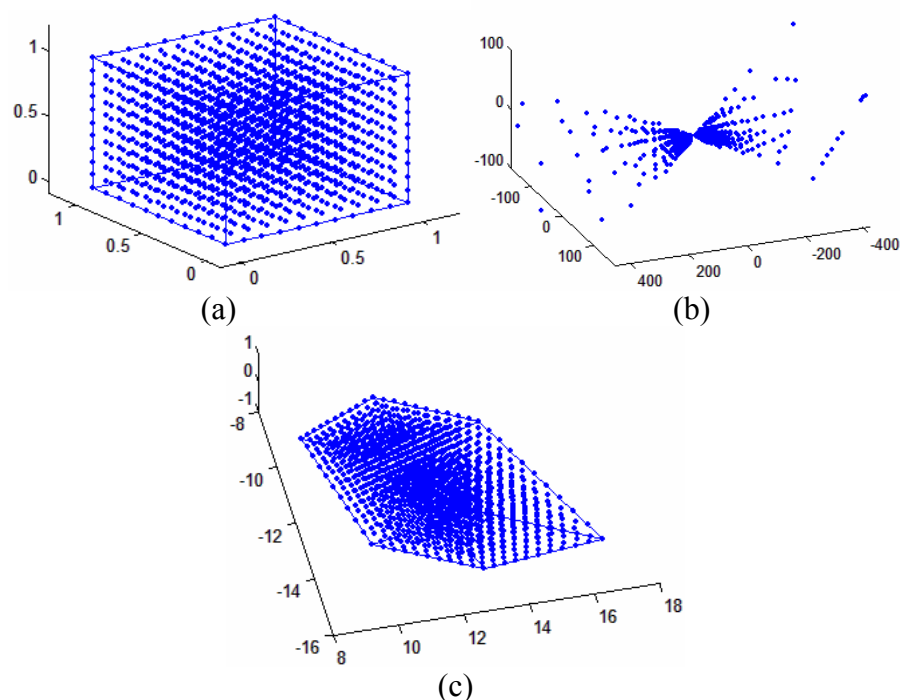
8.3 Faktorizace

Výsledky faktorizační metodě v závislosti na kvalitě vstupu, tedy šumu a outlierech.

8.3.1 Syntetická data

Nejprve vygenerujeme skupinu 3D bodů a tyto body promítneme několika kamerami. Do promítnutých bodů poté přidáme šum. Vypočteme projektivní hloubky a faktorizujeme. 3D body získané faktorizací zpět promítneme kamerami získanými faktorizací. Chybu výpočtu vyjádříme jako průměr vzdáleností nově promítnutých bodů od původně promítnutých, tuto chybu označíme ε_{2D} . U syntetických experimentů použijeme pro lepší interpretaci výsledků chybu $\varepsilon_{2D\%}$, která počítá průměr vzdáleností relativně, vzhledem k velikosti scény. Pro výpočet projektivních hloubek použijeme metodu založenou na epipolární geometrii. Velikost šumu, který budeme do dat vkládat je v jednotkách 0.001 velikosti scény, což přibližně odpovídá jednomu pixelu při rozlišení 800x600.

Na obrázku (8.3.1) vidíme vstupní data a možné výstupy projektivní rekonstrukce:



Obrázky 8.3.1: Vstupní data (a) a různé projektivní rekonstrukce (b) a (c). Obě tyto rekonstrukce jsou správné, nesmíme zapomínat že jde o projektivní rekonstrukci. Výsledky experimentu jsou shrnuty v následujících tabulkách:

chyba $\varepsilon_{2D\%}$	velikost šumu [0.1%]				
	5	10	15	20	25
1. snímek	0.46	0.94	1.30	1.86	2.13
2. snímek	0.63	1.22	1.95	2.30	3.18
3. snímek	0.11	0.15	0.35	0.30	0.74
průměr	0.40	0.77	1.20	1.49	2.02

Tabulka 8.3.1: Závislost chyby ε_{2D} na šumu, šum má rovnoměrné rozdělení a udáváme ho v desetínách procenta velikosti scény. V tomto experimentu byly použity 3 snímky s přibližně 1200 body.

chyba $\varepsilon_{2D\%}$	počet snímků					
	2	3	4	5	7	8
průměrná chyba	0.59	0.70	1.24	1.64	3.01	3.25

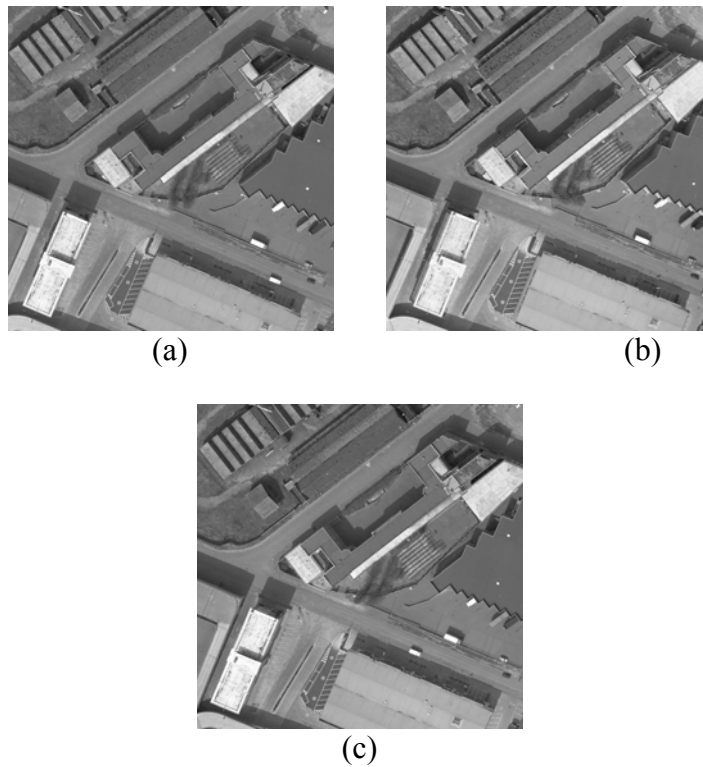
chyba $\varepsilon_{2D\%}$	počet snímků					
	9	10	13	15	18	20
průměrná chyba	3.42	4.11	3.87	3.93	4.92	5.05

Tabulka 8.3.2: Závislost chyby ε_{2D} na počtu snímků. Šum má rovnoměrné rozdělení a udáváme ho v desetínách procenta velikosti scény, hodnota šumu je v tomto případě 10 (tedy ± 10 pixelů při rozlišení 800x600).

Při větším počtu snímků je patrná zvětšující se chyba. To ovšem neznamená, že menší počet snímků je výhodnější, jak ukážeme v kapitole (8.4.1).

8.3.2 Reálná data

Při experimentu s reálnými daty využijeme výstupů z předchozích experimentů (kapitoly (8.1-8.4)). Vedle sady snímků (8.1.1) použijeme také několik leteckých snímků – obrázky (8.3.2).



Obrázky 8.3.2¹⁵: Trojice leteckých snímků z různých pozic.

korespondence pro trojici snímků byly získány jako průnik korespondencí mezi snímky 1-2 a 2-3. Výsledek experimentu je shrnut v následující tabulce:

Počet snímků	Počet korespondencí	Chyba ε_{2D}
Dinosaurus	98	0.49
Letecký pohled	172	0.23

Tabulka 8.3.3: Hodnota chyby ε_{2D} pro reálná data. Chyba ε_{2D} byla vypočtena jako průměr z chyb na jednotlivých snímcích.

Podobných výsledků, tedy průměrné chyby okolo 0.4 pixelu, jsme dosáhli u většiny snímků, které byly testovány. Jen výjimečně byly výsledky horší (ne více než 0.7 pixelu) nebo lepší (ne méně než 0.2 pixelu).

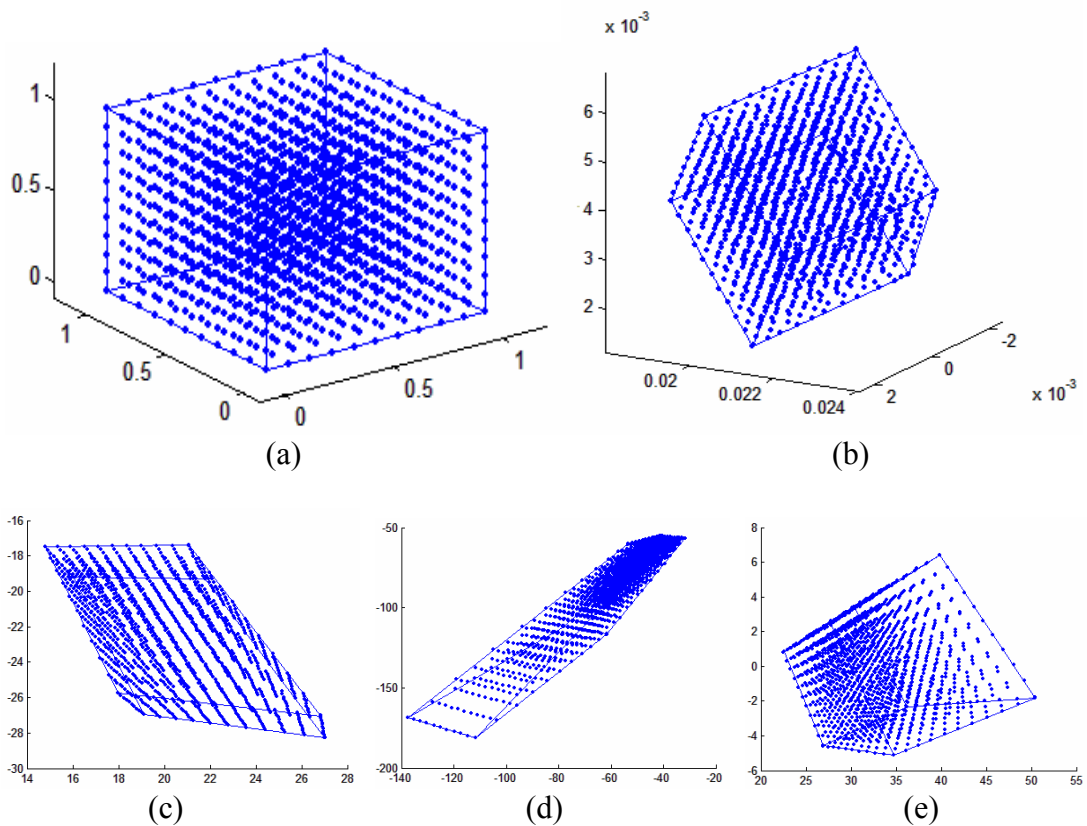
¹⁵ Zdroj obrázků: URL= <http://www.robots.ox.ac.uk/~vgg/data1.html>

8.4 Normalizace

Výstupem normalizačního algoritmu jsou 3D body a matice kamer získané projektivní rekonstrukcí.

8.4.1 Syntetická data

V tomto experimentu použijeme množinu bodů a kamer získaných projektivní rekonstrukcí v experimentu (4.3.1). Podobně jako u projektivní rekonstrukce budeme testovat robustnost algoritmu vůči šumu a vliv počtu použitých snímků na kvalitu výsledku.



Obrázky 8.4.1: Originální body (a) , jejich průměty různými kamerami (c,d,e) a metricky zrekonstruované body (b).

Protože zrekonstruované body jsou stále ještě podrobeny metrické rekonstrukci (otočení posunutí, změna měřítka), určíme chybovou funkci určující kvalitu rekonstrukce následujícím postupem:

1. Náhodně vybereme dvojici originálních bodů a vypočteme jejich vzájemnou vzdálenost l_i .
2. Vypočteme vzdálenost l'_i mezi dvojicí zrekonstruovaných bodů , které odpovídají bodům z kroku 1.
3. Vypočteme poměr mezi oběma vzdálenostmi $s_i = l_i / l'_i$.
4. Kroky 1–3 opakujeme pro N náhodných vzorků.

s_i určuje změnu měřítka mezi originálním a zrekonstruovaným prostorem. Hodnota s_i by měla být konstantní, chybu vyjádříme jako

$$\varepsilon = \frac{\sum_{i=1}^N \text{abs}(s_i - \bar{s})}{N}, \quad (8.4.1)$$

kde ε je průměr vzdáleností s_i od průměrné vzdálenosti \bar{s} .

Abychom zachovali nezávislost na hodnotě měřítka stanovíme chybu $\varepsilon_{\%}$, která je relativní vzhledem k \bar{s} a je vyjádřena v procentech:

$$\varepsilon_{\%} = \frac{\varepsilon}{\bar{s}} * 100 [\%], \quad (8.4.2)$$

Takové vyjádření chyby vychází z poznatku, že poměr vzdáleností je v metrickém prostoru invariantem, proto je takové měření chyby legitimní.

Výsledky metrické rekonstrukce jsou shrnuty v následujících tabulkách:

	velikost šumu [0.1%]					
	1	3	5	7	9	11
chyba $\varepsilon_{\%}$ [%]	1.19	1.42	3.56	3.64	7.82	8.75

Tabulka 8.4.1: Závislost chyby $\varepsilon_{\%}$ na šumu. Podobně jako u metrické rekonstrukce používáme tři snímky.

	počet snímků					
	3	4	5	7	8	9
chyba $\varepsilon_{\%}$ [%]	0.54	3.25	1.75	2.13	2.18	1.59

	počet snímků					
	10	13	15	18	20	25
chyba $\varepsilon_{\%}$ [%]	1.41	0.78	0.85	0.93	0.86	0.86

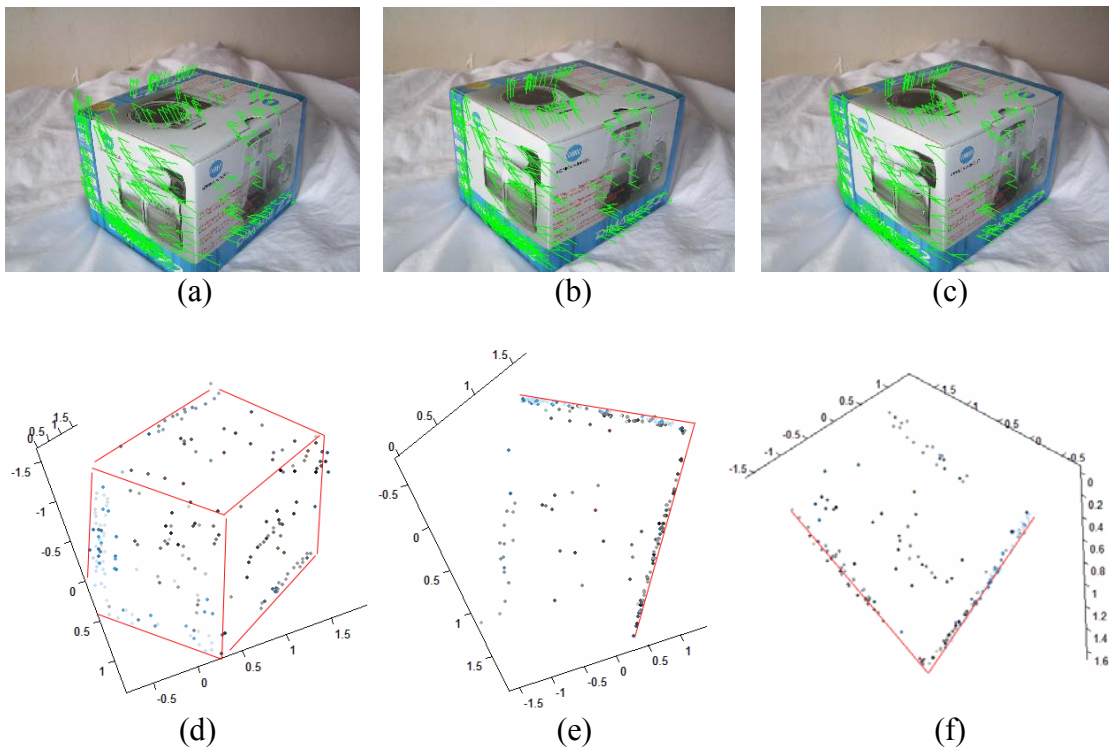
Tabulka 8.4.2: Závislost chyby $\varepsilon_{\%}$ na počtu snímků. Šum je nastaven na 1 (tedy ± 1 pixel při rozlišení 800x600).

I přes to, že výsledky projektivní geometrie naznačovaly, že zvýšení počtu snímků zhoršuje výsledky, v tabulce (8.4.2) vidíme, že je tomu právě naopak.

8.4.2 Reálná data

Uvažujeme pouze případ 1 normalizačního algoritmu tak jak byl popsán v kapitole (6.3.1). Z toho důvodu použijeme jiné snímky než v předchozích experimentech, ve kterých neznáme s jistotou polohu principálního bodu.

Při experimentech s reálnými daty narážíme na problém, že implementované hledání korespondencí není v momentální fázi vývoje dostatečně robustní, aby bylo možno uspokojivě detekovat větší množství korespondencí na trojici snímků. Pro metrickou rekonstrukci je tedy potřeba, aby snímky byly snímány z téměř stejné pozice, což zajistí, že bude nalezeno velké množství korespondencí.



Obrázky 8.4.2: Demonstrace metrické rekonstrukce s reálnými daty. Originální snímky s vyznačenými korespondencemi (a,b,c). Zrekonstruovaná množina bodů (d) pohled odpovídající původním snímkům, (e) pohled shora, (f) pohled z boku. Červenými čarami jsou zvýrazněny hrany objektu.

Ve výsledku Metrické rekonstrukce jsou jasně patrné tři vzájemně kolmé roviny, což odpovídá původnímu objektu.

Kapitola 9

Implementační část

Implementace byla rozdělena na dvě hlavní části:

1. Jednotlivé metody byly vyvíjeny a laděny v prostředí MatLab¹⁶.
2. Výsledná aplikace byla naprogramována v jazyce C#, kde byly použity odladěné algoritmy z MatLab-u.

Experimenty, provedené v kapitole 8 byly kompletně provedeny v MatLabu.

Implementováno je pouze několik vybraných metod z těch které byly popsány v předchozích kapitolách. Tyto metody však na základní úrovni dostačují k tomu, aby mohla být scéna zrekonstruována.

Algoritmy jednotlivých metod jsou poměrně podrobně popsány v příslušných kapitolách, v této kapitole shrneme jednotlivé funkce vytvořené v MatLabu. Aplikace vytvořená v jazyce C# je zdokumentována v programátorském a uživatelském manuálu (viz přílohy (B,C)). Jednou z nejdůležitějších metod 3D rekonstrukce je Singular Value Decomposition (viz Příloha (A)), jejíž implementace byla převzata z vynikající publikace [Press-95], která obsahuje programovou realizaci celé řady numerických metod.

Zpracování videa

Aplikace v C# obsahuje funkci, která vytvoří z *.avi souboru bitmapy, program je tedy možné použít na videosekvenci tak, že detekujeme korespondence vždy na dvojici po sobě jdoucích snímků.

LCD brýle

Protože algoritmus v momentální fázi vývoje nemá uspokojivý 3D výstup, pouze množinu bodů, není možno výsledky uspokojivě trojrozměrně vizualizovat.

Při stereoskopickém zobrazování pomocí brýlí s LCD clonou (LCD Shutter Glasses) se levý a pravý obraz střídá na displeji s pevně danou frekvencí. Pokud pozorovatel sleduje tento obraz skrz LCD brýle, clony obou polovin brýlí jsou synchronizovány s displejem tak, aby každé oko vždy vidělo pouze obraz jemu určený.

Protože použitá (ani žádná jiná existující) implementace OpenGL v jazyce C# neumožňuje využití LCD brýlí (funkce GLUTu), a zároveň výstup programu není pro vizualizaci LCD brýlemi vhodný nebyla implementace LCD brýlí v programu implementována.

Tento problém jsme vyřešili externím programem napsaným v programovacím jazyce Delphi, který načte výsledek rekonstrukce ze souboru a zobrazí na stereo brýlích.

¹⁶ Z anglického Matrix Laboratory: Software vyvinutý americkou společností The MathWorks, Inc.

Funkce vytvořené v MatLabu a jejich popis

<i>Harris</i>	vstup : Šedotónový snímek I , maximální počet bodů No . výstup : Množina detekovaných bodů M . popis : Implementace <i>Harrisova detektoru</i> , tak jak byl popsán v kapitole (4.1.1).
<i>FindFP</i>	vstup : Šedotónový snímek I , maximální počet bodů No . výstup : Množina detekovaných bodů M . popis : Implementace algoritmu <i>FindFP</i> , tak jak byl popsán v kapitole (4.1.2).
<i>MEM_PAIRS</i>	vstup : Dvojice snímků I1 , I2 Detekované body na prvním a druhém snímku M , N . výstup : Seznam korespondujících bodů P . popis : Implementace algoritmu <i>MEM_Pairs</i> , tak jak byl popsán v kapitole (4.2.2).
<i>Normalize</i>	vstup : Množina 2D bodů X . výstup : Transformační matice A . popis : Vypočte matici A , kterou můžeme body transformovat tak, aby jejich těžiště bylo v počátku a průměrná vzdálenost od počátku byla rovna $\sqrt{2}$.
<i>FMatrix_4_affine</i>	vstup : 2 množiny 2D bodů X1 , X2 (korespondující body). výstup : Fundamentální matice F . popis : Vypočte epipolární geometrii afinní kamery 4-bodovým algoritmem.
<i>FMatrix_8</i>	vstup : 2 množiny 2D bodů X1 , X2 (korespondující body). výstup : Fundamentální matice F . popis : Vypočte epipolární geometrii projektivní kamery 8-bodovým algoritmem.
<i>FMatrix_8n</i>	vstup : 2 množiny 2D bodů X1 , X2 (korespondující body). výstup : Fundamentální matice F . popis : Vypočte epipolární geometrii projektivní kamery 8-bodovým normalizovaným algoritmem (využívá funkci <i>Normalize</i>).

<i>FMatrix_7</i>	<p>vstup : 2 množiny 2D bodů X1, X2 (korespondující body) .</p> <p>výstup : Fundamentální matice F.</p> <p>popis : Vypočte epipolární geometrii projektivní kamery 7-bodovým algoritmem.</p>
<i>FMatrix_7n</i>	<p>vstup : 2 množiny 2D bodů X1, X2 (korespondující body) .</p> <p>výstup : Fundamentální matice F.</p> <p>popis : Vypočte epipolární geometrii projektivní kamery 7-bodovým normalizovaným algoritmem (využívá funkci <i>Normalize</i>).</p>
<i>RANSAC7</i>	<p>vstup : 2 množiny 2D bodů X1, X2 (korespondující body) .</p> <p>výstup : Fundamentální matice F, množina inliers které jsou rovny jedné u správné korespondence a rovna 0 u chybné korespondence.</p> <p>popis : Na základě 7-mi bodového algoritmu odhadne model epipolární geometrie RANSACem, který je popsán v kapitole (4.3).</p>
<i>PRec</i>	<p>vstup : Matice korespondencí W.</p> <p>výstup : Projektivně zrekonstruované kamery PR a projektivně zrekonstruované 3D body XR. Matice W doplněná o projektivní hloubky a projektivní hloubky L.</p> <p>popis : Projektivní rekonstrukce, kapitola (5).</p>
<i>MRec</i>	<p>vstup : Matice korespondencí s projektivními hloubkami W, projektivní hloubky L, projektivně zrekonstruované kamery PR a projektivně zrekonstruované 3D body XR.</p> <p>výstup : Metricky zrekonstruované kamery PRm a projektivně zrekonstruované 3D body XRm.</p> <p>popis : Metrická rekonstrukce, kapitola (6).</p>

Kapitola 10

Závěr

V této práci jsme zkoumali principy a možnosti rekonstrukce 3D scény z 2D pohledů. Popsali jsme základní teorii která je pro rekonstrukci nezbytná. Na základě této teorie byly pro každou fázi postupu rekonstrukce implementovány a testovány nejdůležitější algoritmy (Harrisův bodový detektor, RANSAC, Faktorizace, Normalizace). Je-li vstupem místo množiny snímků videosekvence, můžeme s tím počítat a určitým způsobem modifikovat základní algoritmy. Jednotlivé modifikace jsou popsány vždy u příslušných kapitol.

Hlavním přínosem této práce je, že shrnuje celý proces rekonstrukce, který se skládá z několika odlišných problémů a je značně rozsáhlý. Naším cílem není aby byla každá z těchto částí dovedena k dokonalosti, ale aby byl celý proces do určité míry funkční.

V kapitole (7) jsme ukázali jak by měla 3D rekonstrukce ideálně v budoucnu vypadat. Popsali jsme cesty a metody, které zkvalitňují výsledek rekonstrukce. Tyto postupy umožňují dosáhnout praktické využitelnosti.

Tato práce dává kvalitní základ pro další výzkum v oblasti 3D rekonstrukce.

Literatura

- [Baumberg-00] A. Baumberg, *Reliable feature matching across widely separated views*, CVPR, p612-618, 2000.
- [Chai-98] J. Chai, S. D. Ma, *Robust Epipolar Geometry Estimation Using Genetic Algorithm*, ACCV 1998, p. 272-279, 1998.
- [Chum-01] O. Chum, *Rekonstrukce 3D scény z korespondenci v obrazech*, Master thesis, 2001. In Czech.
- [Chum-02] O. Chum and J. Matas, *Randomized RANSAC and T(d,d) test*, BMVC 2002, 2002. In Czech.
- [Denton-02] J. Denton, J. R. Beveridge, *Two Dimensional Projective Point Matching*, SSIAI 2002, p. 77-81, 2002.
- [Devernay-95] F. Devernay, O. D. Faugeras, *Automatic calibration and removal of distortion from scenes of structured environments*, SPIE vol 2567, p. 62-72, 1995.
- [Edberg-00] C. Edberg and A. Ericsson, *3D reconstruction from uncalibrated images*, Master thesis, 2000.
- [Faugeras-92] O.D. Faugeras, *What can be seen in tree dimension with an uncalibrated stereo rig?*, ECCV 1992, p. 563-578, 1992.
- [Fischler-81] M. Fischler and R. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Communication of the ACM, vol. 24, p. 381-395, 1981.
- [Golub-96] G. H. Golub and C. F. Van Loan, *Matrix Computations 3rd Edition*, Baltimore, MD: Johns Hopkins University Press, 1996.
- [Harris-88] C. Harris and M. Stephens, *A combined corner and edge detector*, in Proc. 4th Alvey Vision Conf. 1988, p. 189 -192, 1988.
- [Hartley-92] R. I. Hartley, *Estimation of relative positions for uncalibrated cameras*, European Conference on Computer Vision, LNCS 5888, p. 579-587, 1992.
- [Hartley-94a] R. I. Hartley and P. Sturm, *Triangularization*, Proceedings of the ARPA Image Understanding Workshop, Defense Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc., p. 957-966, 1994.
- [Hartley-94b] R.I. Hartley, *Lines and points in three views - a unied approach*, IUW'94, p. 1009-1016, 1994.
- [Hartley-95] R. I. Hartley, *In defence of the 8-point-algorithm*, In: Proc. Fifth International Conference on Computer Vision, IEEE Computer Society Press, p. 1064-1070, June 1995.
- [Hartley-04] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision 2nd Edition*, Cambridge University Press, ISBN 0521540518, 2004.
- [Han-00] M. Han and T. Kanade, *Creating 3D models with uncalibrated cameras*, In proceeding of IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000), 2000.
- [Jacobs-97] D. Jacobs, *Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images*, CVPR, p. 206-212, 1997.

- [Janyška-01] J. Janyška and A. Sekaninová, *Analytická teorie kuželoseček a kvadrik 2. vydání*, Brno: Masarykova univerzita v Brně, ISBN 8021026049, 2001. In Czech.
- [Kolb-95] C. Kolb, D. Mitchell, and P. Hanrahan, *A realistic camera model for computer graphics*, SIGGRAPH 1995, p. 317-324, 1995.
- [Lindeberg-97] T. Lindeberg a J. Gårding, *Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure*, IVC 1997, p 415-434, 1997.
- [Lundberg-00] F. Lundberg, *Maximum entropy matching: An approach to fast template matching*, report nr. LiTH-ISY-R-2313 , Linköping university Sweden, 2000.
- [Martinec-00] D. Martinec, *Konzistentní rekonstrukce scény z velkého množství snímků*, Master thesis, 2000. In Czech.
- [Mikolajczyk-02] K. Mikolajczyk and C. Schmid, *An affine invariant interest point detector*, In Proc. ECCV. SpringerVerlag, 2002.
- [Moons-98] T. Moons, *A guided tour through multiview relations*, In Proceedings SMILE Workshop (post-ECCV '98), LNCS. Springer, 1998.
- [Montesinos-98] P. Montesinos, V. Gouet a R. Deriche, *Differential invariants for color images*, ICPR 1998, p. 838-840, 1998.
- [Press-95] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C, Second edition*, Cambridge University Press, 1995.
- [Sainz-02] M. Sainz and N Bagherzadeh, *Recovering 3d metric structure and motion from multiple uncalibrated cameras*, In proceeding of IEEE Conference on Information Technology: Coding and Computing 2002, p. 268-273, 2002
- [Schaffalitzky-02] F. Schaffalitzky and A. Zisserman, *Multi-view matching for unordered image sets, or „How do I organize my holiday snaps?“*, Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, p. 414-431, 2002
- [Scott-91] G. L. Scott and H. C. Longuet-Higgins, *An algorithm for associating the features for two images*, Proceedings of the Royal Statistical Society of London, volume 244, p. 21-26, 1991.
- [Schmid-98] C. Schmid, R. Mohr, and C. Bauckhage, *Comparing and evaluating interest points*, In Proc. of the 6th International Conference on Computer Vision, p. 230-235, Bombay, 1998.
- [Shashua-94] A. Shashua, *Trilinearity in visual recognition by alignment*, ECCV'94, Lecture Notes in Computer Science, Vol. 800, 1994.
- [Sturm-96] P. Sturm and B Triggs, *A factorization based algorithm for multi-image projective structure and motion*, 4th European Conference on Computer Vision, p. 709-720, 1996.
- [Strecha-04] C. Strecha, R. Fransens, L. Van Gool , *"Wide-baseline Stereo from Multiple Views: a Probabilistic Account"*, CVPR 2004, 2004.
- [Šeda-04] J. Šeda, *Konverze avi videa na volumetrická data*, Bc. thesis, 2004. In Czech.
- [Torr-97] P. Torr and A. Zisserman, *Robust parametrization and computation of the trifocal tensor*, Image and Vision Computing, 15(1997), p. 591-605, 1997.

- [Torr-98] P. Torr, A. Zisserman, S. J. Maybank, *Robust detection of degenerate configurations while estimating the fundamental matrix*, Computer Vision and Image Understanding: CVIU, p. 312-333, 1998.
- [Zitová-00] B. Zitová, J. Flusser, J. Kautský, and G. Peters, *Feature point detection in multiframe images*, Proceedings of Czech Pattern Recognition Workshop 2000, p. 117-122, 2000.

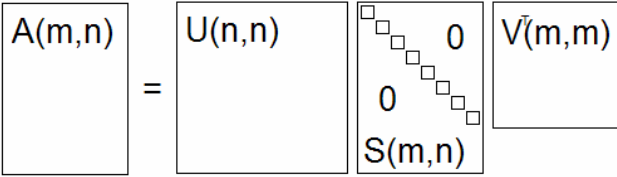
Příloha A

Singular value decomposition (SVD)

Singular Value Decomposition (rozklad na singulární hodnoty), často označována SVD. Necht' A je libovolná matice o rozměrech m, n . SVD nazýváme rozklad matice A na:

$$A = USV^T, \quad (\text{A.1})$$

kde $U(n, n)$ a $V(m, m)$ jsou ortonormální matice a S je matice diagonální $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, $r = \min(m, n)$.



$$\boxed{A(m,n)} = \boxed{U(n,n)} \boxed{S(m,n)} \boxed{V^T(m,m)} \quad (\text{A.2})$$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ nazýváme singulární hodnoty.

Matice U a V jsou ortonormální, to znamená, že jejich sloupce jsou lineárně závislé a jejich norma je rovna jedné. To ve svém důsledku znamená, že determinant je roven 1 nebo -1 a inverzní matice je maticí transponovanou $U^T = U^{-1}$, $V^T = V^{-1}$.

σ_i^2 jsou vlastní čísla matice AA^T nebo $A^T A$ a sloupce matic U a V , u_i a v_i jsou vlastní vektory AA^T resp. $A^T A$ odpovídající vlastním číslům σ_i^2 .

Jedna věta bez důkazu: Rozklad na singulární hodnoty SVD existuje pro každou matici.

Další informace o SVD a zejména o její implementaci lze získat v publikaci [Press-95], podrobnější teoretické pozadí v [Golub-96].

V kapitolách (A.1-A.3) si předvedeme několik algebraických problémů, které je možno užitím SVD vyřešit. Metoda SVD má velmi široké uplatnění, příkladem může být její aplikace při hledání korespondencí ve stereo-vidění [Scott-91] nebo použití v metodě projektivní rekonstrukce [Sturm-96].

A.1 Řešení soustavy homogenních rovnic

Hledáme nenulový vektor x , který nejlépe řeší homogenní soustavu $Ax = 0$, tedy minimalizuje Ax .

Nejprve provedeme SVD rozklad $A = USV^T$. Je-li $V = [v_1 | \dots | v_m]$ matice vlastních vektorů, pak v_m je vlastní vektor korespondující nejmenšímu vlastnímu číslu a zároveň řešením homogenní soustavy. Tedy řešením je $x = v_m$.

Chyba tohoto řešení $\varepsilon = \|Ax\|$ je rovna nejmenší singulární hodnotě σ_m .

Je-li počet nezávislých řádek matice A roven počtu sloupců je nalezené řešení přesné. Je-li menší (nedourčená soustava), nebo větší (přeuročená soustava) je nalezené řešení minimální ve smyslu metody nejmenších čtverců.

A.2 Pseudoinverze a řešení soustavy nehomogenních rovnic

Inverze A^{-1} matice A existuje pouze je-li matice A čtvercová a má plnou hodnot, pak řešením $Ax = b$ je $x = A^{-1}b$. Pseudoinverze A^+ je zobecněním inverze a existuje pro libovolnou (m, n) matici. Rovnice $Ax = b$ má řešení $x = A^+b$, dokážeme-li určit pseudoinverzi matice můžeme tedy řešit homogenní soustavu.

Výpočet pseudoinverzní matice pomocí SVD: Nejprve provedeme SVD rozklad $A = USV^T$. Pseudoinverzní matici získáme jako:

$$A^{-1} = VS^{-1}U^T \quad (\text{A.2.1})$$

Protože matice $S = \text{diag}(\sigma_1, \dots, \sigma_r)$ je diagonální S^{-1} určíme jednoduše jako $S^{-1} = \text{diag}(1/\sigma_1, \dots, 1/\sigma_r)$, je-li $\sigma_i = 0$, položíme $S^{-1}(i, j) = 0$.

A.3 Snížení hodnosti matice

Nejprve provedeme SVD rozklad $A = USV^T$, kde $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$. Požadujeme-li, upravit matici A na hodnot h , vytvoříme matici $S' = \text{diag}(\sigma_1, \dots, \sigma_h, 0, \dots, 0)$ a opět složíme do matice

$$A' = US'V^T. \quad (\text{A.3.1})$$

Protože matice S' má hodnotu maximálně h a matice U a V jsou ortonormální, jejich hodnota je n (resp. m) $< h$, výsledná matice A' bude mít hodnotu rovnou hodnotě matice S' (tedy $\leq h$).

Tímto postupem vytvoříme matici A' s požadovanou hodnotí a zároveň minimalizujeme Frobeniovu normu $\|A - A'\|_F$. Frobeniova maticová norma je obdobou L2 normy pro vektory a je definována jako:

$$\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2}. \quad (\text{A.3.2})$$

A.4 Rozklad $Q=A^T A$

Potřebujeme-li rozložit matici Q na součin $A^T A$, je nutnou podmínkou, aby matice Q měla kladná čísla na diagonále a byla symetrická, ideálním případem je symetrická pozitivně definitní matice. Matici Q můžeme rozdělit SVD rozkladem na $Q = USV^T$, splňuje-li Q popsané podmínky, pak platí, že $U = V$. Dekomponujeme-li matici S na $S = S'S'$, protože je S diagonální postačuje k získání S' odmocnit všechny prvky S . Pak můžeme psát

$$Q = USU^T = \underbrace{US'}_A \underbrace{S'U^T}_{A^T} = AA^T. \quad (\text{A.4.1})$$

Příloha B

Programátorský manuál

Před prostudováním programátorského manuálu je dobré, alespoň zhruba prostudovat manuál uživatelský (příloha (C)), ve kterém čtenář získá představu o vzhledu programu a bude snazší pochopit jak funguje programová část.

Protože jazyk C# umožňuje automaticky generovanou dokumentaci z komentářů v programu, je tato umístěna na přiloženém CD. V této kapitole se tedy budeme zabývat pouze principiálním popisem jednotlivých tříd.

Program můžeme rozdělit do tří částí

1. GUI¹⁷ - Windows aplikace (okna). Zajišťuje otevírání a úpravu obrázků a sekvencí obrázků, dále slouží k zobrazení panelu nástrojů a spouštění vlastních algoritmů.
2. Panely nástrojů. Panel nástrojů je zobrazen v hlavním okně a mění se v závislosti na vybrané metodě, tedy každá metoda má vlastní formulář, který se automaticky zobrazuje při výběru metody.
3. Vlastní algoritmická část. Obsahuje konkrétní algoritmy, které jsou striktně odděleny od GUI (část 1) a jsou spouštěny z příslušného panelu nástrojů, který jim připraví data zobrazená v GUI. Dále obsahuje implementaci metody SVD (viz příloha (A)) a knihovnu pro maticové operace.

Vstupní metoda programu je umístěna ve třídě *MainClass*, která pouze spouští hlavní okno – *MainForm*, viz (B.1) GUI.

Program mimo základních funkcí umí také konvertovat video ve formátu avi na bitmapy, implementace této části aplikace byla získána z [Šeda-04].

B.1 GUI

Třída *MainForm* reprezentuje hlavní okno, aplikace byla vytvořena jako MDI¹⁸. Tato třída obsahuje otevírání souborů ovládání některých funkcí (úprava obrázků), zajišťuje správný kontext panelu nástrojů a správné vykreslení dokumentů.

Pravděpodobně nejdůležitější třídou tohoto bloku je třída *PictureControl*, která zajišťuje nejen vizualizaci obrázků (nebo sekvence obrázků) ale zároveň umožňuje vizualizovat detekované body, korespondence. Další podstatnou vlastností třídy *PictureControl* je, že ve spolupráci s některými nástroji umožňuje zadávat body, korespondence, apod. Výsledky rekonstrukce jsou zobrazeny v *PictureControl3D*, který využívá OpenGL. *PictureControl3D* vizualizuje množinu rekonstruovaných bodů.

Do GUI dále zahrnujeme několik dalších podpůrných dialogů pro výběr otevřených souborů *ImagesSelection*, pro konverzi videa na obrázky *Avi2Bmp*, atd. Příklady těchto dialogů naleznete v uživatelském manuálu (příloha (C)).

¹⁷ GUI – Graphics User Interface – Grafické uživatelské rozhraní

¹⁸ MDI – Multi Document Interface – Hlavní okno obsahuje dokumenty (viz uživatelský manuál, příloha (C))

Z programátorského hlediska není tato část nijak zajímavá (jde pouze o hrubou programátorskou práci) a není tedy potřeba podrobně ji rozebírat.

B.2 Panely Nástrojů

Panel nástrojů je vizuální rozhraní mezi GUI a algoritmickeou částí. Panely nástrojů jsou zděděny od třídy *ToolControl*. Panel nástrojů umožňuje nastavit parametry metody ke které přísluší. Dále zajišťuje vstupy algoritmů, tedy konverguje data (obrázky, body, korespondence) z podoby jakou mají v aktivním *PictureControlu* do podoby požadované daným algoritmem. Například vstupem Bodového detektoru je dvourozměrné pole, zatímco v *PictureControlu* je obraz reprezentován objektem *Bitmap*, je tedy potřeba příslušné pole z bitmapy získat.

Pro každý algoritmus existuje právě jeden nástrojový panel, následuje seznam jednotlivých nástrojových panelů:

ManPointSelectionControl

- Nástroj pro manuální výběr a mazání bodů.

ManCorrControl

- Nástroj pro manuální výběr korespondujících bodů.

HarrisControl

- Rozhraní pro Harrisův detektor, umožňuje nastavit parametry Harrisova detektoru, vybrat snímky, na kterých se má detekce provést a provedení vlastní detekce.

MEMPairsControl

- Rozhraní pro algoritmus MEM-Pairs, tj. hledání korespondencí. Pracuje s detekovanými, nebo zadanými body.

RANSACControl

- Rozhraní pro algoritmus RANSAC, tj. robustní vylepšení odhadu korespondencí. Pracuje s detekovanými, nebo zadanými korespondencemi.

PRecControl

- Rozhraní pro algoritmus Projektivní rekonstrukce. Výstup není možno vizualizovat, je tedy vytvořen *PictureControl3D*, který nic neobsahuje, pouze obsahuje vypočtená data

MRecControl

- Rozhraní pro algoritmus Metrické rekonstrukce. Vstupem je výsledek projektivní rekonstrukce v aktivním *PictureControl3D*, výstup je zobrazen přímo v *PictureControl3D-u*, ze kterého byla získána vstupní data.

B.3 Algoritmická část

Tato část obsahuje metody, které se přímo zabývají jednotlivými kroky rekonstrukce. Algoritmická část je zcela oddělena od GUI, což umožňuje jednoduchou výměnu, nebo přidání nových metod.

Soubory *Vector*, *Matrix*, *SVD*, obsahují třídy a metody které jsou nezbytné pro snazší programování matematicky založených algoritmů, např. násobení matic, výpočet inverzní matice, Třída *ImProcessing* obsahuje několik základních funkcí pro úpravu obrázků: konvoluce, dilatace, apod. .

Hlavní implementované algoritmy jsou reprezentovány následujícími třídami

Harris

– Algoritmus detekující body na šedotónovém obrázku. Vstupem je dvourozměrné pole, integrační měřítko, derivační měřítko a maximální povolený počet korespondencí výstupem je pak pole obsahující nalezené body.

MEMPairs

– Algoritmus hledající inicializační korespondence na dvou snímcích. Vstupem je trojice polí (snímky) a detekované body na každém snímku. Výstupem jsou pak nalezené korespondence.

RANSAC

– Algoritmus na robustní zpřesnění nalezených (zadaných) korespondencí. Vstupem jsou pouze korespondence, výstupem jsou ty korespondence, které vyhovují nalezenému modelu

PRec

– Algoritmus projektivní rekonstrukce, vstupem jsou korespondence získané RANSACem, nebo manuálně zadané. Výstupem jsou projektivně transformované matice kamer a pole 3D bodů.

MRec

– Algoritmus metrické rekonstrukce, vstupem je výsledek rekonstrukce projektivní. Výstupem jsou metricky transformované matice kamer a pole 3D bodů.

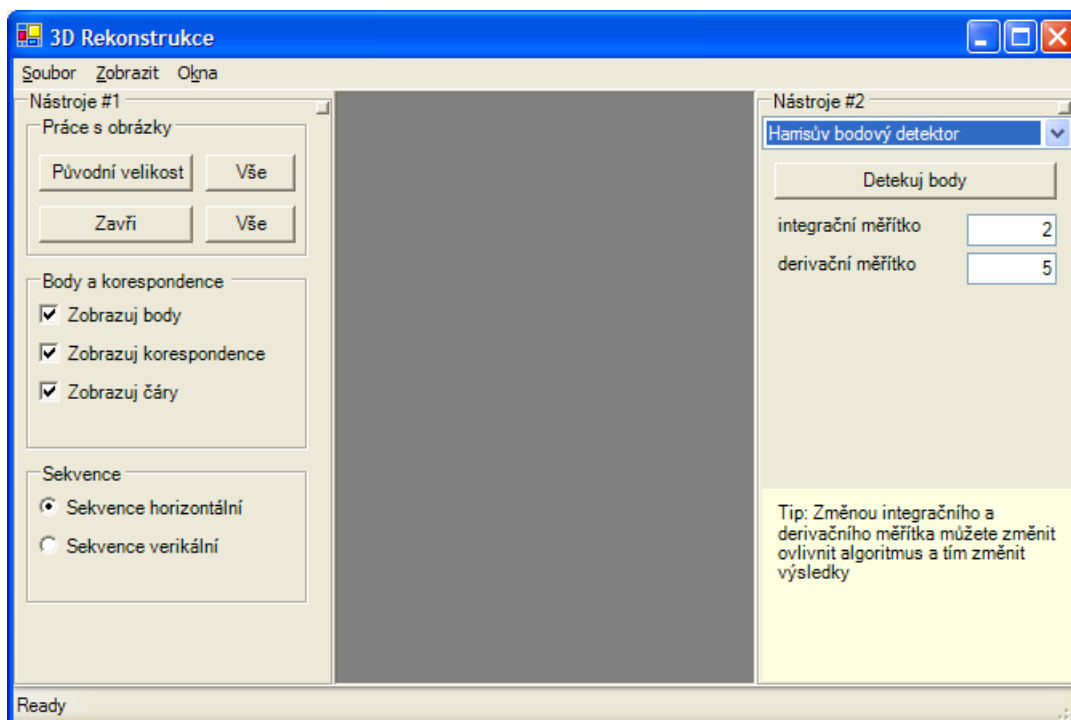
Teoretické pozadí implementovaných metod je možno nalézt v příslušných kapitolách (4, 5, 6). Program byl vytvořen tak, aby bylo jednoduchým způsobem přidávat další nástroje, nebo upravovat nástroje existující.

Příloha C

Uživatelský manuál k programu

Aplikace nepotřebuje žádnou speciální instalaci, stačí spustit soubor rekonstrukce.exe. Program je napsán v jazyce C#, k jeho funkci je tedy potřeba mít nainstalován .Net Framework 1.1¹⁹.

Po spuštění programu se otevře hlavní okno aplikace, viz obrázek (C.1)



Obrázek C.1: Hlavní okno aplikace.

Na obrázku (C.1) vidíme lištu hlavního menu a dva nástrojové panely. Hlavní menu a Nástroje #1 se nemění, panel Nástroje #2 je závislý na vybraném nástroji, mění se tedy s tím jaký nástroj je vybrán. V panelu Nástroje #2 je možno nastavit jednotlivé parametry vybrané metody, například na obrázku (C.1) je vybraným nástrojem Harrisův detektor

¹⁹ .Net Framework 1.1 je možno stáhnout z URL=<http://msdn.microsoft.com/netframework/downloads/>

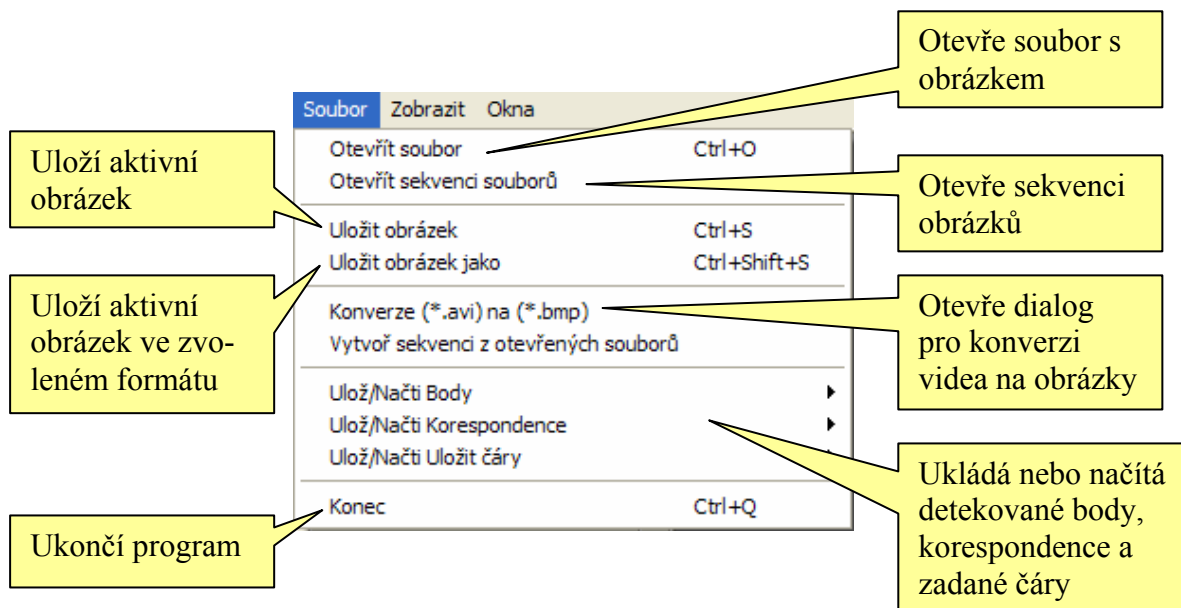
Popis hlavní nabídky

Nástrojové panely je možno vypnout, nebo zapnout v hlavním menu:

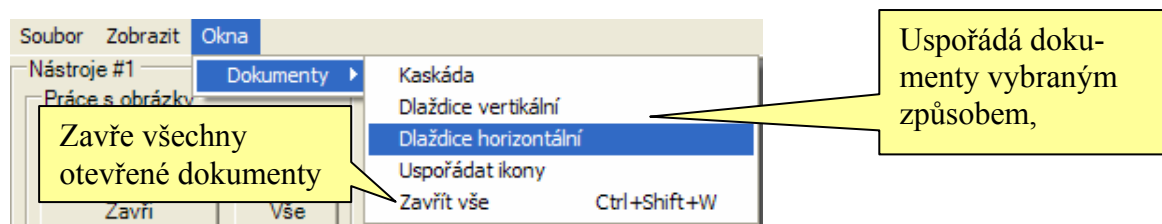


Obrázek C.2: Vypínání a zapínání nástrojových panelů.

Podrobný popis jednotlivých položek menu je názorně zobrazen na následujících obrázcích (C.3, C.4):

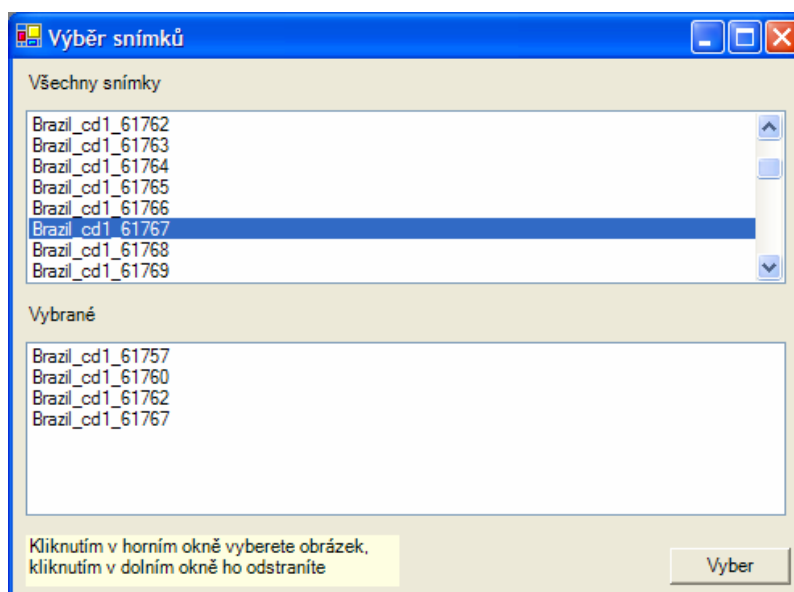


Obrázek C.3: Popis jednotlivých položek v nabídce „Soubor“.



Obrázek C.4: Popis jednotlivých položek v nabídce „Okna“.

Vybereme-li v nabídce Soubor možnost „Vytvoř sekvenci z otevřených souborů“ otevře se následující dialog, ve kterém si vybereme některé z otevřených souborů, které se nám spojí do sekvence:



Obrázek C.5: Výběr několika z otevřených obrázků.

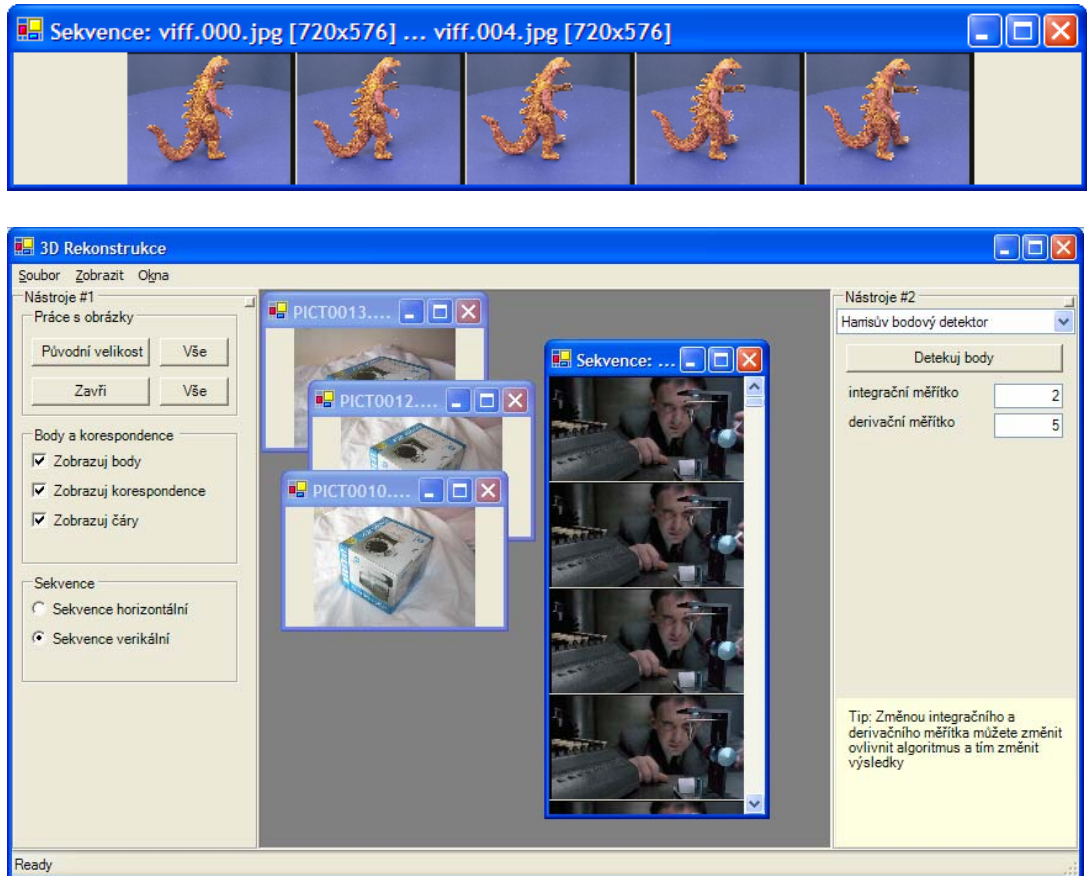
Vybereme-li v nabídce Soubor nabídku „Konverze (*.avi) na (*.bmp)“ otevře se standardní okno na otevírání dokumentů a po výběru souboru se zobrazí následující dialog:



Obrázek C.6: Převod z videa ve formátu avi na obrázky.

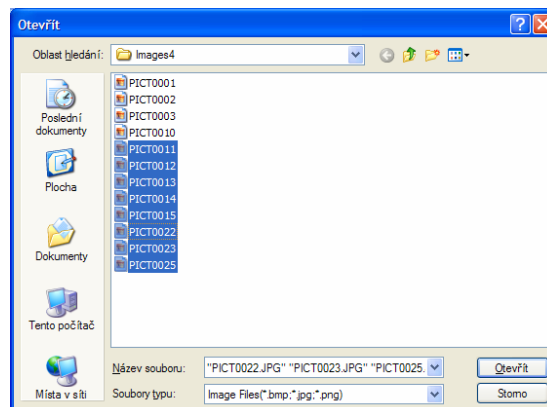
Již několikrát jsme se zmínili o sekvenci souborů, nyní si ukážeme co to vlastně je:

Sekvence souborů je několik snímků umístěných v jednom okně, zobrazení můžeme přepínat mezi vertikálním a horizontálním zobrazení. Sekvence obrázků ukážeme na obrázcích (C.7).



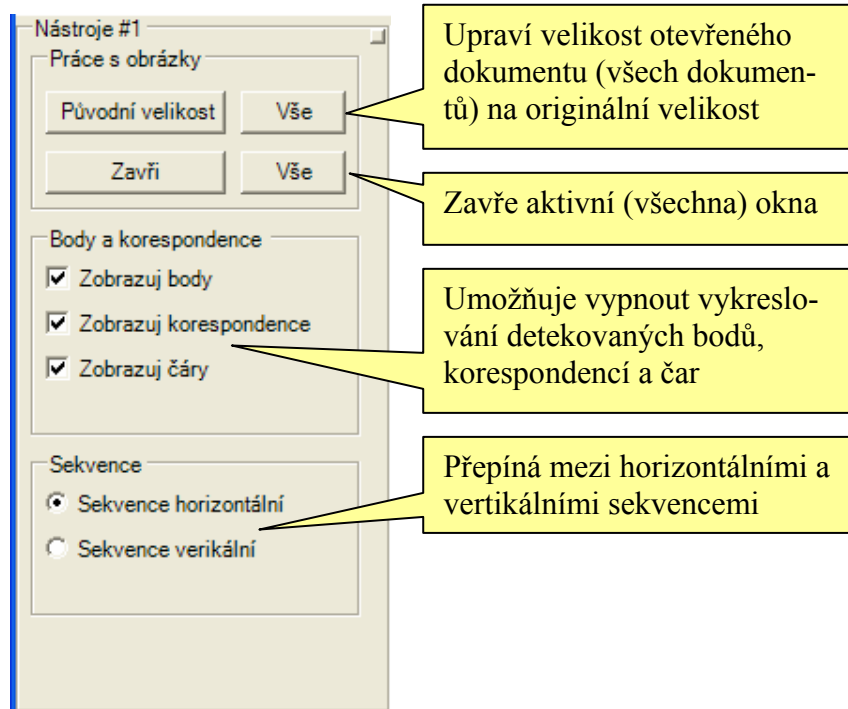
Obrázky C.7: Ukázka sekvencí obrázků.

Pro načítání sekvencí ze souborů je potřeba nejprve zvolit „Soubor→Otevřít sekvenci souborů“ v dialogu na otvírání souborů vybrat více souborů viz obrázek (C.8).



Obrázky C.7: Otevření sekvence souborů.

Popis panelu Nástroje #1



Obrázek C.9: Popis panelu nástrojů #1

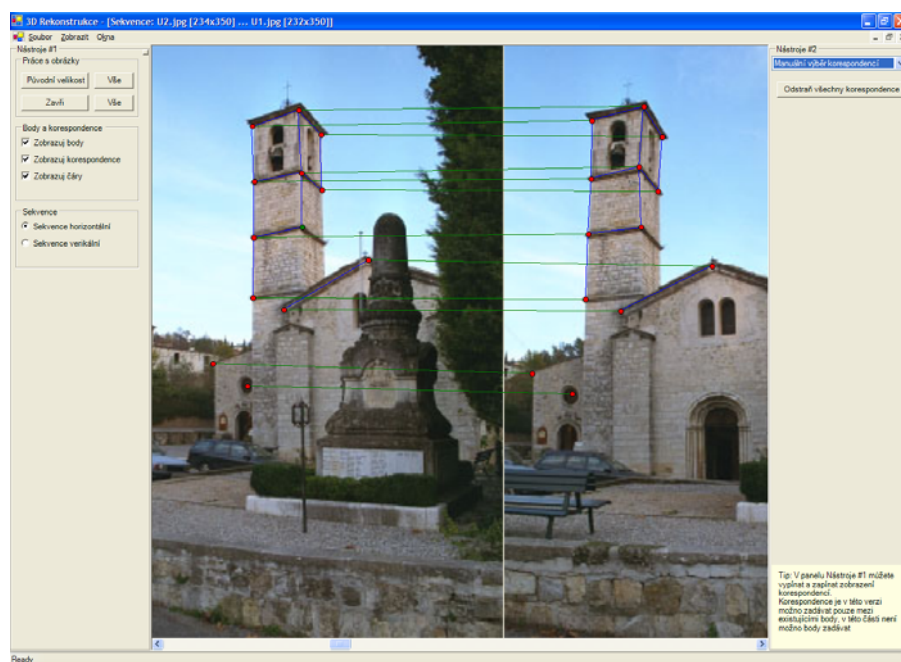
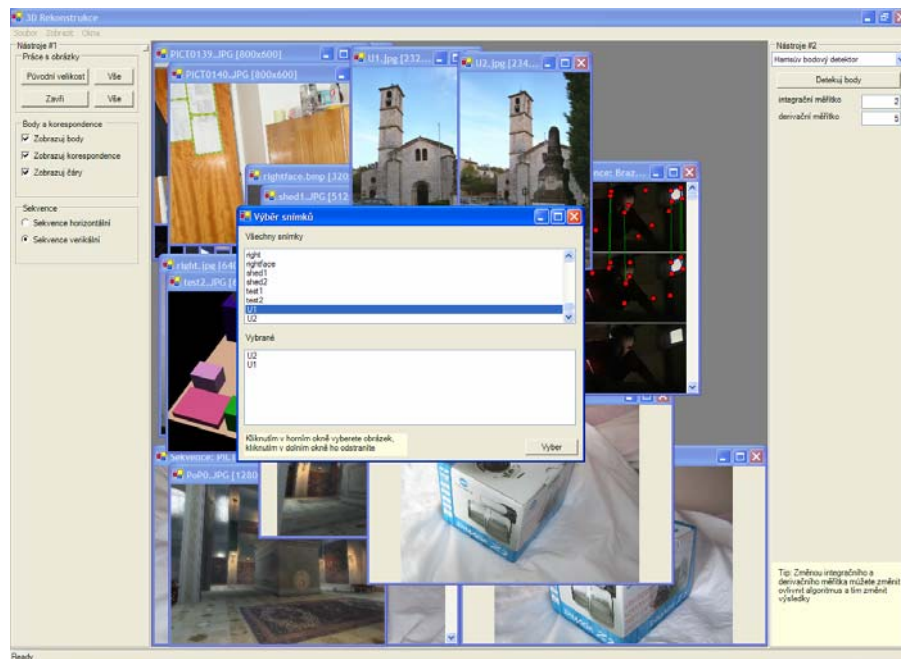
Panel Nástrojů #2

Obsah panelu nástrojů se mění podle toho, který z nástrojů je vybraný. Vybírat můžeme jeden z následujících nástrojů:

1. Harrisův bodový detektor
2. Manuální výběr korespondencí
3. Manuální zadávání/mazání bodů
4. Automatické hledání korespondencí MEM-Pairs
5. Automatické hledání korespondencí RANSAC
6. Projektivní rekonstrukce
7. Metrická rekonstrukce

Každý z těchto nástrojů má svá specifika a jejich ovládání je velmi intuitivní.

Na závěr ukážeme jak vypadá program v plném provozu, tedy se zadanými korespondencemi, detekovanými body, atd. .



Obrázek C.10: Ukázka aplikace za chodu.

Příloha D

Evidenční list

Souhlasím s tím, aby moje diplomová práce byla půjčována k prezenčnímu studiu v Univerzitní knihovně ZČU v Plzni.

V Plzni dne 19.5.2004,
Ladislav Lang

Uživatel stvrzuje svým čitelným podpisem, že tuto diplomovou práci použil ke studijním účelům a prohlašuje, že ji uvede mezi použitými prameny.

Jméno	Fakulta/Katedra	Datum	Podpis