



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

REKONSTRUKCE 3D OBJEKTŮ Z VÍCE POHLEDŮ

STRUCTURE FROM MOTION FROM MULTIPLE VIEWS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Daniel Mrkvička

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. Jana Procházková, Ph.D.

BRNO 2018

Zadání diplomové práce

Ústav: Ústav matematiky
Student: **Bc. Daniel Mrkvička**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **Mgr. Jana Procházková, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Rekonstrukce 3D objektů z více pohledů

Stručná charakteristika problematiky úkolu:

Rekonstrukce 3D objektů z fotografií je velmi zajímavá oblast zpracování obrazu. Je používána v mnoha aplikacích, jako navigace robotu, autonomní řízení a rozšířená realita. Jedná se o složení několika metod: nalezení klíčových bodů v každém obraze, nalezení pohybu kamery mezi dvěma záběry, spárování bodů a výsledná rekonstrukce.

Cíle diplomové práce:

Popis algoritmu rekonstrukce.

Výběr vhodných metod na základě různých vstupů. Naprogramování aplikace.

Seznam doporučené literatury:

SZELISKI, Richard. Computer vision: algorithms and applications. London: Springer, c2011. Texts in computer science. ISBN 978-1-84882-934-3.

WEINMANN, Martin. Reconstruction and Analysis of 3D Scenes. Switzerland: Springer, 2016.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato diplomová práce se zabývá rekonstrukcí scény užitím dvou nebo více obrazů. Popisuje celý proces rekonstrukce skládající se z detekce bodů v obrazech, nalezení příslušné geometrie mezi obrazy a výsledné promítnutí těchto bodů do prostoru scény. Práce dále zahrnuje i popis vlastní aplikace, která demonstruje popsání metody.

Summary

This thesis deals with the reconstruction of the scene using two or more images. It describes the whole reconstruction process consisting of detecting points in images, finding the appropriate geometry between images and resulting projection of these points into the space of scene. The thesis also includes a description of the application, which demonstrates the described methods.

Klíčová slova

Rekonstrukce scény, projektivní geometrie, fundamentální matice, trifokální tenzor, SURF

Keywords

Scene reconstruction, projective geometry, fundamental matrix, trifocal tensor, SURF

MRKVIČKA, D. *Rekonstrukce 3D objektů z více pohledů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2019. 74 s. Vedoucí Mgr. Jana Procházková, Ph.D.

Prohlašuji, že jsem diplomovou práci *Rekonstrukce 3D objektů z více pohledů* vypracoval samostatně pod vedením Mgr. Jany Procházkové, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Bc. Daniel Mrkvička

Chtěl bych poděkovat své vedoucí práce Mgr. Janě Procházkové, Ph.D. za odborné vedení během tvorby diplomové práce. Dále bych chtěl poděkovat svým rodičům za podporu během celého mého dosavadního studia.

Bc. Daniel Mrkvička

Obsah

Úvod	3
1 Projektivní geometrie a transformace	4
1.1 Euklidovský prostor	4
1.2 Rozšířený Euklidovský prostor	5
1.3 Projektivní rovina \mathbb{P}^2	6
1.3.1 Homogenní souřadnice projektivní roviny \mathbb{P}^2	6
1.3.2 Transformace projektivní roviny \mathbb{P}^2	7
1.4 Projektivní prostor \mathbb{P}^3	11
1.4.1 Homogenní souřadnice v projektivním prostoru \mathbb{P}^3	11
1.4.2 Transformace projektivního prostoru \mathbb{P}^3	13
2 Model Kamery	15
2.1 Popis základního modelu	15
2.1.1 Dírková komora	15
2.1.2 Digitální zařízení	16
2.1.3 Rotace a translace kamery	17
2.1.4 Perspektivní kamera	18
2.2 Radiální distorze	19
2.3 Kalibrace	20
2.3.1 Kalibrace s použitím kalibračního vzoru	20
3 Geometrie dvou pohledů	23
3.1 Epipolární geometrie	23
3.2 Fundamentální matice	24
3.2.1 Výpočet fundamentální matice	24
3.3 Esenciální matice	26
3.3.1 Určení matice kamery užitím esenciální matice	27
3.4 Rekonstrukce scény	28
3.5 Výpočet kamery z korespondencí mezi scénou a obrazem	29
4 Geometrie tří pohledů	30
4.1 Trifokální tenzor	30
4.1.1 Značení a symbolika	30
4.1.2 Vztahy mezi třemi pohledy	31
4.1.3 Výpočet epipolární geometrie z trifokálního tenzoru	32
4.1.4 Výpočet trifokálního tenzoru užitím matic kamer	33
4.2 Výpočet trifokálního tenzoru užitím Carlsson-Weinshallové duality	35
4.2.1 Algoritmus rekonstrukce	35
5 Detekce bodů metodou SURF	40
5.1 Matematický popis digitálního obrazu	40
5.2 Detekce významných bodů	41
5.3 Deskriptor	43

6	Podrobná rekonstrukce scény	45
6.1	Rektifikace obrazu	45
6.1.1	Transformace prvního obrazu	45
6.1.2	Transformace druhého obrazu	45
6.2	Disparita	46
6.2.1	Metoda SSD	47
7	Implementace popsaných metod	49
7.1	Popis aplikace	49
7.1.1	Rekonstrukce ze dvou obrazů	49
7.1.2	Implementace	52
7.1.3	Rekonstrukce z více obrazů	55
7.2	Testování výsledků na reálných datech	56
7.2.1	Rekonstrukce ze dvou obrazů	57
7.2.2	Rekonstrukce z více obrazů	61
7.2.3	Výsledné rekonstrukce	62
	Závěr	66
	Literatura	67
	Seznam použitých symbolů a veličin	70
A	Postup instalace knihovny mexopencv	71
B	Obsah příloženého CD	74

Úvod

S rozvojem digitální fotografie se otevírá řada možností, jak pracovat s informacemi uloženými ve fotografii. Jednou z mnoha možností je i rekonstrukce scény. Jedná se o proces, při kterém dochází k výpočtu bodů trojrozměrného prostoru na základě fotografií. Jedná se tedy o převod dvojrozměrných informací na informace trojrozměrné.

Rekonstrukci scény lze využít například při vytváření počítačových modelů historických památek a cenných předmětů. Dalším polem využití je například rozšířená realita. Jedná se o doplnění reálného obrazu o předměty vytvořené počítačem. S rozšířenou realitou se stále častěji setkáváme v mobilních hrách, neboť mobilní telefony dnes disponují vestavěnou kamerou. Rekonstrukci scény lze dále využít například při řízení robotů a velmi specifickým případem rekonstrukce je i výpočetní tomografie využívaná v lékařství.

Vzhledem k široké řadě využití se v případě rekonstrukce scény jedná o oblast, která je v dnešní době stále zkoumána a rychle rozvíjena. Již bylo popsáno mnoho algoritmů, které umožňují jednotlivé kroky celého procesu. Jednotlivé algoritmy se kombinují a cílem je získat rychlý proces, který umožní rekonstrukci s dostatečnou přesností. Podrobná rekonstrukce scény je totiž stále výpočetně náročná na čas i paměť počítače. Rekonstrukce scény je tedy vhodnou oblastí pro další výzkum.

Tato diplomová práce má za cíl přiblížit jednotlivé kroky procesu rekonstrukce scény od analýzy jednotlivých fotografií až po bodovou rekonstrukci a vybrané algoritmy implementovat do funkční aplikace.

Práce je rozdělena do sedmi kapitol. Kapitola 1 popisuje výstavbu rozšířeného Euklidovského prostoru. V její druhé části je obsažen popis a transformace projektivních prostorů.

Kapitola 2 obsahuje popis modelu kamery. Model kamery je zde nejprve představen v nejjednodušší formě jako dírková komora a následně je rozšířen na model, který popisuje moderní digitální fotoaparáty. Tento model zahrnuje matici kamery, kalibrační matici a koeficienty radiální či tangenciální distorze. Na závěr kapitoly je popsán proces kalibrace, tedy způsob zjištění parametrů kamery nutných pro správnou rekonstrukci.

Kapitola 3 se zabývá geometrií dvou pohledů, tzv. epipolární geometrií. V rámci epipolární geometrie je představena fundamentální matice a způsoby jejího výpočtu. Kapitola popisuje převod fundamentální matice na matici esenciální a využití této matice k výpočtu matic kamer. V další části je popsána samotná rekonstrukce scény.

Kapitola 4 obsahuje popis geometrie tří pohledů. Kapitola zahrnuje definici trifokálního tenzoru a nastínění postupu pro jeho výpočet. Dále kapitola vysvětluje některé závislosti mezi geometrií dvou pohledů a geometrií tří pohledů.

Kapitola 5 zahrnuje popis metody SURF. Ta je použita pro detekci významných bodů v jednotlivých obrazech. Kapitola se věnuje tomu, jakým způsobem SURF tyto body nalézá, a jak jsou body ze dvou obrazů mezi sebou porovnávány.

Kapitola 6 rozebírá rektifikaci obrazu, tedy jakým způsobem je transformována dvojice obrazů. Dále popisuje výpočet disparity mezi obrazy, která umožňuje z této dvojice získat více bodů pro rekonstrukci scény.

Kapitola 7 se zabývá popisem aplikace, která implementuje vybrané metody. První část zahrnuje popis prostředí a vysvětlení jednotlivých kroků vnitřního algoritmu. V druhé části jsou pak uvedeny výstupy této aplikace. Závěr obsahuje shrnutí celé práce a zhodnocení dosažených výsledků.

1. Projektivní geometrie a transformace

V této kapitole nejdříve zavedeme Euklidovské prostory, které následně použijeme pro uvedení rozšířených Euklidovských prostorů. Budou uvedeny a blíže rozebrány dva případy – rozšířená Euklidovská rovina a rozšířený Euklidovský prostor. Jedná se o speciální případy projektivních prostorů. Pro oba uvedené případy bude rozebráno zavedení speciálního typu souřadnic, tzv. homogenních souřadnic, a jak se využijí při definici vztahů mezi jednotlivými geometrickými objekty. Dále budou představeny základní typy projektivních zobrazení a některé jejich vybrané vlastnosti.

Text této kapitoly předpokládá, že čtenář má základní znalosti z oblasti lineární algebry a počítačové grafiky.

Pokud není uvedeno jinak tato kapitola čerpá z [11],[14], [18] a [19].

1.1. Euklidovský prostor

Euklidovský prostor je pojmenován po Euklidovi, řeckém matematikovi a geometrovi z 3. století před naším letopočtem. Eukleides ve svém díle Základy[9] popisuje mimo jiné rovinnou a prostorovou geometrii pomocí axiomů a postulátů. Daný systém se později ukázal jako nedůsledný a s logickými nedostatky.

Euklidovský prostor popisuje naši běžnou představu o dvojrozměrné rovině i trojrozměrném prostoru. Lze zobecnit i pro prostory s vyšší dimenzí, ale tyto prostory nejsou využity v této práci.

Z hlediska geometrie lze Euklidovský prostor \mathbb{E}^2 definovat pomocí axiomů, které ke konci 19. století zavedl německý matematik David Hilbert. Jedná se o tři axiomy incidence, čtyři axiomy uspořádání, šest axiomů shodnosti, dva axiomy spojitosti a axiom rovnoběžnosti. Pro zavedení euklidovského prostoru \mathbb{E}^3 je nutné zavedení dalších šesti axiomů incidence, které přidávají vztahy roviny k bodům, přímkám a jiným rovinám. Zde z nich uvedeme pouze jeden, který je nutné pozměnit pro definování složitějších prostorů. Ostatní axiomy lze nalézt například ve skriptech Počítačová geometrie a grafika[18].

Axiom 1.1.1 (Euklidův). Bodem A neležícím na přímce p prochází právě jedna přímka q , která s přímkou p nemá společný žádný bod.

Dále lze Euklidovský prostor popsat také algebraicky. Následující definice mimo jiné předpokládá znalost definice vektorového prostoru.

Definice 1.1.2. Nechť V je vektorový prostor. Buď $\langle \cdot, \cdot \rangle$ zobrazení množiny $V \times V$ do množiny reálných čísel \mathbb{R} splňující následující podmínky:

- (1) $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{v}, \mathbf{u} \rangle$ pro každé $\mathbf{u}, \mathbf{v} \in V$,
- (2) $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$ pro všechna $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$,
- (3) $\langle r\mathbf{u}, \mathbf{v} \rangle = r\langle \mathbf{u}, \mathbf{v} \rangle$ pro všechna $\mathbf{u}, \mathbf{v} \in V, r \in \mathbb{R}$,
- (4) $\langle \mathbf{u}, \mathbf{u} \rangle > 0$ pro každé $\mathbf{u} \in V, \mathbf{u} \neq \mathbf{0}$, kde $\mathbf{0}$ je nulový vektor.

Pak řekneme, že na vektorovém prostoru V je definován skalární součin vektorů a vektorový prostor V se nazývá vektorový prostor se skalárním (vnitřním) součinem nebo **euklidovský prostor**. Reálné číslo $\langle \mathbf{u}, \mathbf{v} \rangle$ se nazývá skalární součin vektorů $\mathbf{u}, \mathbf{v} \in V$.

Užitím algebry získáváme analytický model Euklidovského prostoru. Tento model nám umožňuje řešit řadu úloh užitím lineárních rovnic.

Další vlastnosti Euklidova prostoru je možné najít například ve skriptech Lineární algebra: teoretická část[14].

1.2. Rozšířený Euklidovský prostor

Nevýhodou Euklidova prostoru je složité vyjádření některých zobrazení $\mathbb{E}^2 \rightarrow \mathbb{E}^2$, případně $\mathbb{E}^3 \rightarrow \mathbb{E}^3$, které představují základní geometrické operace jako je například osová souměrnost podle obecné přímky v \mathbb{E}^2 . V Euklidovském prostoru je totiž translace uskutečněna nikoli pomocí vynásobení maticí, ale přičtením vektoru. Proto je nutné zavedení rozšířených Euklidovských prostorů \mathbb{E}_∞^2 a \mathbb{E}_∞^3 . V nich je každá transformace vyjádřitelná pouze pomocí součinu různého počtu čtvercových matic. Můžeme tedy libovolnou transformaci vyjádřit jedinou maticí [19].

K zavedení těchto prostorů je nutné nejprve rozšířit stávající systém axiomů Euklidova prostoru o nový axiom.

Axiom 1.2.1 (Projektivní). Každé dvě přímky, které leží v téže rovině, mají společný bod.

Tento axiom je ovšem v rozporu s Euklidovým axiomem 1.1.1. Pro vyřešení tohoto rozporu nazveme všechny body z Euklidovy roviny *vlastními body* a Euklidův axiom nahradíme následujícím axiomem

Axiom 1.2.2. Bodem A neležícím na přímce p prochází právě jedna přímka q , která s přímkou p nemá společný žádný vlastní bod.

Přímky p a q z axiomu 1.2.2 nemají žádný společný vlastní bod. Podle axiomu 1.2.1 však nějaký společný bod mít musí. Tento bod nazveme *nevlátní bod*.

Rovinu a trojrozměrný prostor, kde platí axiom 1.2.1 a axiom 1.1.1 je nahrazen axiomem 1.2.2, nazveme *projektivní rovinou* resp. *projektivním prostorem*. V této práci budeme dále místo značení \mathbb{E}_∞^2 a \mathbb{E}_∞^3 používat značení \mathbb{P}_2 pro projektivní rovinu a označení \mathbb{P}_3 pro trojrozměrný projektivní prostor. Toto značení vychází obecnějšího označování projektivních prostorů.

Podobně jako Euklidův prostor i obecný projektivní prostor lze definovat pomocí algebraických struktur.

Definice 1.2.3. Nechť P je množina, V_{n+1} je vektorový prostor nad tělesem T dimenze $n+1$ a bijekce $\varphi : \{\mathbf{u} \subseteq V_{n+1}, \mathbf{u} \neq \mathbf{0}\} \rightarrow P$. Pak trojici $\mathbb{P}_n = (P, V_{n+1}, \varphi)$ nazýváme n -rozměrný projektivní prostor nad T . V_{n+1} se nazývá aritmetický základ prostoru \mathbb{P}_n . P se nazývá nosič nebo množina bodů prostoru \mathbb{P}_n .

Například projektivní rovinu \mathbb{P}_2 v souladu s touto definicí zavedeme následovně [1]:

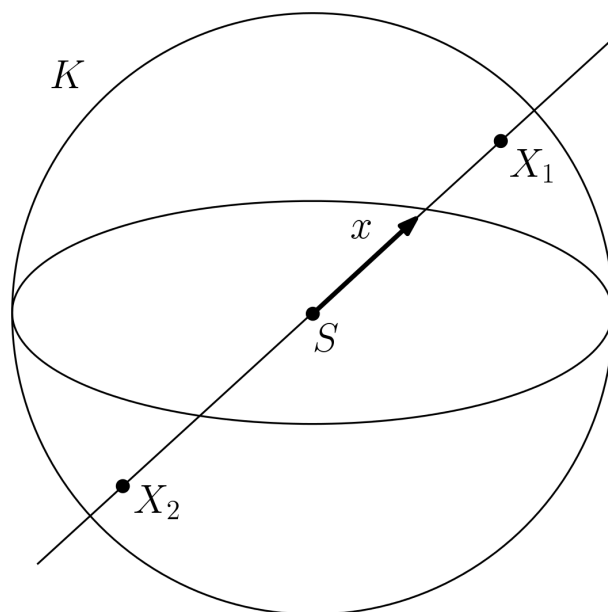
$P : \{\{X_1, X_2\} \subset K, S \in (X_1, X_2)\}$, kde K je jednotková kulová plocha se středem S

aritmetickým základem je V_3

$\varphi : \{[\mathbf{x}] \subseteq V_3, \mathbf{x} \neq \mathbf{0}\} \rightarrow P, [\mathbf{x}] \rightarrow \{S, \mathbf{x}\} \cap K$

$\mathbb{P}_2 = (P, V_3, \varphi)$

Zavedení tohoto prostoru je znázorněno na obr. 1.1.



Obrázek 1.1: Projektivní rovina

1.3. Projektivní rovina \mathbb{P}^2

Po zavedení projektivní roviny \mathbb{P}^2 v předešlé kapitole je nyní možné definovat vybrané operace s body a přímkami. Nejprve je ovšem nutné zavést pro oba útvary speciální typ souřadnic.

1.3.1. Homogenní souřadnice projektivní roviny \mathbb{P}^2

Homogenní souřadnice bodu

V projektivním prostoru obecně je nutné rozlišovat mezi vlastními a nevlastními body. To je umožněno pomocí zavedení *homogenních souřadnic*. V homogenních souřadnicích je každý bod projektivní roviny \mathbb{P}_2 reprezentován vektorem dimenze 3. Vlastní bod, který má v Euklidovské rovině souřadnice (x, y) , bude tedy v projektivním prostoru vyjádřen jako vektor $\mathbf{x} = (kx, ky, k)^\top$, kde $k \neq 0$. Pokud $k = 1$, pak o vektoru \mathbf{x} mluvíme jako o *normalizovaných homogenních souřadnicích* bodu \mathbf{x} .

Lze vidět, že bodu, který má souřadnice v euklidovské rovině (x^E, y^E) , odpovídají každé homogenní souřadnice tvaru (x, y, k) pokud platí

$$x^E = \frac{x}{k}, \quad y^E = \frac{y}{k}. \quad (1.1)$$

Tedy každý vlastní bod má nekonečně mnoho homogenních souřadnic.

Nevlastní bod projektivní roviny má homogenní souřadnice tvaru $\mathbf{x} = (x, y, 0)$. Zatímco vlastní body chápeme jako body Euklidovské roviny, nevlastní body lze chápat jako směrové vektory.

Homogenní souřadnice přímky

Stejně jako body můžeme v homogenních souřadnicích vyjádřit i přímky. Přímkou v Euklidovské rovině lze vyjádřit *obecnou rovnicí přímky* tvaru

$$ax + by + c = 0. \quad (1.2)$$

Volbou koeficientů a , b a c lze jednoznačně určit každou rovinnou přímku. Proto můžeme každou přímku vyjádřit jako vektor $\mathbf{l} = (a, b, c)^\top$. Každé přímce vyjádřené rovnicí $ax + by + c = 0$ opět odpovídá více homogenních souřadnic tvaru (ka, kb, kc) , kde $k \neq 0$.

Vztahy bodů a přímek

Pomocí homogenních souřadnic lze vyjádřit, že bod $\mathbf{x} = (x, y, 1)^\top$ leží na přímce $\mathbf{l} = (a, b, c)$ právě tehdy když

$$\mathbf{x}^\top \mathbf{l} = 0. \quad (1.3)$$

V projektivní rovině je mezi bodem a přímkou speciální vztah nazývaný *princip duality*. Ten nám umožňuje v tvrzeních nahradit slovo *bod* slovem *přímka* a obráceně, přičemž je důležité správně zachovat popisovaný vztah. Například místo věty: "Dvěma body prochází právě jedna přímka." tak dostaneme větu: "Dvě přímky se protínají právě v jednom bodě." Princip duality mezi dvěma geometrickými útvary se projeví i v rovnicích. Proto vedle rovnice (1.3) platí také rovnice

$$\mathbf{l}^\top \mathbf{x} = 0. \quad (1.4)$$

Naopak pomocí vektorového součinu je možné najít průnik dvou přímek $\mathbf{l} = (a, b, c)^\top$ a $\mathbf{l}' = (a', b', c')^\top$. Tyto přímky se protínají v bodě

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}'. \quad (1.5)$$

Využitím principu duality lze použitím předchozí rovnice snadno sestavit rovnici pro nalezení přímky \mathbf{l} procházející body $\mathbf{x} = (x, y, 1)^\top$ a $\mathbf{x}' = (x', y', 1)^\top$

$$\mathbf{l} = \mathbf{x} \times \mathbf{x}'. \quad (1.6)$$

Uvažujme nyní dvě rovnoběžné přímky $\mathbf{l} = (a, b, c)^\top$ a $\mathbf{l}' = (a, b, c')^\top$. Užitím vztahu (1.4) dostáváme, že tyto přímky se protnou v bodě $\mathbf{x} = (b, -a, 0)^\top$. Dosazením do vztahu (1.1) získáme vyjádření tohoto bodu v Euklidově rovině $\lim_{x \rightarrow \infty} (\frac{b}{x}, \frac{-a}{x})$, což odpovídá naší představě, že dvě rovnoběžky se protínají v nekonečnu.

Dále uvažujme dva nevlastní body $\mathbf{x} = (a, b, 0)^\top$ a $\mathbf{x}' = (c, d, 0)^\top$. Užitím vztahu (1.4) lze odvodit, že oba tyto body leží na přímce $\mathbf{l} = (0, 0, ad - bc)^\top$, která po znormování odpovídá tzv. *nevlastní přímce* $\mathbf{l}_\infty = (0, 0, 1)^\top$. Na této přímce leží všechny nevlastní body projektivní roviny.

1.3.2. Transformace projektivní roviny \mathbb{P}^2

Transformací rozumíme vzájemně jednoznačné zobrazení. Významnou skupinou transformací projektivní roviny jsou tzv. *kolineární zobrazení*. Tyto zobrazení zobrazí přímku opět jako přímku. Častěji se toto zobrazení nazývá také jako *projektivní zobrazení* nebo *homografie*.

Definice 1.3.1. *Kolineární zobrazení* je invertibilní zobrazení $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$, takové, že body x_1, x_2 a x_3 leží na stejné přímce tehdy a jen tehdy, když body $h(x_1), h(x_2)$ a $h(x_3)$ leží na stejné přímce.

Inverzní zobrazení kolineárního zobrazení je opět kolineární zobrazení. Složení dvou kolineárních zobrazení je také kolineární zobrazení. Je zřejmé, že kolineární zobrazení tvoří grupu.

Každé kolineární zobrazení mezi dvěma projektivními rovinami je možné vyjádřit pomocí matice.

Věta 1.3.2. *Zobrazení $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ je kolineární právě tehdy když existuje regulární matice 3×3 H taková, že pro každý bod roviny \mathbb{P}^2 , který nechť je reprezentován vektorem \mathbf{x} , platí $h(\mathbf{x}) = H\mathbf{x}$.*

Díky této větě můžeme nyní definovat kolineární zobrazení pomocí maticového zápisu.

Definice 1.3.3. Rovinná kolineární transformace je lineární transformací homogenního vektoru 3×1 reprezentovaná regulární maticí 3×3 :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1.7)$$

případně $\mathbf{x}' = H\mathbf{x}$.

Matice H z rovnice (1.7) může být vynásobena libovolným nenulovým skalárem a stále bude představovat stejnou transformaci. Proto má u ní význam pouze poměr mezi jejími prvky (podobně jako u homogenních souřadnic) a projektivní transformace má tudíž 8 stupňů volnosti (tj. počet prvků, které jsou navzájem nezávislé). Matici projektivní transformace lze určit z homogenních souřadnic čtyř bodů a jejich zobrazení.

Takto popsané projektivní zobrazení je nejobecnější případ transformace mezi dvěma projektivními rovinami. Existují speciálnější případy, které jsou zároveň podgrupami grupy projektivních transformací. Tyto případy nyní popíšeme podrobněji.

Izometrická transformace

První případem je izometrická transformace. Toto zobrazení zachovává Euklidovu metriku. Izometrická transformace je vyjádřena jako

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos \theta & -\sin \theta & t_x \\ \epsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1.8)$$

kde $\epsilon = \pm 1$. Pokud $\epsilon = 1$, pak transformace zachovává orientaci a nazývá se *Euklidovou transformací*. Euklidova transformace se skládá z rotace a translace. V případě, že $\epsilon = -1$, má transformace opačnou orientaci.

Rovinnou Euklidovu transformaci lze také vyjádřit v blokovém zápisu

$$\mathbf{x}' = H_E \cdot \mathbf{x} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}, \quad (1.9)$$

kde R je rovinná rotační matice 2×2 , \mathbf{t} je vektor translace a $\mathbf{0}^\top = (0, 0)$.

Rovinná Euklidovská transformace má tři stupně volnosti, jeden stupeň volnosti pro rotaci a dva stupně volnosti pro translaci. To umožňuje určit transformační matici již ze dvou bodů a jejich zobrazení.

Invarianty této transformace jsou například délka úsečky, úhel dvou přímek a rovinný obsah trojúhelníku.

Podobnostní transformace

Tato transformace stejně jako izometrická transformace zachovává úhly, ale zároveň umožňuje změnu měřítka, přičemž mění rozměr objektů ve všech směrech stejně. Podobnostní transformace je vyjádřena

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1.10)$$

což lze vyjádřit v blokovém zápisu

$$\mathbf{x}' = H_S \cdot \mathbf{x} = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}, \quad (1.11)$$

kde skalár s vyjadřuje *měřítkový faktor*. Rovinná podobnostní transformace má čtyři stupně volnosti. Tři jsou stejné jako u rovinné izometrické transformace a čtvrtým stupněm volnosti je měřítkový faktor.

Invariantním vůči transformaci je poměr délek a úhly.

Afinní transformace

Afinní transformace je složením lineární transformace a translace. V maticovém zápisu je vyjádřena

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (1.12)$$

nebo v blokovém zápisu jako

$$\mathbf{x}' = H_A \cdot \mathbf{x} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x}, \quad (1.13)$$

kde A je regulární matice 2×2 . Afinní transformace má již šest stupňů volnosti a lze ji jednoznačně určit ze tří bodů a jejich zobrazení.

Matici A lze rozložit na součin matic rovinné rotace a matice anizotropní změny měřítka (změna rozměru je v různých směrech odlišná)

$$A = R(\theta)R(-\phi)DR(\phi), \quad (1.14)$$

kde $R(\theta)$ a $R(\phi)$ jsou matice rovinné rotace o úhel θ , respektive ϕ a D je diagonální matice

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

Úhel θ vyjadřuje úhel rotace podobně jako u izometrické transformace. Úhel ϕ a poměr $\lambda_1 : \lambda_2$ určují hlavní směr a velikost deformace.

Invariantem afinní transformace je obsah. Afinní transformace dále zachovává rovnoběžnost tak, že dvě libovolné přímky, které jsou rovnoběžné, zůstávají rovnoběžné i po transformaci. Taktéž zůstává stejný i poměr délek navzájem rovnoběžných úseček.

Projektivní transformace

Tvar matice obecné projektivní transformace již byla zmíněna ve vzorci (1.7). Uvedme ještě její blokový zápis

$$\mathbf{x}' = H_P \cdot \mathbf{x} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix} \mathbf{x}, \quad (1.15)$$

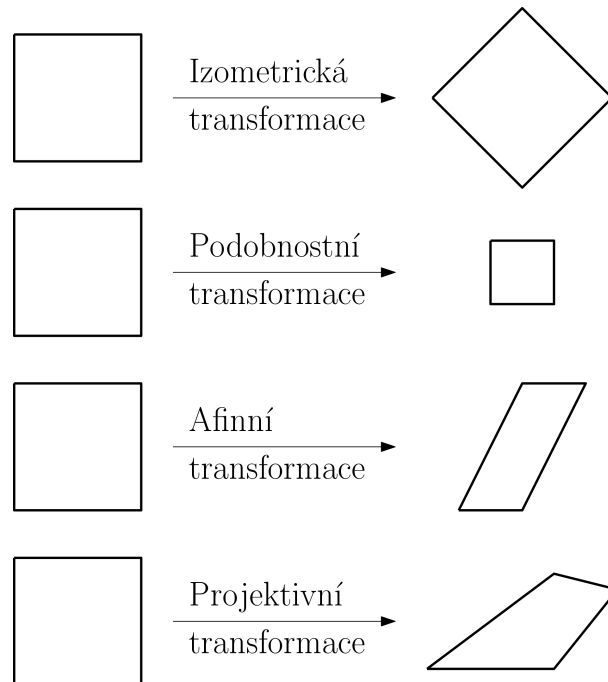
kde vektor $\mathbf{v}^\top = (v_1, v_2)^\top$ a matice A je stejná jako u afinní transformace. Matici projektivní transformace je možné určit ze čtyř bodů a jejich zobrazení pokud žádné tři body nebo jejich zobrazení nejsou kolineární.

Jediným invariantem projektivní transformace je *dvojpoměr* čtyř bodů. Dvojpoměr bodů $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ a \mathbf{x}_4 , které leží na jedné přímce, je definován jako číslo

$$\delta(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{|\mathbf{x}_1\mathbf{x}_2| |\mathbf{x}_3\mathbf{x}_4|}{|\mathbf{x}_1\mathbf{x}_3| |\mathbf{x}_2\mathbf{x}_4|}, \quad (1.16)$$

kde $|\mathbf{x}_i\mathbf{x}_j|$ je délka úsečky $\mathbf{x}_i\mathbf{x}_j$.

Příklady všech typů projektivní transformace jsou ukázány na obr.1.2.



Obrázek 1.2: Příklady projektivních transformací čtverce

1.4. Projektivní prostor \mathbb{P}^3

Jak již bylo zmíněno, projektivní prostor \mathbb{P}^3 je podobně jako projektivní rovina \mathbb{P}^2 získán rozšířením Euklidovského prostoru (resp. roviny). Proto má mnoho podobných vlastností a k jejich ukázení je opět nutné nejprve zavést homogenní souřadnice.

1.4.1. Homogenní souřadnice v projektivním prostoru \mathbb{P}^3

V projektivním prostoru \mathbb{P}^3 se zavádí homogenní souřadnice bodu a roviny. Homogenní souřadnice přímky není možné jednoduše zavést. Některé způsoby zavedení souřadnic i pro přímku zde budou také uvedeny.

Homogenní souřadnice bodu

Znovu je nutné rozlišovat mezi vlastními a nevlastními body. Vlastní bod s kartézskými souřadnicemi (X, Y, Z) má homogenní souřadnice tvaru $\mathbf{X} = (kX, kY, kZ, k)^\top$, kde $k \neq 0$. Každý bod má nekonečně mnoho homogenních souřadnic a vektor tvaru $\mathbf{X} = (X, Y, Z, 1)^\top$ se nazývá normalizovaný homogenní vektor. Nevlastní bod má homogenní souřadnice $(kX, kY, kZ, 0)^\top$.

Homogenní souřadnice roviny

Obecná rovnice roviny v Euklidově prostoru \mathbb{E}^3 je

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0. \quad (1.17)$$

Podobně jako v případě obecné rovnice přímky (1.2) i zde je důležitý pouze poměr mezi jednotlivými koeficienty π_i a vynásobení celé rovnice nenulovou konstantou nemá vliv na určení roviny. Homogenní souřadnice roviny, která je dána rovnicí (1.17), jsou vyjádřeny čtyřrozměrným vektorem $(\pi_1, \pi_2, \pi_3, \pi_4)$.

Homogenní souřadnice přímky

Přímka je v prostoru \mathbb{P}^3 určena buď dvěma různými body, které na ni leží, nebo jako průsečnice dvou rovin. Přímka má v tomto případě čtyři stupně volnosti. Proto by měla být vyjádřitelná pěti-rozměrným vektorem. Bod i rovina jsou ale v prostoru \mathbb{P}^3 vyjádřeny pomocí čtyřrozměrných homogenních vektorů. Rozdíl v dimenzi vektorů by znesnadnil většinu prováděných operací. To je důvodem, proč se hledají jiné způsoby vyjádření přímky.

Jedním z možných způsobů je užitím jádra lineárního zobrazení. Necht' \mathbf{A} a \mathbf{B} jsou dva různé body. Pak přímka procházející těmito body je dána jako lineární obal matice W , která má rozměry 2×4

$$W = \begin{bmatrix} \mathbf{A}^\top \\ \mathbf{B}^\top \end{bmatrix} \quad (1.18)$$

Z tohoto vyplývá, že libovolný bod \mathbf{C} na této přímce lze vyjádřit jako $\mathbf{C} = \lambda \mathbf{A} + \mu \mathbf{B}$, kde $\lambda, \mu \in \mathbb{R}$.

Přímku lze vyjádřit velmi podobným způsobem při použití rovin. Nechť \mathbf{P} a \mathbf{Q} jsou dvě roviny, pak přímka, jež je průsečnicí těchto rovin, je dána jako lineární obal matice W^* , která má opět rozměry 2×4

$$W^* = \begin{bmatrix} \mathbf{P}^\top \\ \mathbf{Q}^\top \end{bmatrix}. \quad (1.19)$$

V tomto případě rovnice $\mathbf{R} = \lambda' \mathbf{P} + \mu' \mathbf{Q}$, kde $\lambda', \mu' \in \mathbb{R}$ představuje svazek všech rovin, ve kterých leží daná přímka.

Dalším způsobem jak vyjádřit přímku v prostoru \mathbb{P}^3 je například užitím tzv. *Plückerových souřadnic*. Tento způsob zde však uvádět nebudeme, ale lze jej nalézt například v knize *Multiple view geometry in computer vision*[11].

Vztahy bodů, přímek a rovin

V prostoru \mathbb{P}^3 je duální bod a rovina. Princip duality platí taktéž pro dvě přímky navzájem.

Skutečnost, že bod $\mathbf{X} = (X, Y, Z, 1)^\top$ leží v rovině $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^\top$ je vyjádřena rovnicí

$$\boldsymbol{\pi}^\top \mathbf{X} = 0. \quad (1.20)$$

Uvažujme tři body $\mathbf{X}_1, \mathbf{X}_2$ a \mathbf{X}_3 , které neleží na stejné přímce. Každý bod \mathbf{X}_i a rovinu $\boldsymbol{\pi}$, ve které všechny leží, splňují rovnici (1.20). Proto také musí platit rovnice

$$\begin{bmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \\ \mathbf{X}_3^\top \end{bmatrix} \boldsymbol{\pi} = \mathbf{0}. \quad (1.21)$$

Máme-li tedy dány tři body, které nejsou kolineární, pak rovinu $\boldsymbol{\pi}$ obdržíme jako jádro lineárního zobrazení matice $[\mathbf{X}_1 \ \mathbf{X}_2 \ \mathbf{X}_3]^\top$. Pokud by body byly kolineární, pak jádro je dimenze dva a vyjadřuje množinu rovin, které mají společnou přímku, na které dané body leží.

Z duality bodu a roviny vyplývá, že podobným způsobem lze určit bod X , který je průsečíkem tří rovin $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ a $\boldsymbol{\pi}_3$. Vše vyplývá ze vztahu

$$\begin{bmatrix} \boldsymbol{\pi}_1^\top \\ \boldsymbol{\pi}_2^\top \\ \boldsymbol{\pi}_3^\top \end{bmatrix} \mathbf{X} = \mathbf{0}. \quad (1.22)$$

Pokud roviny mají společný pouze jeden bod, pak jej obdržíme jako jádro lineárního zobrazení matice $[\boldsymbol{\pi}_1 \ \boldsymbol{\pi}_2 \ \boldsymbol{\pi}_3]^\top$. Pokud mají roviny společnou přímku, pak je jádro lineárního zobrazení dvourozměrné a jeho lineárním obalem je množina všech bodů ležících na této přímce.

Rovina, která je dána přímkou W (viz vztah (1.18)) a bodem \mathbf{X} , je získána jako jádro lineárního zobrazení matice

$$M = \begin{bmatrix} W \\ \mathbf{X}^\top \end{bmatrix}. \quad (1.23)$$

Pokud je jádro dimenze dva, pak bod leží na přímce.

Pokud je naopak bod dán jako průsečík roviny $\boldsymbol{\pi}$ a přímky W^* (viz vztah (1.19)), bod je určen jako jádro lineárního zobrazení matice

$$M = \begin{bmatrix} W^* \\ \boldsymbol{\pi}^\top \end{bmatrix}. \quad (1.24)$$

Pokud je jádro dimenze dva, pak přímka leží v rovině.

Podobně jako v prostoru \mathbb{P}^2 existuje nevlastní přímka l_∞ , v prostoru \mathbb{P}^3 existuje nevlastní rovina $\pi_\infty = (0, 0, 0, 1)^\top$. Tato rovina obsahuje všechny nevlastní body.

Dvě roviny jsou rovnoběžné právě tehdy, když jejich průsečnice leží v rovině π_∞ . Dvě přímky jsou rovnoběžné právě tehdy, když jejich průsečíkem je nevlastní bod.

1.4.2. Transformace projektivního prostoru \mathbb{P}^3

Projektivní transformace prostoru \mathbb{P}^3 přejímá všechny vlastnosti projektivní transformace prostoru \mathbb{P}^2 . V prostoru \mathbb{P}^3 je transformace bodu \mathbf{X} na bod \mathbf{X}' zcela obecně dána vztahem

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (1.25)$$

případně užitím maticového zápisu $\mathbf{X}' = H\mathbf{X}$.

Matici H lze přehledně zapsat v blokovém zápisu

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix}, \quad (1.26)$$

kde A je regulární matice 3×3 , \mathbf{t} je trojrozměrný vektor translace, \mathbf{v} je obecný trojrozměrný vektor a v je nenulová konstanta. Matice H má patnáct stupňů volnosti. Následující speciální případy projektivní transformace mají všechny vlastnosti odpovídajících transformací uvedených v kapitole 1.3.2.

Euklidovská transformace

Euklidovská transformace je nejjednodušším typem zde rozebírané transformace v prostoru \mathbb{P}^3 . Matice H je v tomto případě

$$H = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (1.27)$$

kde R je rotační matice 3×3 , \mathbf{t} je trojrozměrný vektor translace a $\mathbf{0}^\top = (0, 0, 0)$ je nulový vektor.

Euklidovská transformace má celkem šest stupňů volnosti a je jednoznačně určena třemi body neležícími na jedné přímce a jejich zobrazeními.

Podobnostní transformace

Podobnostní transformace umožňuje oproti transformaci Euklidovské změnu měřítka. Matice H je v tomto případě

$$H = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (1.28)$$

kde R je rotační matice 3×3 , skalár s vyjadřuje globální změnu měřítka, \mathbf{t} je trojrozměrný vektor translace a $\mathbf{0}^\top = (0, 0, 0)$ je nulový vektor.

Tato transformace má sedm stupňů volnosti a je stejně jako Euklidovská transformace určena třemi body a jejich zobrazeními.

Afinní transformace

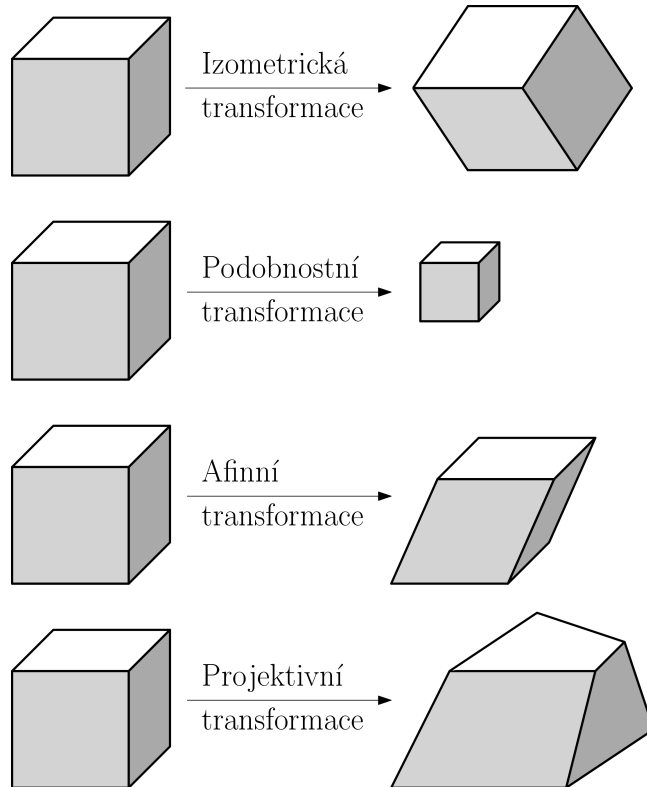
Afinní transformace umožňuje různou změnu měřítka v různých směrech. Matice H je v tomto případě

$$H = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (1.29)$$

kde R je regulární 3×3 , \mathbf{t} je trojrozměrný vektor translace a $\mathbf{0}^\top = (0, 0, 0)$ je nulový vektor.

Afinní transformace má 12 stupňů volnosti a lze ji určit ze čtyř bodů a jejich zobrazení, přičemž žádná trojice bodů nesmí ležet na jedné přímce.

Příklady výše zmíněných transformací jsou znázorněny na obr. 1.3.



Obrázek 1.3: Příklady projektivních transformací krychle

2. Model Kamery

Model kamery popisuje zobrazení trojrozměrného prostoru do dvojrozměrné roviny. V případě rekonstrukce prostoru z fotografií se prostor označuje jako *scéna* a rovina jako *obraz*.

Nejdříve popíšeme nejjednodušší případ kamery – *dírkovou komoru* a poté ji postupně rozšíříme na model, který lépe popisuje dnešní fotoaparáty. Popis dírkové komory a odvození matic kamery je převzato z [11].

2.1. Popis základního modelu

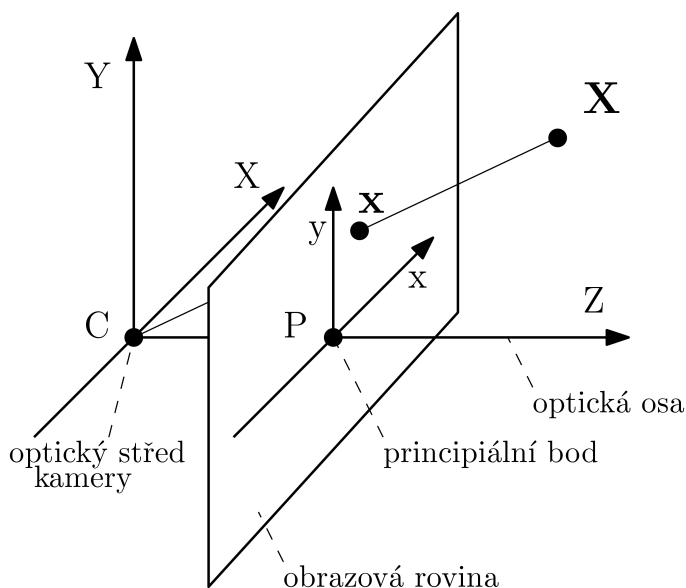
Proces snímání kamerou můžeme chápat jako lineární zobrazení bodu v prostoru na bod v obraze. Kamera tedy bude popsána pomocí tzv. *matice kamery*.

2.1.1. Dírková komora

Dírková komora (latinsky *Camera Obscura*) je jednoduché optické zařízení, které využívá fyzikální optický jev. Světlo, které se odrazí od předmětu, projde dírkou ve zdi temné místnosti a na protější stěně vytvoří převrácený obraz předmětu. Tento princip byl v Číně znám již v 5. století před naším letopočtem[27].

Matematický model, který zde bude popsán, se liší oproti skutečnosti v tom, že neuvazuje převrácený obraz. Obrazová rovina je totiž v modelu umístěna před optickým středem kamery a nikoli za ním.

Středem promítání uvažujme počátek Euklidovského prostoru. Dále uvažujme rovinu $Z = f$, která představuje *rovinu obrazu*. Někdy se této rovině také říká *ohnisková rovina*. Obrazová rovina je rovnoběžná s osami X a Y a leží ve vzdálenosti f od optického středu. f se nazývá ohnisková vzdálenost. Bod prostoru $\mathbf{X} = (X, Y, Z)^T$ se zobrazí jako bod $\mathbf{x} = (x, y)^T$ roviny obrazu, tak, že přímka, která spojuje bod \mathbf{X} s jeho obrazem \mathbf{x} , prochází středem promítání (viz obr. 2.1).



Obrázek 2.1: Dírková komora

Střed promítání tvoří *optický střed kamery*. Osa Z je v tomto případě totožná s *optickou osou* a protíná obrazovou rovinu v *principiálním bodě*. V literatuře je někdy tento bod nazýván též jako *hlavní bod* nebo jako *hlavní snímkový bod*. Optická osa je dána natočením kamery.

Jestliže tedy bod \mathbf{X} má souřadnice $(X, Y, Z)^\top$, pak se zobrazí na bod \mathbf{x} , který bude mít v prostoru souřadnice $(\frac{fX}{Z}, \frac{fY}{Z}, f)^\top$. Za použití homogenních souřadnic lze tuto závislost zapsat jako lineární zobrazení

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.1)$$

Matrice 3×4 v této rovnici představuje homogenní *matici kamery*. Matici kamery budeme obecně značit P . Můžeme tak lineární transformaci prostorového bodu \mathbf{X} na rovinný bod \mathbf{x} zjednodušeně zapsat jako

$$\mathbf{x} = P\mathbf{X} \quad (2.2)$$

V případě dírkové komory lze kamerovou matici P vyjádřit jako

$$P = K \cdot [I|\mathbf{0}] = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot [I|\mathbf{0}], \quad (2.3)$$

kde I je jednotková matice 3×3 a $\mathbf{0} = (0, 0, 0)^\top$. Matice K se nazývá *kalibrační matice* pro dírkovou komoru.

2.1.2. Digitální zařízení

Digitální kamery a fotoaparáty obsahují snímač, nejčastěji CCD nebo CMOS čip[17]. Čip je rozdělen na jednotlivé pixely, které mají určitý rozměr. Navíc ne vždy mají dokonale čtvercový tvar. Dále k jednotlivým pixelům digitální fotografie přistupujeme pomocí indexování tak, že pixel s indexy $(0, 0)$ je umístěn v jednom z rohů fotografie. Aby se tyto skutečnosti zahrnuly do matematického modelu, je potřeba upravit kalibrační matici.

Obrazová rovina má vlastní souřadný systém. Mluvíme o něm jako o *souřadném systému obrazu*. Souřadný systém, pro který platí, že jeho počátek je totožný s principiálním bodem, nazýváme *souřadnicový systém kamery*. Osy souřadného systému obrazu budeme značit x a y , osy souřadného systému kamery budeme značit x_{cam} a y_{cam} . Pokud má principiální bod v obrazovém souřadném systému souřadnice $(p_x, p_y)^T$, pak kalibrační matice je tvaru

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

Ideální pixel má tvar čtverce, ale u skutečného pixelu tomu tak nemusí být vždy. U některých zařízení mají pixely tvar obdélníku nebo dokonce kosodélníku. Kalibrační matice je proto upravena na tvar

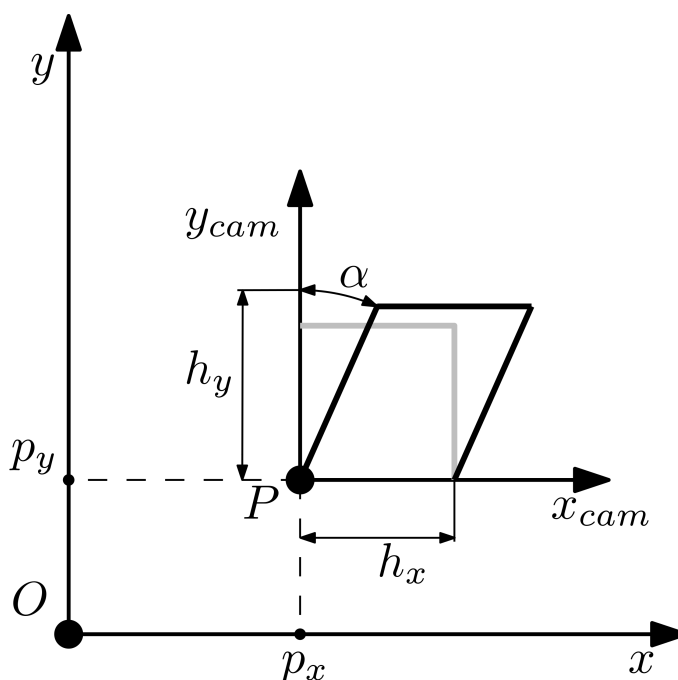
$$K = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.5)$$

kde

$$\alpha_x = \frac{f}{h_x}, \quad \alpha_y = \frac{f}{h_y}, \quad s = \alpha_y \operatorname{tg} \alpha,$$

kde h_x a h_y jsou délky stran jednoho pixelu na snímači a úhel α udává případné zkosení stran. V naprosté většině případů platí $s = 0$.

Dostali jsme tak tvar kalibrační matice, který umožňuje práci s digitálními fotografiemi. Všechny prvky kalibrační matice se považují za *vnitřní parametry kamery*. Vybrané parametry jsou společně se souřadným systémem obrazu a souřadným systémem kamery znázorněny na obr. 2.2.



Obrázek 2.2: Vnitřní parametry kamery

2.1.3. Rotace a translace kamery

Mimo vnitřní parametry má kamera i parametry vnější. *Vnější parametry* popisují polohu a natočení kamery. Optická osa kamery totiž nemusí být nutně rovnoběžná s osou Z . Její natočení vůči ose Z lze popsat pomocí rotační matice R . Podobně optický střed nemusí být umístěn v počátku, ale může se nacházet v libovolném bodě $\mathbf{C} = (X, Y, Z)^T$. Matice kamery je v takovém případě rovna

$$P = KR [I | -\mathbf{C}] = K [R | -R\mathbf{C}]. \quad (2.6)$$

Translaci kamery \mathbf{t} rozumíme vektor \mathbf{C} . Během výpočtu rekonstrukce je často vypočítána nejdříve relativní rotace R'_1 a translace \mathbf{t}'_1 jedné kamery v souřadném systému kamery druhé. Pokud má tato kamera rotaci R_2 a translaci \mathbf{t}_2 v globálním souřadném systému, pak lze rotaci R_1 a translaci \mathbf{t}_1 první kamery v globálním souřadném systému určit jako

$$\begin{aligned} R_1 &= R'_1 \cdot R_2 \\ \mathbf{t}_1 &= \mathbf{t}_2 + \mathbf{t}'_1 \cdot R_2. \end{aligned} \quad (2.7)$$

Pokud budeme uvažovat kalibrační matici K ve tvaru uvedeném v rovnici (2.5), dostaneme obecný model perspektivní kamery. Perspektivní kameru a její vybrané vlastnosti rozebereme v následující podkapitole.

2.1.4. Perspektivní kamera

Matici perspektivní kamery lze zcela obecně zapsat ve tvaru

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}. \quad (2.8)$$

V případě, že součin kalibrační a rotační matice vyjádříme jako $M = KR$, pak lze matici P z rovnice (2.6) také vyjádřit ve tvaru

$$P = KR [I | -\mathbf{C}] = [M | -M\mathbf{C}] = [M | \mathbf{p}_4]. \quad (2.9)$$

Pro souřadnice optického středu tedy platí

$$-M\mathbf{C} = \mathbf{p}_4, \quad (2.10)$$

a proto je získáme jako

$$\mathbf{C} = M^{-1}\mathbf{p}_4. \quad (2.11)$$

Kalibrační matici K a rotační matici R lze obdržet *RQ rozkladem*[15] matice M . Matice R je totiž ortogonální matice a matice K je v horním trojúhelníkovém tvaru. RQ rozklad lze provést následovně:

Mějme matice

$$Q_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \quad (2.12)$$

$$Q_y = \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \quad (2.13)$$

$$Q_z = \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.14)$$

Matici M obsahující prvky m_{mn} nejdříve vynásobme zprava maticí Q_x , přičemž konstanty c_x a s_x volíme následovně

$$c_x = \frac{-m_{33}}{\sqrt{m_{32}^2 + m_{33}^2}} \quad (2.15)$$

$$s_x = \frac{m_{32}}{\sqrt{m_{32}^2 + m_{33}^2}}. \quad (2.16)$$

Získáme tak matici $A = MQ_x$, jejíž prvek $a_{32} = 0$. Tuto matici vynásobme zprava maticí Q_y s konstantami

$$c_y = \frac{a_{33}}{\sqrt{a_{31}^2 + a_{33}^2}} \quad (2.17)$$

$$s_y = \frac{a_{31}}{\sqrt{a_{31}^2 + a_{33}^2}}. \quad (2.18)$$

Dostáváme tak matici $B = AQ_y$, jejíž oba prvky $b_{31} = 0$ a $b_{32} = 0$. K získání kalibrační matice postačuje vynásobit matici B zprava maticí Q_z , přičemž

$$c_z = \frac{-b_{22}}{\sqrt{b_{21}^2 + b_{22}^2}} \quad (2.19)$$

$$s_z = \frac{b_{21}}{\sqrt{b_{21}^2 + b_{22}^2}} \quad (2.20)$$

Výsledkem je tedy kalibrační matice

$$K = BQ_z = MQ_xQ_yQ_z. \quad (2.21)$$

Rotační matici pak snadno určíme jako

$$R = Q_z^T Q_y^T Q_x^T. \quad (2.22)$$

2.2. Radiální distorze

Obraz je oproti skutečnosti zkreslen také vlivem čočky. Nejčastější a největší typ zkreslení má závislost na vzdálenosti od středu obrazu. Tomuto typu zkreslení se říká *radiální distorze*. Obecně roste vliv radiální distorze s klesající ohniskovou vzdáleností.

Existují dva druhy radiální distorze. Prvním druhem je distorze typu soudek a druhým distorze typu poduška (viz obr. 2.3). Případně může celkové zkreslení vzniknout jako kombinace obou typů.

Označme (x, y) ideální souřadnice bodu v obraze, tedy místo, kde by se daný bod nacházel bez vlivu radiální distorze. Dále označme (x_d, y_d) souřadnice tohoto bodu tak, jak je zobrazen pod vlivem distorze. Pak

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = D_f(r) \begin{pmatrix} x \\ y \end{pmatrix}, \quad (2.23)$$

kde r je vzdálenost bodu od středu radiální distorze. Funkce $D_f(r)$ se nazývá *distorzní faktor*. Střed radiální distorze budeme značit (x_c, y_c) a vzdálenost r je pak určena jako

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (2.24)$$

Přepočítání v pixelových souřadnicích je vyjádřeno jako

$$x_d = x_c + D_f(r)(x - x_c) \quad (2.25)$$

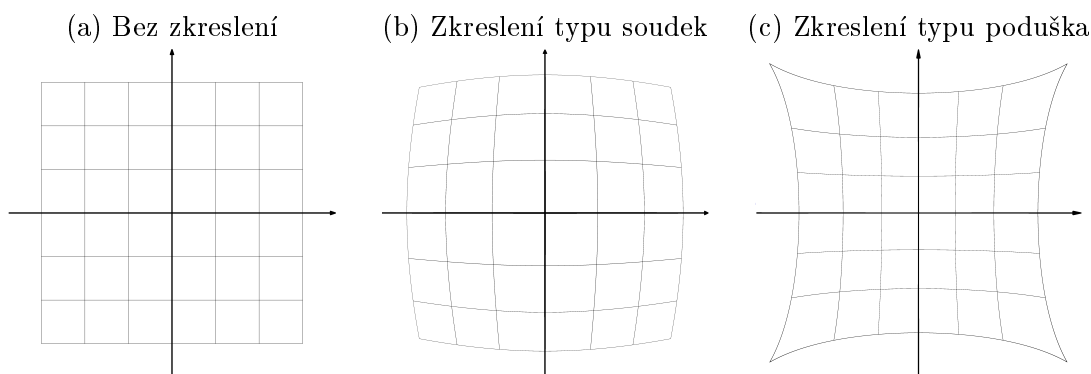
$$y_d = y_c + D_f(r)(y - y_c) \quad (2.26)$$

Distorzní faktor je vhodné aproximovat polynomem tvaru

$$D_f(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \dots \quad (2.27)$$

Koeficienty $\kappa_1, \kappa_2, \kappa_3, \dots$ společně se souřadnicemi středu radiální distorze (x_c, y_c) jsou také považovány za vnitřní parametry kamery. Je vhodné tyto parametry určit pomocí kalibrace. Při ní se střed radiální distorze často kvůli zjednodušení uvažuje shodný s principiálním bodem, i když ve skutečnosti totožné nejsou.

Vedle radiální distorze existuje ještě například *tangenciální distorze*. Působí kolmo na směr radiální distorze a je často zanedbatelná. Bývá způsobena špatným umístěním snímacího čipu vůči čočkám uvnitř fotoaparátu.



Obrázek 2.3: Příklady radialní distorze

2.3. Kalibrace

Kalibrací rozumíme proces, při kterém jsou nalezeny vnitřní parametry kamery. V případě, že tyto parametry jsou odhadnuty přímo z fotografií zachycujících rekonstruovanou scénu, hovoříme o tzv. *autokalibraci*.

Nalezení vnitřních parametrů kamery nám umožňuje provést tzv. *metrickou rekonstrukci* scény. Bez jejich nalezení je možné provést pouze *projektivní rekonstrukci* a takto zrekonstruovaná scéna je většinou zkreslena vůči skutečnosti, tj. mezi scénou a její rekonstrukcí existuje projektivní transformace, která obecně není afinní. Pokud je daná transformace podobnostní, pak se jedná o metrickou rekonstrukci.

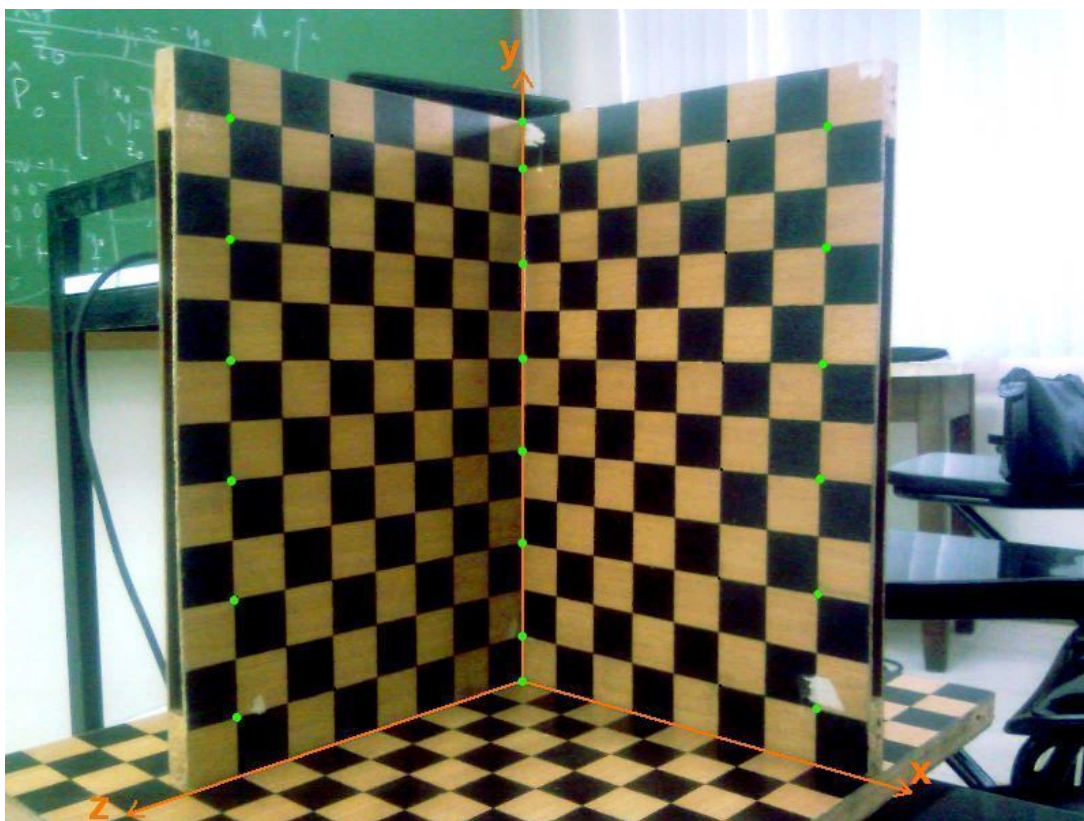
2.3.1. Kalibrace s použitím kalibračního vzoru

Klasická kalibrace vyžaduje použití *kalibračního vzoru*. Jeho konkrétní podoba záleží na použité metodě, často má podobu šachovnice s různým počtem polí na každé straně. Před samotnou rekonstrukcí je tedy s kamerou nutné provést nasnímání této šachovnice z různých vzdáleností a směrů, které jsou pro daný vzor přípustné, a následné odhadnutí parametrů z takto pořízených fotografií.

Některé metody místo jediné rovinné šachovnice používají dvě nebo tři navzájem kolmé desky s šachovnicí nebo jiným vzorem (viz obr. 2.5). Tento typ kalibračního vzoru sice zpravidla nabízí přesnější odhad parametrů, ale je mnohem náročnější na výrobu a správné sestavení.

Odhadnutí parametrů je možné díky znalosti reálných rozměrů šachovnice a předpokladu, že všechna pole šachovnice leží v jedné rovině. Přesný rozměr šachovnice bývá často zadáván jako vstup při spuštění kalibračního procesu.

Výhodou tohoto přístupu oproti autokalibraci je zpravidla přesnější odhad parametrů. Nevýhodou je naopak nutnost mít daný kalibrační vzor, který vyžaduje výrobu s co největší přesností. Zároveň je nutné pro každou použitou kameru pořídit samostatnou sérii fotografií. Důležité během pořizování snímků pro kalibraci je zachovat stejnou ohniskovou vzdálenost kamery a tuto vzdálenost posléze zachovat i při fotografování samotné scény. Ohnisková vzdálenost se mění nejen při zoomování, ale i se zaostřením fotoaparátu na focený objekt. Proto by měl být kalibrační vzor během kalibrace ve stejné vzdálenosti od kamery jako rekonstruovaný objekt scény.



Obrázek 2.5: Příklad vícerozměrného kalibračního vzoru. Převzato z [28].

Aplikace programu Matlab: Camera Calibrator

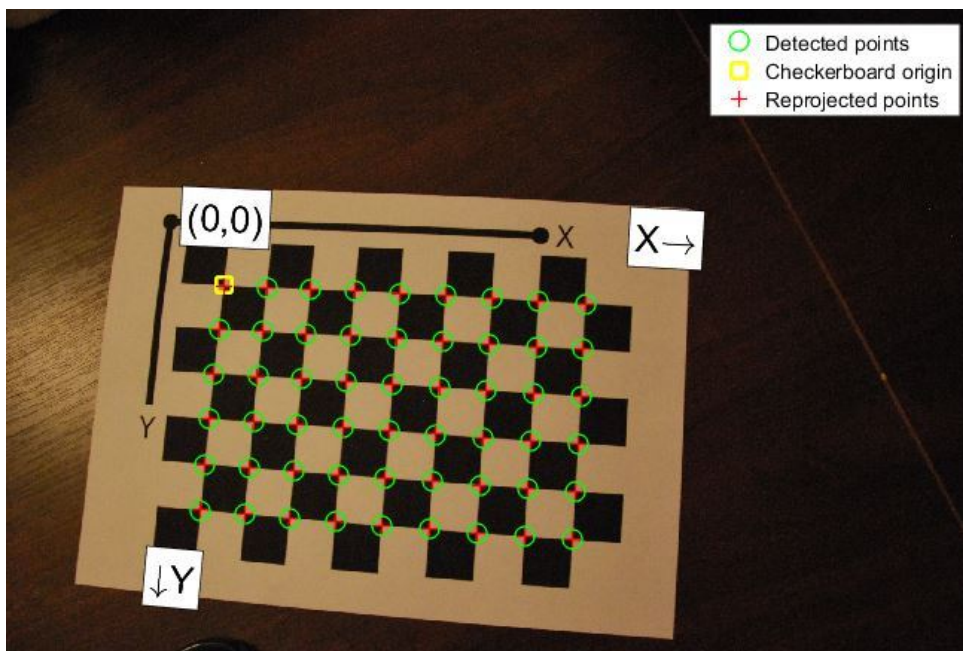
Pro kalibraci vlastního fotoaparátu byla v rámci této práce použita zabudovaná aplikace *Camera Calibrator* v programu *Matlab* která je dostupná od roku 2013.

Aplikaci lze spustit v *Matlabu* pomocí příkazu `cameraCalibrator`, nebo ji lze případně nalézt v záložce aplikací. Ovládání aplikace je poměrně intuitivní. Vyžaduje nejdříve vložit nejlépe mezi deseti a dvaceti fotografiemi kalibračního vzoru a poté zadat délku strany jednoho pole na kalibračním vzoru.

Aplikace vyžaduje šachovnicový kalibrační vzor. Tento vzor by měl mít na jedné straně sudý počet polí a na druhé straně lichý počet polí. To umožňuje lépe určit orientaci vzoru. Vhodný kalibrační vzor lze získat užitím příkazu `open checkerBoardPattern.pdf`.

Po vložení fotografií se automaticky spustí detekce vzoru v jednotlivých fotografiích (viz obr.2.6). Poté je možno v panelu nástrojů v sekci *Options* zvolit, zda se má distorzní faktor radiální distorze aproximovat jako polynom se dvěma nebo třemi členy (tj. polynom čtvrtého nebo šestého řádu). Případně lze navolit, zda se má uvažovat tangenciální distorze či zkosení s kalibrační matice K . Příslušná teorie je popsána v částech 2.1.2 a 2.2.

Následně lze již spustit samotnou kalibraci tlačítkem *Calibrate* v panelu nástrojů. Aplikace odhadne vnitřní parametry kamery. Zároveň určí natočení a umístění kamery vůči vzoru pro každou fotografii. Každé fotografii je tak přiřazena matice kamery. Aplikace poté odhadne přesnost určení vnitřních parametrů. Tato přesnost je vyjádřena pomocí průměru rozdílů mezi skutečnou polohou a polohou určenou užitím matice kamery pro vybrané body v obraze. Uživatel má zároveň možnost odstranit fotografie, které vykazují velkou nepřesnost a tedy mohou zkreslovat kalibraci.



Obrázek 2.6: Detekovaný kalibrační vzor na fotografii

Na závěr lze pomocí volby *Export Camera Parameters* v panelu nástrojů vyexportovat vnitřní parametry do prostředí Matlabu a zde již uložit například ve formátu *MAT*.

Více informací k aplikaci lze nalézt v nápovědě k programu *Matlab*. Podrobněji popsany způsob kalibrace lze nalézt v literatuře [12], [26].

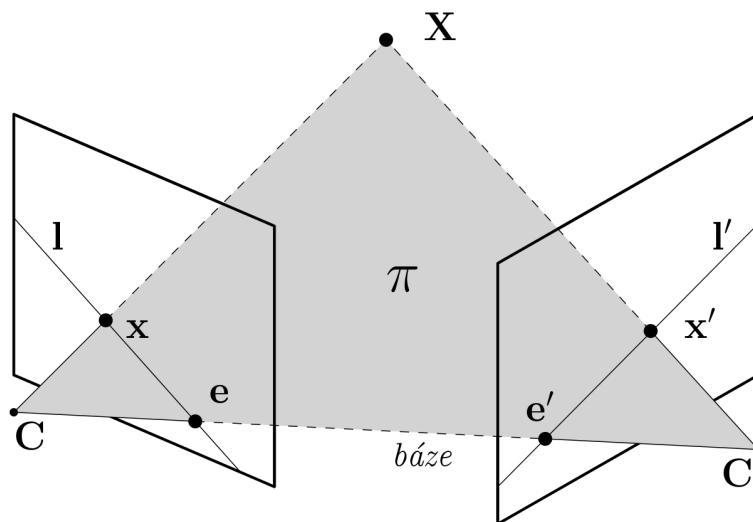
3. Geometrie dvou pohledů

V této kapitole se budeme zabývat vztahy mezi dvěma obrazy, které zobrazují stejnou scénu. Každý z těchto obrazů zobrazuje danou scénu z jiného směru (pohledu). Závislosti mezi objekty v těchto dvou pohledech popisuje *epipolární geometrie*. Užitím této geometrie ukážeme, že body v těchto pohledech jsou svázány pomocí *fundamentální matice*. Dále bude popsáno, jak tuto matici pro danou dvojici pohledů získat a při znalosti daných kalibračních matic dále určit *esenciální matici*. Esenciální matice již umožňuje provést rekonstrukci vzájemného umístění a natočení obou kamer. Následně lze již provést rekonstrukci scény. Na závěr kapitoly je popsána rektifikace pohledů, která umožňuje provést detailnější rekonstrukci celé scény. Tato kapitola čerpá především z [11].

Matici kamery prvního pohledu budeme značit P . Matici kamery druhého pohledu budeme značit P' . Podobně budeme značit například body v jednotlivých pohledech či kalibrační matice – objekt související s druhým pohledem budeme značit s apostrofem a objekt související s prvním pohledem bez apostrofu.

3.1. Epipolární geometrie

Uvažujme bod \mathbf{X} v prostoru \mathbb{P}^3 . Dále mějme dány dvě kamery se středy \mathbf{C} , \mathbf{C}' a dvě obrazové roviny. Při průmětu bodu \mathbf{X} s použitím obou kamer dostáváme body \mathbf{x} a \mathbf{x}' v příslušných obrazových rovinách. Paprsek spojující body \mathbf{X} a \mathbf{x} se zobrazí v druhém pohledu jako přímka l' . Obdobně paprsek spojující bod \mathbf{x}' s bodem \mathbf{X} se promítne na přímku l v prvním pohledu (viz. obr 3.1). Přímky l a l' se nazývají *epipolární přímky*. Tyto přímky leží v jedné rovině π , která se nazývá *epipolární rovina*. V této rovině také leží přímka spojující optické středy kamer \mathbf{C} a \mathbf{C}' . Této přímce se říká *báze* nebo *bázová přímka*. Průsečík přímky l s bází je tzv. *epipól* e . Průsečík přímky l' s bází je pak epipól e' . Je zřejmé, že všechny epipolární přímky jednoho pohledu se protínají právě v epipólu a všechny epipolární roviny se navzájem protínají v bázi.



Obrázek 3.1: Epipolární geometrie.

3.2. Fundamentální matice

Mějme v prostoru rovinu ρ , která neprochází ani jedním optickým středem a bod \mathbf{x} ležící v prvním obraze. Uvažujme, že bod \mathbf{x} je obrazem bodu \mathbf{X} , který leží v rovině ρ . Pak odpovídající bod \mathbf{x}' je dán zcela jednoznačně bodem \mathbf{X} a tedy i volbou roviny ρ . Pro každý bod \mathbf{X}_i roviny ρ existuje jiná dvojice bodů \mathbf{x}_i a \mathbf{x}'_i . To nám umožňuje zavést zobrazení H_π bodů v prvním pohledu na body v druhém pohledu. Protože existuje projektivní zobrazení mezi body \mathbf{x} a \mathbf{X} a také mezi body \mathbf{x}' a \mathbf{X} , je zobrazení H_ρ homografie. Zobrazení lze vyjádřit v maticovém zápisu jako

$$\mathbf{x}' = H_\rho \mathbf{x}, \quad (3.1)$$

kde H_ρ je matice tvaru 3×3 .

Epipolární přímku \mathbf{l}' , která prochází bodem \mathbf{x}' a epipólem \mathbf{e}' lze vyjádřit pomocí vektorového součinu

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_{\times} \mathbf{x}'. \quad (3.2)$$

Zápis $[\mathbf{a}]_{\times}$ značí antisymetrickou matici tvaru

$$[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \quad (3.3)$$

kde $\mathbf{a} = (a_1, a_2, a_3)^T$. Toto značení slouží k výpočtu vektorového součinu pomocí maticového součinu. Platí

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = (\mathbf{a}^T [\mathbf{b}]_{\times})^T. \quad (3.4)$$

Dosazením výrazu (3.1) do rovnice (3.2) dostáváme

$$\mathbf{l}' = [\mathbf{e}']_{\times} H_\rho \mathbf{x} = F \mathbf{x}, \quad (3.5)$$

kde F nazýváme *fundamentální matice*. Tato matice má hodnost 2, což vyplývá z hodnoty matic $[\mathbf{e}']_{\times}$ a H_π , a má 7 stupňů volnosti.

Dále platí, že pro dvojici odpovídajících si bodů \mathbf{x} a \mathbf{x}' platí

$$\mathbf{x}'^T F \mathbf{x} = 0, \quad (3.6)$$

přičemž pohledy musí mít navzájem různé optické středy.

Pro epipóly \mathbf{e} a \mathbf{e}' platí

$$F \mathbf{e} = \mathbf{0}, \quad (3.7)$$

$$F^T \mathbf{e}' = \mathbf{0}. \quad (3.8)$$

Nyní si uvedeme způsob, kterým lze provést výpočet fundamentální matice.

3.2.1. Výpočet fundamentální matice

Mějme dva odpovídající si body $\mathbf{x} = (x, y, 1)^T$ a $\mathbf{x}' = (x', y', 1)^T$ a fundamentální matici F s prvky f_{ij} , pak jejich dosazením do rovnice (3.6) dostáváme

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (3.9)$$

Součet v rovnici lze přepsat jako skalární součinu vektorů a rovnice je tak převedena do tvaru

$$\begin{aligned} (x'x, x'y, x', y'x, y'y, y', x, y, 1) \cdot (f_{11}, f_{12}, f_{13}, f_{22}, f_{23}, f_{23}, f_{31}, f_{32}, f_{33}) = \\ (x'x, x'y, x', y'x, y'y, y', x, y, 1) \cdot \mathbf{f} = 0. \end{aligned} \quad (3.10)$$

Pokud budeme mít n dvojic odpovídajících si bodů $\mathbf{x}_i, \mathbf{x}'_i, i = 1, \dots, n$ pak lze sestavit soustavu lineárních rovnic

$$\mathbf{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (3.11)$$

Jedná se o homogenní soustavu rovnic. Aby pro soustavu existovalo exaktní řešení, musí mít matice A nejvýše hodnotu rovnu osmi. Navíc se často přidává požadavek

$$\|\mathbf{f}\| = 1. \quad (3.12)$$

V případě, že hodnota matice je právě osm, tak dostaneme jediné řešení soustavy. Pokud by hodnota matice byla vyšší než osm, pak lze k nalezení přibližného řešení použít například metodu nejmenších čtverců.

K výpočtu fundamentální matice lze spolehlivě použít například následující dvě metody, které se liší počtem vstupních dvojic bodů.

7-bodová metoda

Vstupem této metody je sedm dvojic odpovídajících si bodů. Matice A v rovnici (3.11) má tedy obecně hodnotu rovnu sedmi. Řešením rovnice (3.11) za předpokladu (3.12) je množina vektorů tvaru

$$\mathbf{f} = \alpha \cdot \mathbf{f}_1 + (1 - \alpha) \cdot \mathbf{f}_2, \quad (3.13)$$

kde $\alpha \in (0, 1)$ a vektory \mathbf{f}_1 a \mathbf{f}_2 jsou lineárně nezávislé jednotkové vektory, které jsou řešením rovnice $\mathbf{A}\mathbf{f} = \mathbf{0}$. Tyto vektory lze získat například užitím SVD rozkladu[13].

Odpovídající fundamentální matice k vektoru \mathbf{f} je pak tvaru

$$F = \alpha \cdot F_1 + (1 - \alpha) \cdot F_2. \quad (3.14)$$

K přesnému určení matice F využijeme její významné vlastnosti. Fundamentální matice je totiž vždy singulární. Proto při jejím výpočtu můžeme zavést požadavek na nulovost determinantu

$$\det(F) = 0. \quad (3.15)$$

Dosazením vztahu (3.14) do vztahu (3.15) tak dostaneme kubickou rovnici pro neznámou α

$$\det(\alpha \cdot F_1 + (1 - \alpha) \cdot F_2) = 0, \quad (3.16)$$

která má jedno nebo tři reálná řešení. Pokud vychází tři řešení, je nutné dále počítat se všemi třemi a v některém z dalších kroků rekonstrukce pak vybrat nejlepší z nich. Proto je vhodné tuto metodu užít uvnitř iterační metody, která je schopna porovnat nalezené matice. Vhodná je například metoda RANSAC[10].

Normalizovaná 8-bodová metoda

Tato metoda vyžaduje jako vstupy osm dvojic odpovídajících si bodů. Řešení rovnice (3.11) za předpokladu (3.12) lze pak určit například užitím SVD rozkladu[13].

Jak název napovídá, metoda ke zvýšení své přesnosti vyžaduje normalizaci vstupních dat. Souřadnice bodů musí být upraveny tak, aby jejich společné těžiště leželo v počátku a průměrná eukleidovská vzdálenost bodů od počátku byla $\sqrt{2}$. Úprava se provádí zvlášť pro body v každém z pohledů.

Body z prvního pohledu se změni na body splňující tyto dvě podmínky pomocí projektivního zobrazení T_1 a body z druhého pohledu pomocí zobrazení T_2 . Následně se pro dvojice transformovaných bodů určí fundamentální matice \hat{F} . Výsledná matice F je pak určena vztahem

$$F = T_2^T \cdot \hat{F} \cdot T_1. \quad (3.17)$$

3.3. Esenciální matice

Esenciální matice E je speciální případ fundamentální matice. Udává vztah mezi dvěma odpovídajícími si body převedenými do tzv. *normalizovaných souřadnic*. Uvažujme, že bod \mathbf{x} leží v pohledu, pro který známe příslušnou kalibrační matici K . Pak normalizované souřadnice tohoto bodu jsou

$$\hat{\mathbf{x}} = K^{-1}\mathbf{x}. \quad (3.18)$$

Rozklad matice kamery uvedený v rovnici (2.6) je možné přepsat do vhodnějšího tvaru

$$P = K [R|\mathbf{t}]. \quad (3.19)$$

Na bod s normalizovanými souřadnicemi můžeme tedy nahlížet jako na bod, který vznikne zobrazením bodu \mathbf{X} do obrazu s jednotkovou kalibrační maticí, neboť platí

$$\hat{\mathbf{x}} = K^{-1}\mathbf{x} = K^{-1}P\mathbf{X} = K^{-1}K [R|\mathbf{t}] \mathbf{X} = [R|\mathbf{t}] \mathbf{X} = I [R|\mathbf{t}] \mathbf{X}. \quad (3.20)$$

Matice kamery s jednotkovou kalibrační maticí se nazývá *normalizovaná matice kamery*.

Pro esenciální matici a libovolnou dvojici odpovídajících si bodů s normalizovanými souřadnicemi platí

$$\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0. \quad (3.21)$$

Substitucí za $\hat{\mathbf{x}}$ a $\hat{\mathbf{x}}'$ dostáváme rovnici

$$\mathbf{x}'(K'^{-1})^T E K^{-1}\mathbf{x} = 0, \quad (3.22)$$

ze které nám při porovnání s rovnicí (3.6) vyplyne vztah

$$E = K'^T F K. \quad (3.23)$$

Esenciální matici je proto možné jednoduše určit z fundamentální matice, pokud známe kalibrační matice obou pohledů.

3.3.1. Určení matice kamery užitím esenciální matice

Nalezení esenciální matice je důležitým krokem při rekonstrukci scény. Z esenciální matice je možné určit matice kamer obou pohledů. Předpokládá se přitom, že obě matice kamer jsou normalizované a tvaru $P = [I|\mathbf{0}]$ a $P' = [R|\mathbf{t}]$. Potom lze esenciální matici rozložit na

$$E = [\mathbf{t}]_{\times} R \quad (3.24)$$

K nalezení vzájemné pozice kamer je proto postačující nalézt vektor translace \mathbf{t} a matici rotace R popisující umístění a natočení kamery druhého pohledu.

Další vlastností esenciální matice je, že jedno její singulární číslo je rovno nule a zbylá dvě singulární čísla se sobě rovnají. Esenciální matici tak lze užitím SVD rozkladu[13] rozložit na

$$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V, \quad (3.25)$$

kde U a V jsou matice typu 3×3 . Vektor translace \mathbf{t} lze získat z antisymetrické matice $[\mathbf{t}]_{\times}$ užitím vztahu (3.3). Vektor translace však může nabývat i opačné orientace. Matici $[\mathbf{t}]_{\times}$ určíme z rovnice

$$[\mathbf{t}]_{\times} = UZU^{\top}, \quad (3.26)$$

kde

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.27)$$

Rotaci R lze určit užitím rovnice

$$R_1 = UWV^{\top} \quad (3.28)$$

nebo

$$R_2 = UW^{\top}V^{\top}, \quad (3.29)$$

kde

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.30)$$

přičemž $R_1 \neq R_2$.

Protože ani vektor translace \mathbf{t} ani matice rotace R nejsou určeny jednoznačně, pro výpočet matice kamery P' máme čtyři následující možnosti

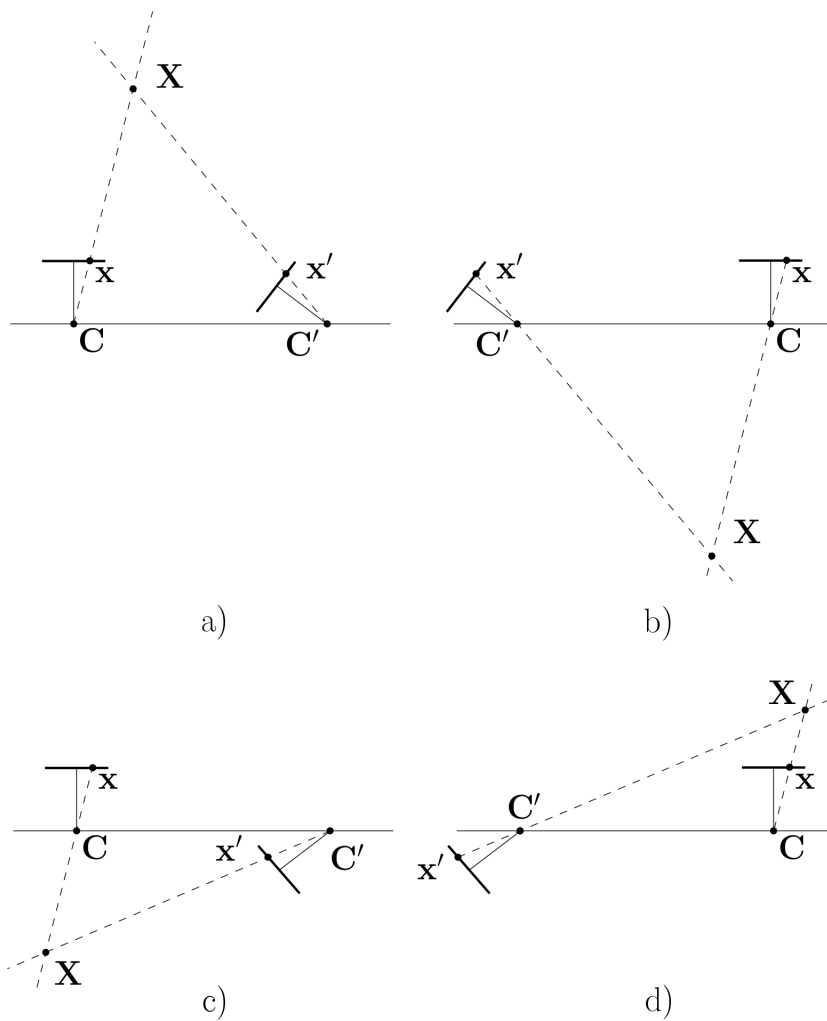
$$P'_1 = [R_1|\mathbf{t}] \quad (3.31)$$

$$P'_2 = [R_1|-\mathbf{t}] \quad (3.32)$$

$$P'_3 = [R_2|\mathbf{t}] \quad (3.33)$$

$$P'_4 = [R_2|-\mathbf{t}]. \quad (3.34)$$

Správnou matici kamery vybereme pomocí rekonstrukce malého počtu bodů. Dostatečně velký vzorek se skládá alespoň z dvanácti bodů. Při rekonstrukci se kontroluje, zda body leží před kamerou nebo za ní (viz. obr. 3.2). Kontrolu provedeme pomocí zpětného zobrazení bodu scény na bod obrazu. Pokud je souřadnice z před homogenizací záporná,



Obrázek 3.2: Čtyři možnosti řešení kalibračních kamer získaných z esenciální matice a příklad rekonstrukce jednoho bodu, pouze možnost a) umožňuje správnou rekonstrukci.

pak bod leží za kamerou. Z daných čtyř matic se tedy vybere ta, pro niž nejvíce bodů leží v prostoru před oběma kamerami.

Po získání matice kamery P' je ještě nutné zleva vynásobit obě kamery P a P' příslušnými kalibračními maticemi.

3.4. Rekonstrukce scény

Po získání obou matic kamer P a P' je již možné provést výslednou rekonstrukci scény. Jedním z nejjednodušších přístupů je užití lineární triangulace.

Pro bod scény \mathbf{X} a jeho zobrazení \mathbf{x} v prvním pohledu platí

$$\mathbf{x} = P\mathbf{X}. \quad (3.35)$$

Obě strany rovnice představují dva navzájem rovnoběžné vektory a proto pro jejich vektorový součin platí

$$\mathbf{x} \times P\mathbf{X} = \mathbf{0}. \quad (3.36)$$

Po jejím rozepsání získáme tři samostatné rovnice

$$\begin{aligned} x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) &= 0, \end{aligned} \quad (3.37)$$

kde $\mathbf{p}^{i\top}$ je i -tý řádek matice kamery P . Jelikož jsou tyto rovnice lineární pro složky \mathbf{X} lze využitím obou pohledů sestavit soustavu rovnic tvaru $A\mathbf{X} = \mathbf{0}$, kde

$$A = \begin{pmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{pmatrix} \quad (3.38)$$

Po jejím vyřešení, například užitím SVD rozkladu[13], získáme výsledný bod scény \mathbf{X} .

3.5. Výpočet kamery z korespondencí mezi scénou a obrazem

Na závěr této kapitoly bude ukázáno, jak je možné určit matici kamery z korespondencí mezi body scény a body obrazu. Platí rovnice (3.36) a navíc

$$\mathbf{x} \times P\mathbf{X} = \begin{pmatrix} y\mathbf{p}^{3\top}\mathbf{X} & - & \mathbf{p}^{2\top}\mathbf{X} \\ \mathbf{p}^{1\top}\mathbf{X} & - & x\mathbf{p}^{3\top}\mathbf{X} \\ x\mathbf{p}^{2\top}\mathbf{X} & - & y\mathbf{p}^{1\top}\mathbf{X} \end{pmatrix}. \quad (3.39)$$

Pro vektory zřejmě platí

$$\mathbf{p}^{i\top}\mathbf{X} = \mathbf{X}^\top\mathbf{p}^i, \quad (3.40)$$

což nám umožňuje sestavit soustavu tří rovnic.

$$\begin{bmatrix} \mathbf{0}^\top & -\mathbf{X}^\top & y\mathbf{X}^\top \\ \mathbf{X}^\top & \mathbf{0}^\top & -x\mathbf{X}^\top \\ -y\mathbf{X}^\top & x\mathbf{X}^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0}. \quad (3.41)$$

Tyto tři rovnice jsou lineárně závislé, takže použijeme pouze dvě z nich

$$\begin{bmatrix} \mathbf{0}^\top & -\mathbf{X}^\top & y\mathbf{X}^\top \\ \mathbf{X}^\top & \mathbf{0}^\top & -x\mathbf{X}^\top \end{bmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} = \mathbf{0}. \quad (3.42)$$

Máme tedy 12 neznámých prvků matice kamery p_{ij} . Matice kamery má 11 stupňů volnosti, k jejímu získání tedy postačuje 11 rovnic.

Každá dvojice bodů $\mathbf{x}_i, \mathbf{X}_i$, pro které platí $\mathbf{x}_i = P\mathbf{X}_i$ nám dodá dvě rovnice. K určení matice kamery nám tedy postačuje 6 korespondencí, přičemž z šesté korespondence použijeme pouze jednu rovnici. Sestavíme tak homogenní soustavu rovnic, která je tvaru $A\mathbf{p} = \mathbf{0}$, kde $\mathbf{p}^\top = (\mathbf{p}^{1\top}, \mathbf{p}^{2\top}, \mathbf{p}^{3\top})$. Tuto rovnici lze podobně jako v minulé kapitole řešit pomocí SVD rozkladu[13].

Výše popsaný algoritmus je označován jako DLT – Direct Linear Transformation.

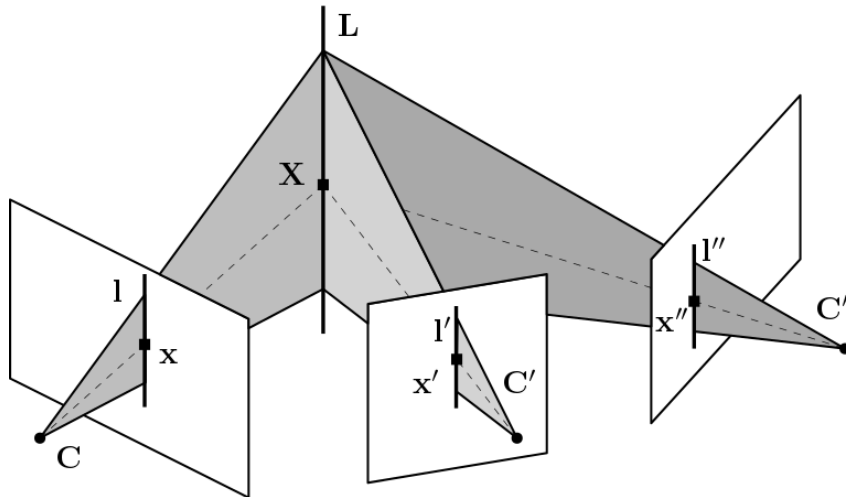
4. Geometrie tří pohledů

Kromě geometrie dvou pohledů existuje i geometrie tří pohledů, která popisuje vztah mezi trojicí vzájemných bodů, přímkou nebo jejich kombinací. Tato geometrie využívá tzv. *trifokálního tenzoru*. Kromě těchto geometrií lze ještě popsat geometrii pro čtyři pohledy, ale tato už je vysoce složitá a její praktické užití není tak časté.

V této kapitole budou popsány vztahy mezi objekty ve třech pohledech a jak s tím souvisí výše zmíněný trifokální tenzor. Dále bude popsána vybraná metoda pro výpočet trifokálního tenzoru.

Podobně jako v případě geometrie dvou pohledů, budeme i zde v značení rozlišovat mezi scénou a jednotlivými obrazy. Uvažujme bod \mathbf{X} ležící na přímce \mathbf{L} v prostoru \mathbb{P}^3 . Dále mějme dány tři kamery se středy \mathbf{C} , \mathbf{C}' , \mathbf{C}'' a tři odpovídající obrazové roviny. Při průmětu bodu \mathbf{X} s použitím všech kamer dostáváme body \mathbf{x} , \mathbf{x}' a \mathbf{x}'' v příslušných obrazových rovinách. Stejně tak při průmětu přímky \mathbf{L} dostaneme přímky \mathbf{l} , \mathbf{l}' a \mathbf{l}'' (viz obr. 4.1).

Tato kapitola čerpá z [11] a [16].



Obrázek 4.1: Geometrie tří pohledů

4.1. Trifokální tenzor

Trifokální tenzor \mathcal{T} slouží k popsání vztahů odpovídajících si bodů a přímek mezi třemi pohledy. Funguje tedy podobně jako fundamentální matice pro dva pohledy (viz. kapitola 3.2 na str. 24).

4.1.1. Značení a symbolika

Trifokální tenzor \mathcal{T} je často reprezentován pomocí tří matic řádu 3×3 . Má tedy 27 prvků, ale stupeň volnosti trifokálního tenzoru je pouze 18. Tato reprezentace se vyjadřuje v maticovém zápisu

$$\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}. \quad (4.1)$$

Dále budeme používat symbol $|\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3|$, případně $|\mathcal{T}_i|$. Pro tyto symboly platí

$$\mathbf{l}'^\top |\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3| \mathbf{l}'' = \mathbf{l}'^\top |\mathcal{T}_i| \mathbf{l}'' = (\mathbf{l}'^\top \mathcal{T}_1 \mathbf{l}'', \mathbf{l}'^\top \mathcal{T}_2 \mathbf{l}'', \mathbf{l}'^\top \mathcal{T}_3 \mathbf{l}''). \quad (4.2)$$

Dále budeme v některých rovnicích používat tzv. *tenzorovou notaci*. Zápis T_i^{jk} značí prvek i -té matice, index j značí řádek a index k značí sloupec matice, ve kterém se prvek nachází. Pokud se libovolný index nachází v rovnici opakovaně, přičemž se vyskytuje pouze na jedné straně rovnice, pak v dané rovnici uvažujeme součet přes všechny hodnoty tohoto indexu. Pro větší přehlednost bude v některých rovnicích tento součet vyjádřen sumačním znakem \sum a součet přes daný index v takovém případě neprovádíme dvakrát.

4.1.2. Vztahy mezi třemi pohledy

Jak již bylo zmíněno, podobně jako fundamentální matice popisuje jistou závislost mezi dvěma odpovídajícími si body, tak i trifokální tenzor popisuje závislost odpovídajících si bodů a přímek. Při použití výše zmíněné notace a symbolů platí následující vztahy. Nejprve si uveďme vztahy pro trojici přímek

$$\mathbf{I}^\top | \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 | \mathbf{I}'' = \mathbf{I}^\top \quad (4.3)$$

$$(\mathbf{I}^\top | \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 | \mathbf{I}'') [\mathbf{I}]_\times = \mathbf{0}^\top. \quad (4.4)$$

Další vztah platí pro bod \mathbf{x} z prvního pohledu a přímky \mathbf{I}, \mathbf{I}'' z druhého a třetího pohledu

$$\mathbf{I}^\top \left(\sum_i x_i \mathcal{T}_i \right) \mathbf{I}'' = 0. \quad (4.5)$$

Následující vztah popisuje geometrii mezi body \mathbf{x}, \mathbf{x}'' v prvním a třetím pohledu a přímkou \mathbf{I} v druhém pohledu

$$\mathbf{I}^\top \left(\sum_i x_i \mathcal{T}_i \right) [\mathbf{x}'']_\times = \mathbf{0}^\top. \quad (4.6)$$

Předposlední vztah popisuje opět vztah mezi dvěma body a jednou přímkou. Tentokrát však uvažujeme body \mathbf{x}, \mathbf{x}' a přímkou \mathbf{I}''

$$[\mathbf{x}']_\times \left(\sum_i x_i \mathcal{T}_i \right) \mathbf{I}'' = \mathbf{0}. \quad (4.7)$$

Poslední vztah pak platí pro trojici bodů $\mathbf{x}, \mathbf{x}', \mathbf{x}''$

$$[\mathbf{x}']_\times \left(\sum_i x_i \mathcal{T}_i \right) [\mathbf{x}'']_\times = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (4.8)$$

Užitím tenzorové notace lze výše zmíněné vztahy zapsat také ve tvaru

$$(l_r \varepsilon_{ris}) l'_j l''_k \mathcal{T}_i^{jk} = 0_s \quad (4.9)$$

$$x_i l'_j l''_k \mathcal{T}_i^{jk} = 0 \quad (4.10)$$

$$x_i l'_j (x''_k \varepsilon_{kqs}) \mathcal{T}_i^{jq} = 0_s \quad (4.11)$$

$$x_i (x'_j \varepsilon_{jpr}) l''_k \mathcal{T}_i^{pk} = 0_r \quad (4.12)$$

$$x_i (x'_j \varepsilon_{jpr}) (x''_k \varepsilon_{kqs}) \mathcal{T}_i^{pq} = 0_{rs}, \quad (4.13)$$

kde $i, j, k, r, s, t = 1, 2, 3$ a ε je Levi-Civitův tenzor s prvky

$$\varepsilon_{jkl} = \begin{cases} 1, & \text{pokud je } jkl \text{ sudá permutace } 123 \\ -1, & \text{pokud je } jkl \text{ lichá permutace } 123 \\ 0, & \text{jinak} \end{cases} \quad (4.14)$$

V této práci se omezíme na rekonstrukci užitím trojice bodů. Budou nás tedy zajímat především vztahy (4.8) a (4.13), které popisují vztahy mezi třemi si odpovídajícími body.

$$[\mathbf{x}']_{\times} \left(\sum_i x_i \mathcal{T}_i \right) [\mathbf{x}'']_{\times} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (4.8)$$

$$x_i (x'_j \varepsilon_{jpr}) (x''_k \varepsilon_{kqs}) \mathcal{T}_i^{pq} = 0_{rs} \quad (4.13)$$

4.1.3. Výpočet epipolární geometrie z trifokálního tenzoru

Trifokální tenzor \mathcal{T} v sobě uchovává informace i o každé dvojici ze tří obrazů. Navíc lze ukázat, že tyto informace jsou dostatečné k odvození epipolárních geometrií a fundamentálních matic.

V této části necht' F_{21} značí fundamentální matici pro druhý a první pohled a F_{31} značí fundamentální matici pro třetí a první pohled. Pro tři pohledy dále máme celkem šest epipólů \mathbf{e}_{12} , \mathbf{e}_{13} , \mathbf{e}_{21} , \mathbf{e}_{23} , \mathbf{e}_{31} , \mathbf{e}_{32} , kde \mathbf{e}_{ij} je epipól ležící v i -tém obraze a představující zobrazení optického středu j -té kamery. Pak epipóly \mathbf{e}' , \mathbf{e}'' budeme v tomto případě rozumět $\mathbf{e}' = \mathbf{e}_{21}$ a $\mathbf{e}'' = \mathbf{e}_{31}$.

Mějme libovolnou přímku \mathbf{l}' v druhém pohledu. Tato přímka je zobrazením roviny π scény. V této rovině leží libovolný bod \mathbf{X} , který se zobrazí jako bod \mathbf{x} v prvním pohledu a jako bod \mathbf{x}' v třetím pohledu. Pak homografii H_{13} popisující zobrazení $\mathbf{x}'' = H_{13}(\mathbf{l}') \cdot \mathbf{x}$ lze vyjádřit jako matici H_{13} pomocí trifokálního tenzoru \mathcal{T}

$$H_{13}(\mathbf{l}') = [\mathcal{T}_1^{\top}, \mathcal{T}_2^{\top}, \mathcal{T}_3^{\top}] \mathbf{l}'. \quad (4.15)$$

Obdobně pro $\mathbf{x}' = H_{12}(\mathbf{l}'') \cdot \mathbf{x}$

$$H_{12}(\mathbf{l}'') = [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] \mathbf{l}''. \quad (4.16)$$

Máme tedy rovnici popisující vztah mezi body \mathbf{x} a \mathbf{x}' .

$$\mathbf{x}' = H_{12}(\mathbf{l}'') \cdot \mathbf{x} = [\mathcal{T}_1 \mathbf{l}'', \mathcal{T}_2 \mathbf{l}'', \mathcal{T}_3 \mathbf{l}''] \mathbf{x} = ([\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] \mathbf{l}'') \mathbf{x}. \quad (4.17)$$

Uvažujme, že \mathbf{l}' je epipolární přímka v druhém pohledu odpovídající bodu \mathbf{x} . Na přímce \mathbf{l}' tedy leží epipól \mathbf{e}' i bod \mathbf{x}' . Dle vztahu (1.6) na str. 7 pak platí

$$\mathbf{l}' = [\mathbf{e}']_{\times} \mathbf{x}' = [\mathbf{e}']_{\times} ([\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] \mathbf{l}'') \mathbf{x} \quad (4.18)$$

a užitím vztahu (3.5) na str. 24 lze dále odvodit

$$F_{21} = [\mathbf{e}']_{\times} [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] \mathbf{l}''. \quad (4.19)$$

Protože vektor \mathbf{l}'' lze zvolit zcela libovolně, je vhodné místo něj zvolit epipól \mathbf{e}'' . Dostáváme tak

$$F_{21} = [\mathbf{e}']_{\times} [\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3] \mathbf{e}''. \quad (4.20)$$

Obdobně pak obdržíme fundamentální matici F_{31}

$$F_{31} = [\mathbf{e}'']_{\times} [\mathcal{T}_1^{\top}, \mathcal{T}_2^{\top}, \mathcal{T}_3^{\top}] \mathbf{e}'. \quad (4.21)$$

Dostali jsme tak fundamentální matice F_{21} a F_{31} užitím tenzoru a epipólů.

Dále uvažujme, že matice kamery prvního pohledu je ve tvaru

$$P = [I|\mathbf{0}], \quad (4.22)$$

pak matice kamery druhého pohledu je

$$P' = [[\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]\mathbf{e}''|\mathbf{e}']. \quad (4.23)$$

Nyní zbývá určit kameru třetího pohledu. Tu nelze určit obdobným způsobem, neboť není zaručeno, že rekonstrukce scény užitím kamer P' , P'' nám dá stejný výsledek jako užitím kamer P , P'' . Musíme tedy hledat matici kamery P'' v obecnějším tvaru

$$P'' = [[\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3]\mathbf{e}' + \mathbf{e}''\mathbf{v}^\top|\lambda\mathbf{e}''], \quad (4.24)$$

kde \mathbf{v} je obecný vektor a λ je konstanta. Pak lze ukázat, že matice P'' nabývá tvaru

$$P'' = [(\mathbf{e}''\mathbf{e}''^\top - I)[\mathcal{T}_1^\top, \mathcal{T}_2^\top, \mathcal{T}_3^\top]\mathbf{e}'|\mathbf{e}'']. \quad (4.25)$$

Jak je patrné, k určení matic kamer je nutné nejprve určit epipóly \mathbf{e}' , \mathbf{e}'' . Ty lze obdržet užitím následujících rovnic

$$\begin{aligned} \mathbf{e}'^\top[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] &= \mathbf{0} \\ \mathbf{e}''^\top[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] &= \mathbf{0}' \end{aligned} \quad (4.26)$$

kde vektory \mathbf{u}_i , \mathbf{v}_i lze získat jako jádra lineárních zobrazení \mathcal{T}_i

$$\begin{aligned} \mathbf{u}_i^\top \mathcal{T}_i &= \mathbf{0}^\top \\ \mathcal{T}_i \mathbf{v}_i &= \mathbf{0} \end{aligned} \quad (4.27)$$

Z tenzoru \mathcal{T} lze tedy získat epipolární geometrii a matice kamer.

4.1.4. Výpočet trifokálního tenzoru užitím matic kamer

Nyní si ukážeme opačný postup, kdy z matic kamer získáme tenzor. Mějme tři pohledy, které jsou dány maticemi kamer

$$\begin{aligned} P &= \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{34} \\ p_{31} & p_{32} & p_{33} & p_{44} \end{pmatrix} \\ P' &= \begin{pmatrix} p'_{11} & p'_{12} & p'_{13} & p'_{14} \\ p'_{21} & p'_{22} & p'_{23} & p'_{34} \\ p'_{31} & p'_{32} & p'_{33} & p'_{44} \end{pmatrix} \\ P'' &= \begin{pmatrix} p''_{11} & p''_{12} & p''_{13} & p''_{14} \\ p''_{21} & p''_{22} & p''_{23} & p''_{34} \\ p''_{31} & p''_{32} & p''_{33} & p''_{44} \end{pmatrix}, \end{aligned} \quad (4.28)$$

pro bod scény \mathbf{X} a jeho zobrazení \mathbf{x} , \mathbf{x}' , \mathbf{x}'' platí

$$\begin{aligned} k\mathbf{x} &= P\mathbf{X} \\ k'\mathbf{x}' &= P'\mathbf{X}, \\ k''\mathbf{x}'' &= P''\mathbf{X} \end{aligned} \quad (4.29)$$

kde k, k', k'' jsou nenulové konstanty. Tyto tři vztahy lze upravit do tvaru

$$\begin{aligned} P\mathbf{X} - \mathbf{x}k &= 0 \\ P'\mathbf{X} - \mathbf{x}'k' &= 0, \\ P''\mathbf{X} - \mathbf{x}''k'' &= 0. \end{aligned} \quad (4.30)$$

a dále lze sloučit do jediné soustavy rovnic

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & x_1 & 0 & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} & x_2 & 0 & 0 \\ p_{31} & p_{32} & p_{33} & p_{34} & x_3 & 0 & 0 \\ p'_{11} & p'_{12} & p'_{13} & p'_{14} & 0 & x'_1 & 0 \\ p'_{21} & p'_{22} & p'_{23} & p'_{24} & 0 & x'_2 & 0 \\ p'_{31} & p'_{32} & p'_{33} & p'_{34} & 0 & x'_3 & 0 \\ p''_{11} & p''_{12} & p''_{13} & p''_{14} & 0 & 0 & x''_1 \\ p''_{21} & p''_{22} & p''_{23} & p''_{24} & 0 & 0 & x''_2 \\ p''_{31} & p''_{32} & p''_{33} & p''_{34} & 0 & 0 & x''_3 \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ -k \\ -k' \\ -k'' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (4.31)$$

Dostáváme tak homogenní soustavu lineárních rovnic tvaru $A\mathbf{b} = \mathbf{0}$. Matice A je řádu 9×7 . Aby soustava rovnic měla nenulové řešení, musí mít matice A hodnotu nejvýše 6. Každý její libovolný subdeterminant matice řádu 7×7 je proto roven 0. Vybereme následující čtvercovou submatici

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} & x_1 & 0 & 0 \\ p_{21} & p_{22} & p_{23} & p_{24} & x_2 & 0 & 0 \\ p_{31} & p_{32} & p_{33} & p_{34} & x_3 & 0 & 0 \\ p'_{i1} & p'_{i2} & p'_{i3} & p'_{i4} & 0 & x'_i & 0 \\ p'_{j1} & p'_{j2} & p'_{j3} & p'_{j4} & 0 & x'_j & 0 \\ p''_{k1} & p''_{k2} & p''_{k3} & p''_{k4} & 0 & 0 & x''_k \\ p''_{l1} & p''_{l2} & p''_{l3} & p''_{l4} & 0 & 0 & x''_l \end{pmatrix}, \quad (4.32)$$

kde $i, j, k, l \in \{1, 2, 3\}$, $i \neq j$, $k \neq l$. Označme $u \in \{1, 2, 3\}$ index chybějícího řádku matice P' a označme $v \in \{1, 2, 3\}$ index chybějícího řádku matice P'' v rovnici (4.32). Podle výběru chybějících řádků v matici tak dostáváme celkem devět různých submatic. Vybranou submatici označíme A_{uv} . Pak determinant této submatice lze určit pomocí rozvoje podle řádků a sloupců a převést[11] užitím Levi-Civitových tenzorů ε na tvar

$$\det A_{uv} = -\frac{1}{2} \sum_{a,b,c,d,e,f,g} x_a x'_b x''_c \varepsilon_{ade} \varepsilon_{bfu} \varepsilon_{cgv} \det \begin{bmatrix} p_{d1} & p_{d2} & p_{d3} & p_{d4} \\ p_{e1} & p_{e2} & p_{e3} & p_{e4} \\ p'_{f1} & p'_{f2} & p'_{f3} & p'_{f4} \\ p''_{g1} & p''_{g2} & p''_{g3} & p''_{g4} \end{bmatrix} = 0, \quad (4.33)$$

kde $a, b, c, d, e, f = 1, \dots, 3$. Dále zavedme tenzor

$$\mathcal{T}_a^{fg} = \frac{1}{2} \sum_{d,e} \varepsilon_{ade} \begin{bmatrix} p_{d1} & p_{d2} & p_{d3} & p_{d4} \\ p_{e1} & p_{e2} & p_{e3} & p_{e4} \\ p'_{f1} & p'_{f2} & p'_{f3} & p'_{f4} \\ p''_{g1} & p''_{g2} & p''_{g3} & p''_{g4} \end{bmatrix}, \quad d, e = 1, \dots, 3 \quad (4.34)$$

Dosažením tohoto tenzoru do rovnice (4.33) dostáváme

$$\sum_{a,b,c,f,g} x_a x'_b x''_c \varepsilon_{bfu} \varepsilon_{cgv} \mathcal{T}_a^{fg} = 0, \quad a, b, c, f, g = 1, \dots, 3 \quad (4.35)$$

Tato rovnice po změně indexů odpovídá rovnici (4.13). Tenzor \mathcal{T}_a^{fg} je tedy trifokální tenzor. Pokud je matice kamery P tvaru

$$P = [I|\mathbf{0}], \quad (4.36)$$

pak se výpočet tenzoru zjednoduší na

$$\mathcal{T}_a^{fg} = p'_{fa} \cdot p''_{g4} - p'_{f4} \cdot p''_{ga} \quad (4.37)$$

a my tak získáme trifokální tenzor.

4.2. Výpočet trifokálního tenzoru užitím Carlsson-Weinshallové duality

Na závěr této části si uveďme příklad metody, která umožňuje výpočet geometrie tří pohledů užitím nalezených korespondencí mezi trojicemi bodů v obrazech.

Výpočet trifokálního tenzoru sestává z několika kroků. K výpočtu potřebujeme šest trojic odpovídajících si bodů tak, aby v žádném z pohledů neležely tři na jedné přímce. Níže popsaná metoda využívá tzv. *Carlsson-Weinshallové duality*.

Tato dualita spočívá v možné záměně mezi body pozorovanými více kamerami a středy kamer. Pokud je provedena tato záměna a následně provedena rekonstrukce, lze nalezené matice kamer převést na body v prostoru a získat tak rekonstrukci scény před záměnou. Tuto skutečnost nezávisle na sobě popsali Stefan Carlsson[4] a tým Daphny Weinshallové[24]. V této části bude popsáno konkrétní využití této vlastnosti při rekonstrukci scény ze tří obrazů.

4.2.1. Algoritmus rekonstrukce

Ze všech nalezených odpovídacích si trojic bodů je pro tento algoritmus nutné vybrat šest trojic. V tomto případě budeme značit příslušnost bodu k pohledu horním indexem a příslušnost k bodu scény spodním indexem. Například bod \mathbf{x}_2^1 leží v prvním pohledu a je obrazem bodu \mathbf{X}_2 .

Z těchto šesti trojic zvolíme čtyři trojice tak, aby žádné tři body neležely na jedné přímce v žádném z pohledů. Tyto body označíme spodním indexem tak, aby příslušely k bodům scény $\mathbf{X}_3, \dots, \mathbf{X}_6$. Zbylým dvěma trojicím přiřadíme spodní index 1 a 2.

Dalším krokem je nalezení transformace T^i pro i -tý pohled tak, aby se body $\mathbf{x}_3^i, \mathbf{x}_4^i, \mathbf{x}_5^i, \mathbf{x}_6^i$ zobrazily postupně na body geometrické báze $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ prostoru \mathbb{P}_2 tvaru

$$\begin{aligned} \mathbf{e}_1 &= (1, 0, 0)^\top \\ \mathbf{e}_2 &= (0, 1, 0)^\top \\ \mathbf{e}_3 &= (0, 0, 1)^\top \\ \mathbf{e}_4 &= (1, 1, 1)^\top. \end{aligned} \quad (4.38)$$

Toto nalezené zobrazení T^i se pak aplikuje na všech šest bodů v i -tém pohledu (obr. 4.2). Kromě bodů \mathbf{e}_k tak ještě získáme body $\bar{\mathbf{x}}_1^i = T^i \mathbf{x}_1^i$ a $\bar{\mathbf{x}}_2^i = T^i \mathbf{x}_2^i$. Tento krok provedeme pro všechny tři pohledy.

Nyní využijeme Carlssonovy duality. Body, které náležely k jednomu pohledu budou nyní náležet k jednomu bodu scény a naopak. Body $\hat{\mathbf{x}}_j^i$ jsou body před dualizací a $\bar{\mathbf{x}}_j^i$ značí body získané dualizací. Dualizací pak rozumíme záměnu $\hat{\mathbf{x}}_i^j = \bar{\mathbf{x}}_j^i$, $i = 1, 2, 3$, $j = 1, 2$. Zároveň odstraníme body \mathbf{e}_k . Po zdualizování ke každému pohledu opět přidáme body $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ (obr. 4.3).

Dostáváme tak 7 dvojic bodů ve dvou pohledech. Tento případ lze řešit například 7-bodovou metodou pro výpočet fundamentální matice (kapitola 3.2.1). Na takto získané body scény lze aplikovat transformaci, která body $\hat{\mathbf{X}}_3, \hat{\mathbf{X}}_4, \hat{\mathbf{X}}_5, \hat{\mathbf{X}}_6$ zobrazí na body $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4$ geometrické báze prostoru \mathbb{P}_3 tvaru

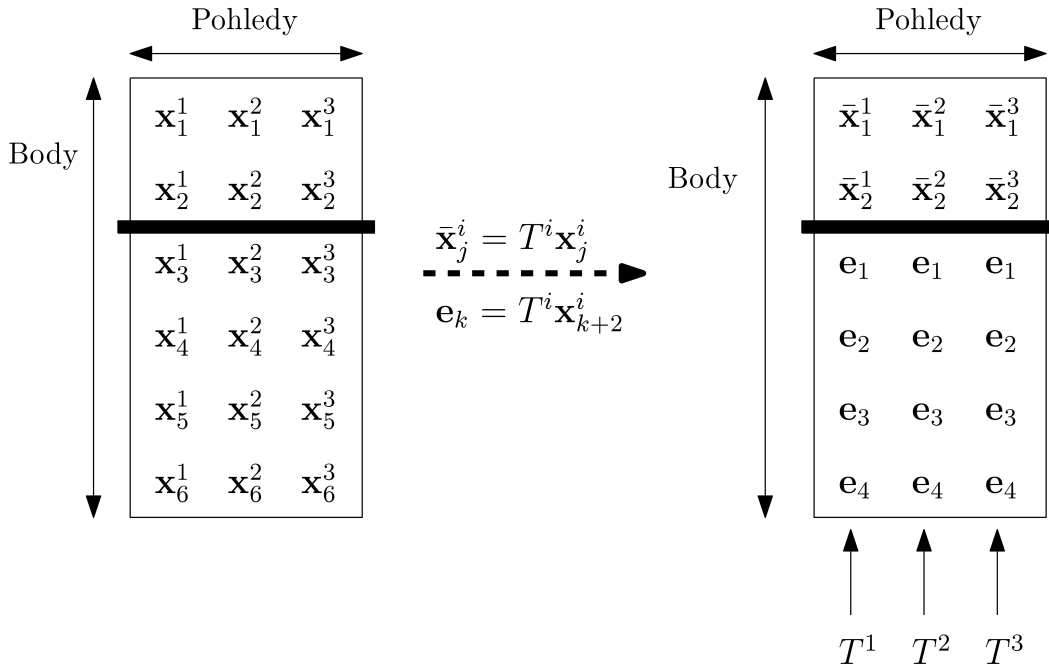
$$\begin{aligned} \mathbf{E}_1 &= (1, 0, 0, 0)^\top \\ \mathbf{E}_2 &= (0, 1, 0, 0)^\top \\ \mathbf{E}_3 &= (0, 0, 1, 0)^\top \\ \mathbf{E}_4 &= (0, 0, 0, 1)^\top. \end{aligned} \tag{4.39}$$

Tato transformace nemá vliv na výslednou rekonstrukci. V dalším kroku proto budeme hledat příslušné matice kamer \hat{P}^1, \hat{P}^2 , které zobrazí body \mathbf{E}_k na \mathbf{e}_k . Platí

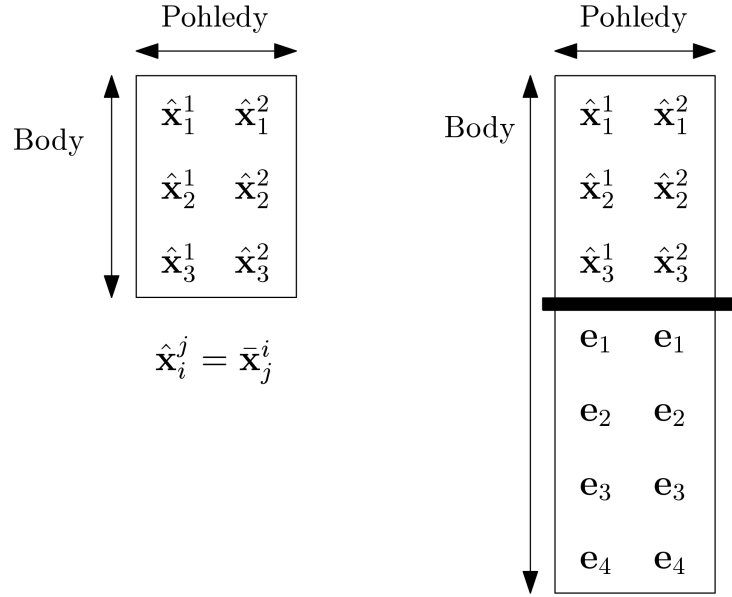
$$\hat{\mathbf{x}}_i^j = \hat{P}^j \hat{\mathbf{X}}_i \tag{4.40}$$

a tedy zřejmě

$$\mathbf{e}_k = \hat{P}^j \mathbf{E}_k, k = 1, \dots, 4. \tag{4.41}$$



Obrázek 4.2: Transformace bodů



Obrázek 4.3: Použití Carlssonovy duality

Lze proto odvodit, že hledané matice kamer jsou tvaru

$$\hat{P}^j = \begin{bmatrix} a_j & 0 & 0 & d_j \\ 0 & b_j & 0 & d_j \\ 0 & 0 & c_j & d_j \end{bmatrix}, j = 1, 2. \quad (4.42)$$

Určili jsme tedy, které prvky těchto matic budou nulové. K vypočtení matic kamer však musíme znát, čemu se rovnají jejich nenulové prvky. Pro jejich určení využijeme vztahů mezi dvěma odpovídajícími si body ve dvou obrazech. Místo obecné fundamentální matice si můžeme zavést tzv. *redukovanou fundamentální matici* \hat{F} . Jedná se o fundamentální matici, která má na hlavní diagonále pouze nuly. Pro dvojici bodů a obecnou fundamentální matici platí (3.6) na str. 24. V našem případě do této rovnice dosadíme dvojice bodů \mathbf{e}_k a dostaneme tak

$$\mathbf{e}_k^\top \hat{F} \mathbf{e}_k = 0, k = 1, \dots, 4. \quad (4.43)$$

Po dosazení bodů $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ do této rovnice dostáváme

$$\begin{aligned} (1, 0, 0) \cdot \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} &= 0 \\ (0, 1, 0) \cdot \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} &= 0 \\ (0, 0, 1) \cdot \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} &= 0. \end{aligned} \quad (4.44)$$

Dostáváme tedy, že $f_{11} = f_{22} = f_{33} = 0$. Dále do rovnice (4.43) dosadíme bod \mathbf{e}_4 . Dostaneme tak rovnici

$$(1, 1, 1) \cdot \begin{pmatrix} 0 & f_{12} & f_{13} \\ f_{21} & 0 & f_{23} \\ f_{31} & f_{32} & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 0. \quad (4.45)$$

Vidíme tedy, že

$$f_{12} + f_{13} + f_{21} + f_{23} + f_{31} + f_{32} = 0 \quad (4.46)$$

Redukovaná matice je proto v našem případě tvaru

$$\hat{F} = \begin{bmatrix} 0 & f_{12} & f_{13} \\ f_{21} & 0 & f_{23} \\ f_{31} & -(f_{12} + f_{13} + f_{21} + f_{23} + f_{31}) & 0 \end{bmatrix}. \quad (4.47)$$

Máme pět neznámých f_{12} , f_{13} , f_{21} , f_{23} , f_{31} . Protože u fundamentální matice je důležitý pouze vzájemný poměr jejich prvků a ne jejich velikost, k jejich určení nám stačí čtyři rovnice. Použijeme rovnosti platící pro zbylé tři body a nulovost determinantu

$$\begin{aligned} \hat{\mathbf{x}}_1^{2\top} \hat{F} \hat{\mathbf{x}}_1^1 &= 0 \\ \hat{\mathbf{x}}_2^{2\top} \hat{F} \hat{\mathbf{x}}_2^1 &= 0 \\ \hat{\mathbf{x}}_3^{2\top} \hat{F} \hat{\mathbf{x}}_3^1 &= 0 \\ \det \hat{F} &= 0. \end{aligned} \quad (4.48)$$

Jejich užitím jsme schopni jednoznačně určit redukovanou fundamentální matici \hat{F} .

Po určení matice \hat{F} již můžeme určit matice kamer. Protože umístění a natočení druhé kamery je závislé na umístění a natočení první kamery, můžeme si matici první kamery zvolit a budeme hledat pouze matici druhé kamery. Budeme tedy předpokládat, že matice první kamery je

$$\hat{P}^1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (4.49)$$

Redukovanou fundamentální matici lze proto vyjádřit pomocí členů matice kamery \hat{P}^j

$$\hat{F} = \begin{bmatrix} 0 & b_2(d_2 - c_2) & -c_2(d_2 - b_2) \\ -a_2(d_2 - c_2) & 0 & c_2(d_2 - a_2) \\ a_2(d_2 - b_2) & -b_2(d_2 - a_2) & 0 \end{bmatrix}. \quad (4.50)$$

Redukovanou fundamentální matici jsme ovšem již určili a tak nyní můžeme určit i koeficienty a_2 , b_2 , c_2 , d_2 . Dostaneme tak členy matice kamery \hat{P}^2 .

Po nalezení matic obou kamer provedeme zpětnou dualizaci. Kamera matice tvaru

$$P = \begin{bmatrix} X & 0 & 0 & T \\ 0 & Y & 0 & T \\ 0 & 0 & Z & T \end{bmatrix}. \quad (4.51)$$

je duální s bodem $\mathbf{X} = (X, Y, Z, T)^\top$. Dostáváme tak šestici bodů, které tvoří výslednou scénu

$$\begin{aligned}\mathbf{X}_1 &= (1, 1, 1, 1)^\top \\ \mathbf{X}_2 &= (a_2, b_2, c_2, d_2)^\top \\ \mathbf{X}_3 &= \mathbf{E}_1 = (1, 0, 0, 0)^\top \\ \mathbf{X}_4 &= \mathbf{E}_2 = (0, 1, 0, 0)^\top \\ \mathbf{X}_5 &= \mathbf{E}_3 = (0, 0, 1, 0)^\top \\ \mathbf{X}_6 &= \mathbf{E}_4 = (0, 0, 0, 1)^\top.\end{aligned}\tag{4.52}$$

Známe tedy body scény i body obrazů. Z nich již můžeme určit kamery matic P^1 , P^2 a P^3 , které mají zcela obecný tvar. Matice lze získat způsobem popsaným v kapitole 3.5. Rekonstrukcí scény užitím těchto kamer bychom obecně nedosáhli metrické rekonstrukce, ale pouze rekonstrukce projektivní. Trifokální tenzor můžeme určit dle postupu v kapitole 4.1.4 na straně 33. K získání metrické rekonstrukce je nutné provést autokalibraci i přes znalost kalibrační matice. Možné postupy lze najít například v [11].

5. Detekce bodů metodou SURF

Aby bylo možné provést rekonstrukci tak, jak je popsáno v předchozích kapitolách, je nutné mít nalezené dvojice (případně trojice) vzájemně si odpovídajících bodů mezi zpracovávanými obrazy. Je tedy nutné použít algoritmus, který dokáže najít vhodné body v jednotlivých obrazech a následně z nich vytvořit dvojice (nebo trojice) tak, že tyto body si navzájem opravdu odpovídají (tzn. zachycují stejný bod scény).

Jeden z možných algoritmů, který umožňuje detekci významných bodů je algoritmus SURF (Speed-Up Robust Features)[2] uveřejněný v roce 2008. Tento algoritmus je používán pro detekci bodů v této diplomové práci, a proto zde bude stručně vysvětlen jeho princip. K jeho vysvětlení je nejprve nutné vyjádřit, co to vlastně je digitální obraz.

5.1. Matematický popis digitálního obrazu

Dnešní digitální fotoaparáty pořizují barevné fotografie. Ty lze jednoduše převést na černobílý obraz, který je mimo jiné použit i metodou SURF. Černobílý obraz je tedy definován následovně [8]:

Definice 5.1.1. Buď $R = \{0, 1, \dots, M - 1\} \times \{0, 1, \dots, N - 1\}$, $M, N \in \mathbb{N}$ a necht' $W = \{0, 1, \dots, w - 1\}$, $w \in \mathbb{N}$. Funkce

$$I(x, y) : R \rightarrow W$$

je nazývána *černobílý obraz*, případně *obraz*. M je *výška obrazu*, N je *šířka obrazu*. Prvky R jsou *pixely*. Hodnota I v pixelu (x, y) je *hodnota pixelu*. Hodnota w udává *dynamický rozsah obrazu*.

V obraze detekované body často představují nejrůznější struktury scény (např. hrany, rohy, ...). Velikost těchto struktur v obraze se může měnit v závislosti na použitém fotografickém zařízení a vzdálenosti fotoaparátu od těchto struktur při pořizování obrazu. Aby metoda metody pro detekci bodů mohly zahrnout tyto změny, musí se během detekce často měnit velikost fotografií, což je obecně časově náročný proces. Metoda SURF tento problém obchází užitím integrálního obrazu [6].

Definice 5.1.2. Necht' funkce $I(x, y) : R \rightarrow W$ je obraz. Dále $W' = \{0, 1, \dots, w'\}$, $w' \in \mathbb{N}$. Pak funkce

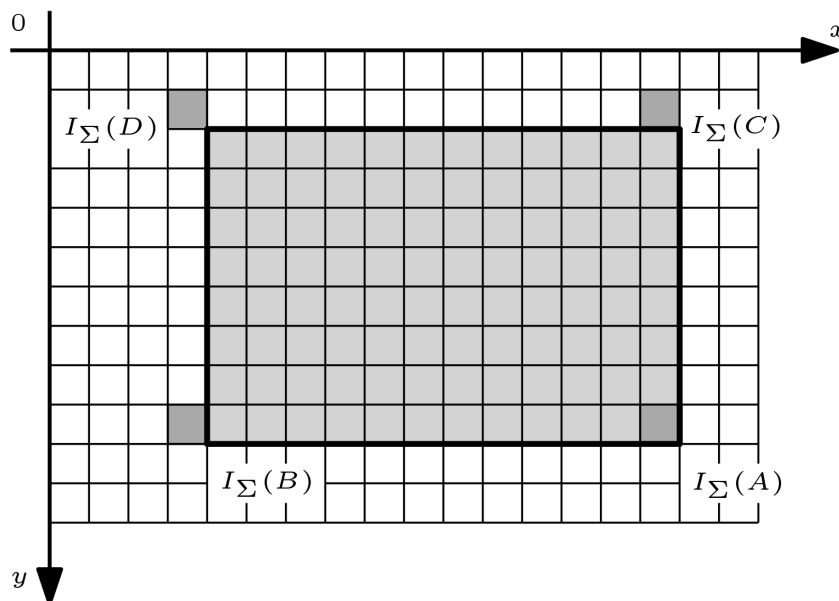
$$I_{\Sigma}(x, y) : R \rightarrow W' = \sum_{i=0}^x \sum_{j=0}^y I(i, j)$$

se nazývá *integrální obraz*.

Integrální obraz umožňuje pomocí tří operací určit součet hodnot pixelů uvnitř libovolné obdélníkové oblasti.

$$I_{ABCD} = I_{\Sigma}(A) - I_{\Sigma}(B) - I_{\Sigma}(C) + I_{\Sigma}(D), \quad (5.1)$$

kde $I_{\Sigma}(A)$, $I_{\Sigma}(B)$, $I_{\Sigma}(C)$, $I_{\Sigma}(D)$ jsou hodnoty pixelů integrálního obrazu v bodech A , B , C , D dle obr. 5.1. Rychlost výpočtu tak nezávisí na velikosti oblasti.



Obrázek 5.1: Součet hodnot pixelů v šedé ohraničené oblasti se dá určit pomocí sčítání a odčítání hodnot tmavě šedých pixelů příslušného integrálního obrazu.

5.2. Detekce významných bodů

Prvním krokem metody SURF je nalezení významných bodů. K jejich nalezení se mimo jiné využívá diskrétní konvoluce [7].

Definice 5.2.1. Buď $I(x, y)$ obraz a $h(x, y) : \{-k, -k + 1, \dots, k\}^2 \rightarrow \mathbb{C}$, $k \in \mathbb{N}$. Pak

$$f(x, y) = I(x, y) \star h(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x - i, y - j) \cdot h(i, j).$$

se nazývá *diskrétní dvourozměrná konvoluce*. Funkci h nazveme konvolučním jádrem.

Konvoluční jádro, které budeme používat, je *2D Gaussova funkce* G , která aproximuje normální rozdělení.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (5.2)$$

Směrodatná odchylka σ bude dále nazývána *měřítkem*. Dále mějme Hessianovu matici H (matice druhých parciálních derivací)

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}, \quad (5.3)$$

kde $L_{xx}(x, y, \sigma)$ je konvoluce vstupního obrazu $I(x, y)$ s druhou parciální derivací podle x Gaussovy funkce $G(x, y, \sigma)$ s měřítkem σ v bodě (x, y) .

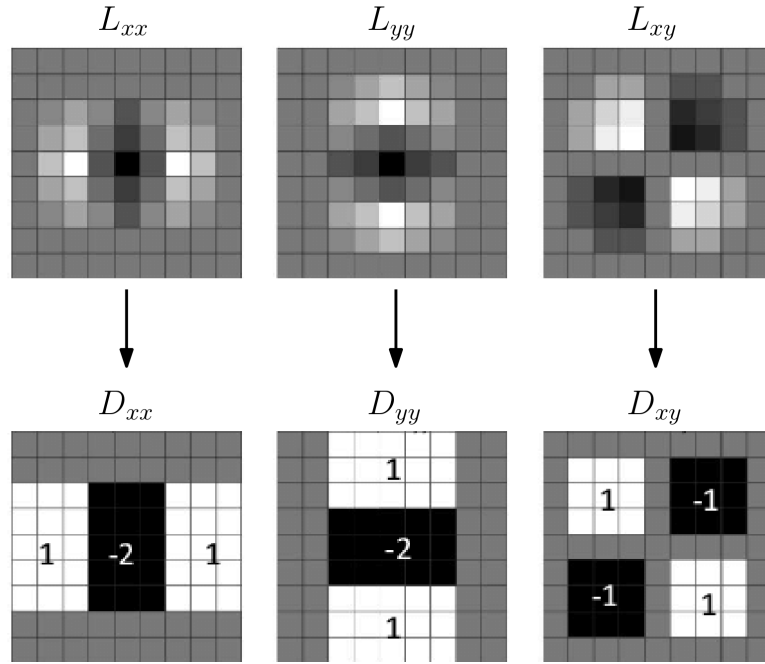
$$L_{xx}(x, y, \sigma) = I(x, y) \star \frac{\partial^2}{\partial x^2} G(x, y, \sigma) \quad (5.4)$$

Obdobně získáme funkce L_{xy} a L_{yy} .

Aby došlo k urychlení výpočtu, aproximujeme funkce L_{xx} , L_{xy} a L_{yy} pomocí tzv. *obdélníkových funkcí* D_{xx} , D_{xy} a D_{yy} . Ty vytvoříme konvolucí obrazu s tzv. *obdélníkovými filtry* B_{xx} , B_{xy} , B_{yy} které aproximují druhé parciální derivace Gaussovy funkce (obr. 5.2).

$$D_{xx}(x, y, \sigma) = I(x, y) \star B_{xx}(x, y, \sigma) \quad (5.5)$$

Ke zrychlení celého výpočtu dochází užitím integrálního obrazu. Jak bylo ukázáno výše, integrální obraz nám umožňuje velice rychle určit součet hodnot pixelů v libovolné obdélníkové oblasti. Při konvoluci s obdélníkovými funkcemi tedy postačuje pracovat s tímto součtem a nikoli s hodnotami jednotlivých pixelů.



Obrázek 5.2: Aproximace druhých derivací Gaussovy funkce obdélníkovými filtry, v tomto případě o velikosti 9x9. Převzato z [25], upraveno.

Nyní již lze jednoduše spočítat hodnotu determinantu H_{det} aproximované Hessiany matice H_{apr}

$$H_{det}(x, y, \sigma) = \det H_{apr}(x, y, \sigma) = D_{xx}(x, y, \sigma)D_{yy}(x, y, \sigma) - (wD_{xy}(x, y, \sigma))^2, \quad (5.6)$$

kde $w = 0,9$ je váhový koeficient, který slouží k zpřesnění výpočtu po aproximaci obdélníkovými funkcemi. Jeho hodnota byla určena experimentálně.

Nyní sestojíme tzv. *scale-space*. Scale-space se dělí na oktávy a jednotlivé oktávy se skládají z oktákových vrstev. Každá vrstva má dané měřítko σ a její body jsou reprezentovány vypočtenými determinanty $H_{det}(x, y, \sigma)$. Zpravidla postačují čtyři oktávy o čtyřech vrstvách. Volbu oktáv a vrstev lze vidět v tab. 5.1.

Posledním krokem je už samotné nalezení významných bodů v jednotlivých vrstvách. Každý bod je porovnán se svými osmi sousedy ve stejné vrstvě, s devíti sousedy ležícími ve vrstvě nad ním a s devíti sousedy ležícími ve vrstvě pod ním. Pokud je daný bod jediným maximem ze všech 27 prvků, pak jej prohlásíme za významný bod. Před porovnáním bodů z různých vrstev je nutné hodnoty vynásobit vhodnými konstantami tak, aby došlo k odstranění vlivu rozdílné velikosti obdélníkových filtrů pro každou vrstvu.

oktáva	vrstva	měřítko σ	velikost filtru
I	1	1, 2	9x9
	2	2, 0	15x15
	3	2, 8	21x21
	4	3, 6	27x27
II	1	2, 0	15x15
	2	3, 6	27x27
	3	5, 2	39x39
	4	6, 8	51x51
III	1	3, 6	27x27
	2	6, 8	51x51
	3	10, 0	75x75
	4	13, 2	99x99
IV	1	6, 8	51x51
	2	13, 2	99x99
	3	19, 6	147x147
	4	26, 0	195x195

Tabulka 5.1: Velikost obdélníkových filtrů pro jednotlivé vrstvy scale-space. Převzato z [21].

5.3. Deskriptor

Způsob, jakým byly určeny významné body, zajišťuje, že tyto body jsou invariantní vůči posunutí a změně měřítka. Zbývá tedy zajistit jejich invarianci vůči otočení. Této invariance je dosaženo během vytvoření deskriptoru. Deskriptor je vektor, který popisuje bod a jeho nejbližší okolí. Pokud je vzdálenost dvou deskriptorů dostatečně malá, pak si příslušné body dvou obrazů odpovídají.

K sestrojení deskriptoru pro metodu SURF se používají *Haarovy vlnky* (anglicky Haar wavelets) W_x, W_y (obr. 5.3). Jedná se o aproximaci prvních derivací Gaussovy funkce ve směru os. Měřítko σ odpovídá oktávové vrstvě, ve které se nalézá významný bod (x, y) . Na měřítku je závislá velikost jádra Haarovy vlnky. Pro každý bod (x_v, y_v) v kruhovém okolí významného bodu o poloměru 6σ určíme odezvu (V_x, V_y) . Nejprve určíme odezvu pouze na základě konvoluce s Haarovými vlnkami

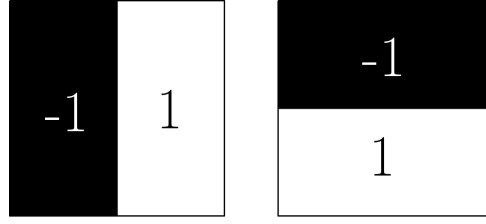
$$\begin{aligned} V'_x(x_v, y_v, \sigma) &= I(x_v, y_v) \star W_x(x_v, y_v, \sigma) \\ V'_y(x_v, y_v, \sigma) &= I(x_v, y_v) \star W_y(x_v, y_v, \sigma). \end{aligned} \quad (5.7)$$

Výpočet lze opět urychlit užitím integrálního obrazu. Následně získanou odezvu ještě vynásobíme Gaussovou funkcí se středem ve významném bodě

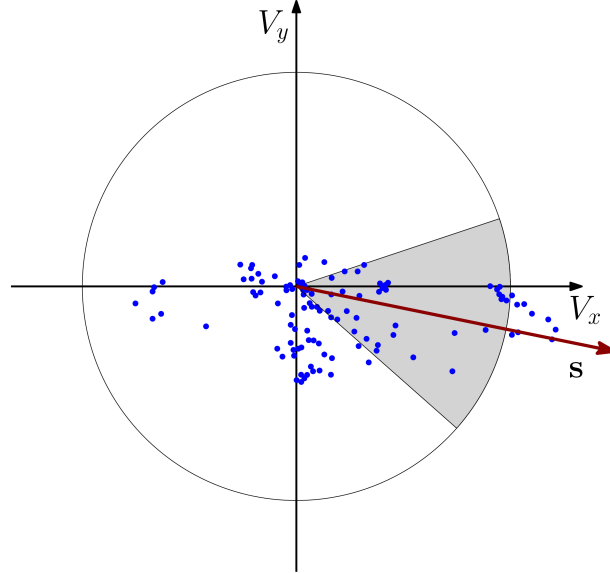
$$(V_x, V_y) = (V'_x, V'_y) \cdot G(x, y, 2\sigma). \quad (5.8)$$

Větší váhu tedy mají body blíže významnému bodu. Tyto odezvy představují vektory se středem v počátku.

Dále určíme výseč s úhlem $\frac{\pi}{3}$ tak, aby délka vektoru vzniklého součtem všech odezev v této výseči byla co největší. Tento nejdelší vektor označíme \mathbf{s} . Vektor \mathbf{s} udává orientaci významného bodu (obr. 5.4). Je tak zajištěna invariance vůči natočení.



Obrázek 5.3: Haarovy vlnky



Obrázek 5.4: Volba výseče a výsledná orientace \mathbf{s} významného bodu.

Nyní již můžeme sestavit samotný deskriptor. Nejprve uvažujme čtverec se středem ve významném bodu o délce strany 20σ , který je orientován ve směru vektoru \mathbf{s} , tj. má s ním rovnoběžné dvě strany. Tento čtverec dále rozdělíme na 16 menších stejných čtverců.

V každém z těchto čtverců zvolíme 5×5 pravidelně rozmístěných bodů (x_i, y_i) a určíme pro ně odezvu na Haarovy vlnky

$$\begin{aligned} d'_x(x_i, y_i, \sigma) &= I(x_i, y_i) \star V_x(x_i, y_i, 2\sigma) \\ d'_y(x_i, y_i, \sigma) &= I(x_i, y_i) \star V_y(x_i, y_i, 2\sigma), \end{aligned} \quad (5.9)$$

přičemž natočení Haarových vlnek je dáno orientací vektoru \mathbf{s} . Nelze již tedy použít integrální obraz. Následně odezvu opět vynásobíme Gaussovou funkcí se středem ve významném bodě.

$$(d_x, d_y) = (d'_x, d'_y) \cdot G(x, y, \frac{33}{10}\sigma). \quad (5.10)$$

Poté v každém čtverci určíme určíme

$$\mathbf{v} = (\sum d_x, \sum d_y, |\sum d_x|, |\sum d_y|). \quad (5.11)$$

Zřetězením vektorů \mathbf{v} dostáváme vektor délky 64. Tento vektor převedeme na vektor jednotkový a dostáváme tak deskriptor významného bodu.

6. Podrobná rekonstrukce scény

Aby bylo možné provést co nejpodrobnější rekonstrukci scény, je nutné mít dostatek nalezených dvojic vzájemně si odpovídajících bodů. Užitím metody SURF obecně nelze získat dostatečně husté mračno bodů. Využijeme ji proto pouze pro rekonstrukci vzájemného umístění a natočení kamer. Pro získání dalších bodů je nutné rektifikovat dvojici obrazů a následně pro tuto dvojici určit tzv. *mapu disparity*. Ta nám umožní provést mnohem podrobnější bodovou rekonstrukci.

6.1. Rektifikace obrazu

Rektifikace obrazu je transformace, kterou je možné provést po nalezení příslušné fundamentální matice. Umožňuje snadnější nalezení odpovídajících si bodů na obou snímcích.

Princip spočívá v nalezení vhodné projektivní transformace každého obrazu do společné roviny tak, aby epipolární přímky na obou obrazech byly rovnoběžné s osou x a odpovídající si body vždy měly stejnou souřadnici y . Pokud poté zvolíme libovolný bod v prvním snímku, pak odpovídající bod v druhém snímku (pokud existuje), má stejnou souřadnici y .

Rektifikace se aplikuje pouze na obraz s minimální nebo žádnou distorzí, neboť ta má vliv na přesnost rektifikace. Rektifikaci je nutné provést pro každou dvojici obrazů zvlášť.

6.1.1. Transformace prvního obrazu

Prvním krokem je nalezení transformace prvního obrazu I . Výsledná transformace obrazu vzniká složením tří transformací. První z nich je zcela libovolná translace T . Poté je nutné otočit obraz tak, aby epipól \mathbf{e} ležel na ose x obrazu. Druhou transformací je tedy rotace R . Poslední transformací se snažíme docílit toho, aby epipolární přímky obrazu byly navzájem rovnoběžné. Protože tyto přímky se navzájem protínají v epipólu, hledáme transformaci G , která změní epipól z vlastního bodu na nevlastní bod.

Po provedení transformací R a T má epipól souřadnice $\mathbf{e} = (f, 0, 1)^T$. Vhodnou transformací je

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-1}{f} & 0 & 1 \end{bmatrix}. \quad (6.1)$$

Tato transformace změní libovolný bod $(x, y, 1)^T$ na bod $(x, y, 1 - \frac{x}{f})^T$. Je tedy zřejmé, že počátek zobrazí opět na počátek.

Výsledná hledaná transformace obrazu je

$$H = GRT. \quad (6.2)$$

Nyní zbývá najít vhodnou transformaci druhého obrazu I' .

6.1.2. Transformace druhého obrazu

Mějme obrazy I a I' . Předpokládejme, že jsme již našli transformaci H prvního obrazu I . Hledáme tedy transformaci H' druhého obrazu I' s požadavkem, aby každé dva odpovídající si body \mathbf{x} , \mathbf{x}' v obrazech I , I' měli po transformaci stejnou souřadnici y .

Pro hledanou transformaci H' platí

$$H' = H_A H M, \quad (6.3)$$

kde M je dáno vztahem

$$M = \left([e']_{\times} \right)^{-1} F' \quad (6.4)$$

a H_A je matice, která představuje afinní transformaci a je tvaru

$$H_A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.5)$$

kde a , b , c jsou v tomto případě koeficienty, které získáme užitím metody nejmenších čtverců při minimalizaci sumy

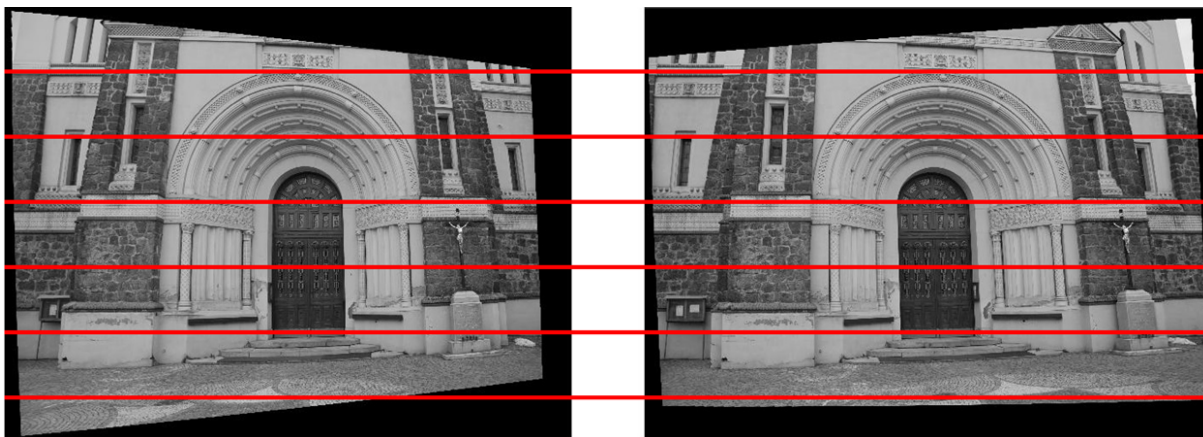
$$\sum_i (a\tilde{x}_i' + b\tilde{y}_i' + c - \tilde{x}_i)^2, \quad (6.6)$$

přičemž

$$\begin{aligned} \tilde{\mathbf{x}}_i &= (\tilde{x}_i, \tilde{y}_i, 1)^\top = H \mathbf{x}_i \\ \tilde{\mathbf{x}}_i' &= (\tilde{x}_i', \tilde{y}_i', 1)^\top = H M \mathbf{x}_i' \end{aligned} \quad (6.7)$$

a \mathbf{x}_i , \mathbf{x}_i' je i -tá dvojice odpovídajících si bodů. Afinní transformace H_A tedy upraví obraz I' tak, aby při vzájemném proložení obou obrazů rozdíl obrazových souřadnic $y_i - y_i'$ každé dvojice odpovídajících si bodů byl co nejmenší.

Možný výsledek po transformaci obou obrazů lze vidět na obr. 6.1.



Obrázek 6.1: Výsledná rektifikace dvojice obrazů včetně několika epipolárních přímek.

6.2. Disparita

Nalezené transformace H a H' promítnou obrazy I , I' do společné roviny tak, že se obrazy obvykle částečně překrývají (viz. obr. 6.2). Získáme tak rektifikované obrazy

$$\begin{aligned} I_r &= H I \\ I_r' &= H' I' \end{aligned} \quad (6.8)$$

Disparita bodu prvního rektifikovaného obrazu I_r pak udává vzdálenost mezi tímto bodem a příslušným odpovídajícím bodem druhého rektifikovaného obrazu I'_r , přičemž oba dva body mají stejnou souřadnici y . Disparita je tedy určena pouze rozdílem x-ových souřadnic. Uvažujme odpovídající si body ležící v rektifikovaných obrazech $\mathbf{x}_r = (x_r, y_r, 1)$, $\mathbf{x}'_r = (x'_r, y'_r, 1)$. Pak disparitu d bodu \mathbf{x}_r určíme jako

$$d(\mathbf{x}_r) = x'_r - x_r \quad (6.9)$$

Abychom mohli pro bod \mathbf{x}_r v prvním obraze určit jeho disparitu, musíme nalézt jemu odpovídající bod \mathbf{x}'_r v druhém obraze. Jednou z jednodušších metod, kterou lze použít a která bude dále blíže popsána, je metoda SSD. Poměrně široký přehled dalších metod lze nalézt v [22].



Obrázek 6.2: Rektifikace obrazů do společné roviny.

6.2.1. Metoda SSD

SSD je zkratka pro *Sum of Square Differences*. Pro metodu je nutné zvolit parametr $k = 0, 1, 2, \dots$ udávající velikost okna. Čím menší bude velikost okna, tím je výsledek je detailnější. Větší velikost okna naopak přispívá k potlačení chyb. Toto okno pro pixel (x, y) je matice $W_k(x, y)$ typu $(2k + 1) \times (2k + 1)$. Jejimi prvky jsou hodnoty pixelů $I(u, v)$ z jeho okolí

$$w_{kij}(x, y) = I(x - k + i - 1, y - k + j - 1), \quad i, j = 1, \dots, 2k + 1 \quad (6.10)$$

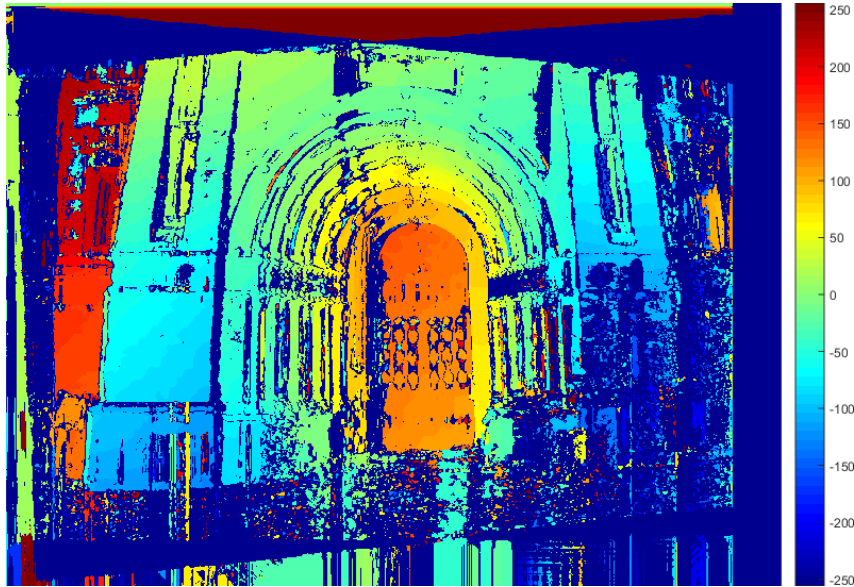
Mějme pixel (x, y) v prvním obraze a pixel (x', y') v druhém obraze s okny $W_k(x, y)$ a $W_k(x', y')$. Pak podobnost C_k těchto dvou pixelů pomocí metody SSD vyjádříme jako

$$C_k(x, y, x', y') = \sum_{i=1}^{2k+1} \sum_{j=1}^{2k+1} \left(w_{kij}(x, y) - w_{kij}(x', y') \right)^2. \quad (6.11)$$

Čím nižší je hodnota C_k , tím jsou si dva pixely podobnější.

V případě rektifikovaných obrazů tedy stačí pro každý pixel v prvním obraze určit jeho podobnost s pixely v druhém obraze se stejnou souřadnicí y a vybrat z nich ten s nejnižší hodnotou C_k . Tento pixel prohlásíme za odpovídající k pixelu z prvního obrazu.

Pokud tento postup provedeme pro každý pixel z prvního obrazu, dostaneme tak tzv. *mapu disparity* – ta vyjadřuje hodnotu disparity pro každý pixel prvního obrazu, viz obr. 6.3.



Obrázek 6.3: Mapa disparity pro rektifikované obrazy.

7. Implementace popsaných metod

Vybraná teorie popsaná v předchozích kapitolách byla užita k vytvoření aplikace využívající grafické uživatelské prostředí programu *MATLAB*. Nejprve bude popsáno prostředí aplikace, princip rekonstrukce, vstupy a výstupy. V druhé části 7.2 této kapitoly pak budou uvedeny výsledky.

7.1. Popis aplikace

Aplikace je rozdělena do tří navzájem propojených oken. Hlavní okno *main.fig* se jako jediné nachází v hlavním adresáři a slouží ke spuštění celé aplikace. Toto okno slouží k ukázce postupu rekonstrukce scén ze dvou obrazů. Z tohoto okna se lze za určitých podmínek přepnout na zbylá dvě okna. Okno *plot3d.fig* slouží k vykreslení výsledné scény. Okno *multi.fig* je použito k rekonstrukci scény z více fotografií.

Aplikace ke svému spuštění vyžaduje program *MATLAB* a toolboxy *Image Processing Toolbox*¹ a *Computer Vision Toolbox*². *Computer Vision Toolbox* je použit k odstranění distorze, k detekci bodů pomocí metody SURF a k výpočtu disparity. Tyto funkce byly v práci použity z důvodu jejich dobré optimalizace.

Při vyhodnocování výsledků této práce byla použita knihovna softwaru programu *MATLAB*: *mexopencv*, která je volně ke stažení (viz příloha A). Tato knihovna umožňuje použití knihovny *OpenCV*³, jež zahrnuje mnoho algoritmů z oblasti počítačového vidění[3]. *OpenCV* je napsána v jazyce C++ a také je volně dostupná. Při použití knihovny *mexopencv* není vyžadován *Computer Vision Toolbox*. Aplikace umí používat oba přístupy a uživatel má možnost volby.

7.1.1. Rekonstrukce ze dvou obrazů

Rekonstrukci ze dvou obrazů je prováděna v hlavním okně aplikace *main.fig*. Náhled tohoto okna je na obr. 7.1. Rozhraní umožňuje načtení vstupů a následné postupné spouštění jednotlivých kroků rekonstrukce uživatelem. Uživatel může některé kroky ovlivnit nastavením vybraných parametrů pomocí editovatelných textových oken. Celý průběh rekonstrukce je uživateli přiblížen pomocí vykreslování průběžných výsledků přímo v okně.

Vstupy

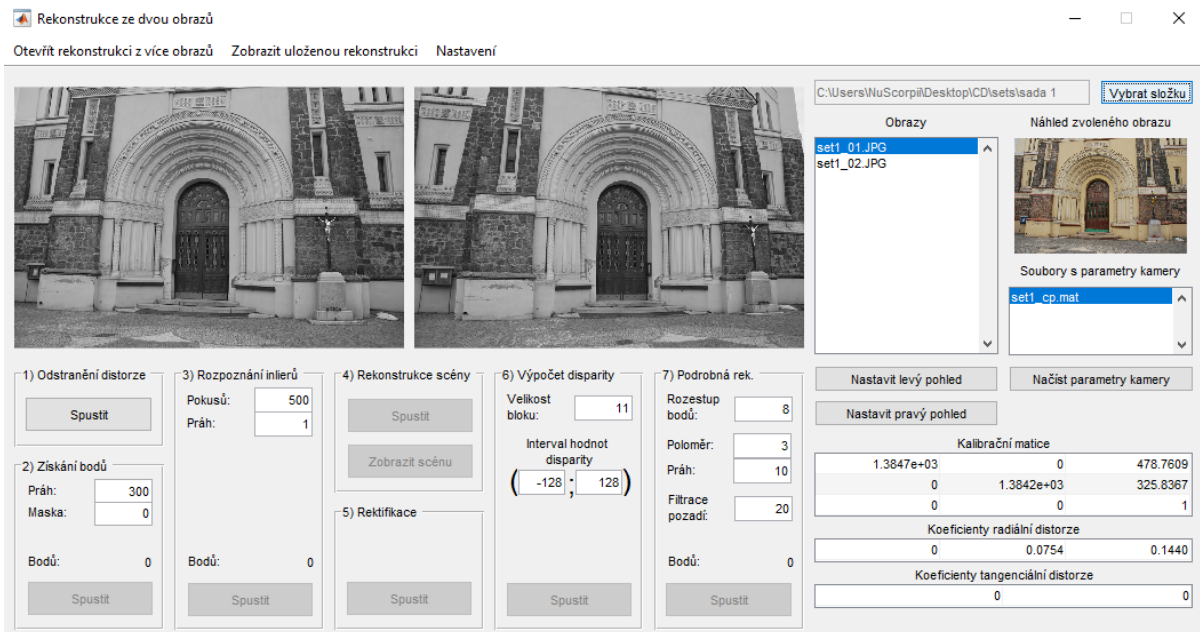
Od uživatele je vyžadováno zadání cesty ke složce, která obsahuje dvojici fotografií ve formátu JPEG nebo PNG. Pro správný chod aplikace musí mít tyto fotografie stejné rozměry. Dále by složka měla obsahovat soubor formátu MAT, který obsahuje data s následující strukturou:

- `camParams`, 1x1 struct
 - `RadialDistortion`, 1x3 matrix: vektor s koeficienty rad. distorze
 - `TangetialDistortion`, 1x2 matrix: vektor s koeficienty tang. distorze
 - `IntrinsicMatrix`, 3x3 matrix: kalibrační matice

¹<https://www.mathworks.com/help/images/index.html>

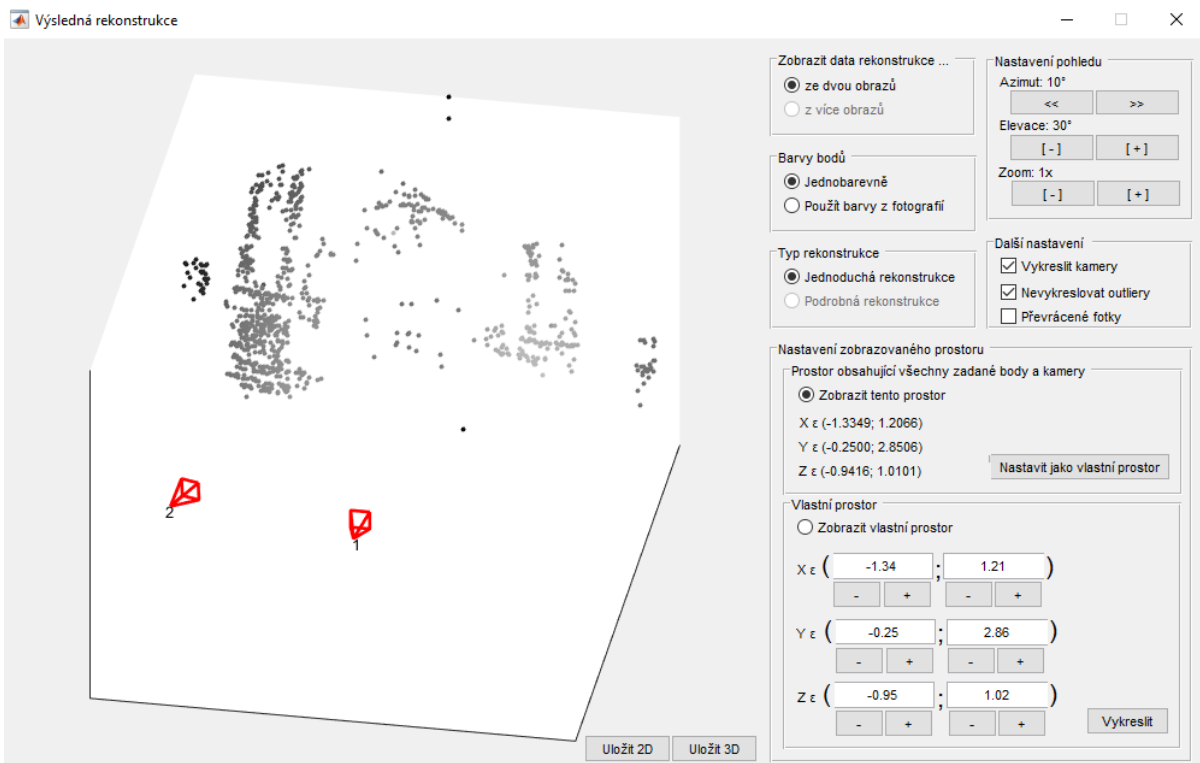
²<https://www.mathworks.com/help/vision/index.html>

³<https://opencv.org/>



Obrázek 7.1: Hlavní rozhraní aplikace *main.fig*.

Po zvolení složky aplikace sama projde všechny soubory v této složce a zobrazí uživateli seznam fotografií i seznam všech souborů formátu MAT, které mají výše zmíněnou strukturu. Uživatel poté nastaví levý a pravý pohled a načte parametry kamery ze zvoleného souboru. Poté je možné provést první krok rekonstrukce – odstranění distorze z obou fotografií. Po dokončení celého procesu má uživatel možnost si prohlédnout výslednou rekonstrukci v okně *3dplot.fig* (viz obr. 7.2).



Obrázek 7.2: Okno aplikace pro zobrazení výsledné rekonstrukce *plot3d.fig*.

Výstupy

Po dokončení rekonstrukce scény je možné výsledek uložit dvěma způsoby. Uživatel má možnost v okně *3dplot.fig* uložit aktuální náhled ve formátu JPG nebo uložit právě zobrazené body a pozice všech kamer do souboru formátu MAT s následující strukturou:

```
- scene, 1x1 struct
  - cameras, 1xn struct
    - matrix, 3x4 matrix:   matice kamery
    - rotation, 3x3 matrix: matice orientace kamery
    - translation, 1x3 matrix: vektor umístění kamery
  - points, 1x1 struct
    - location, 4xm matrix: homogenní souřadnice bodů
    - color, 3xm matrix:   RGB barvy bodů
```

Soubor v tomto formátu je možné později otevřít a zobrazit uloženou rekonstrukci volbou *Zobrazit uloženou rekonstrukci* v hlavním okně.

Postup rekonstrukce

Rekonstrukce ze dvou pohledů je rozdělena do úseků:

- I Odstranění distorze
- II Získání bodů
- III Rozpoznání inlierů
- IV Rekonstrukce scény
- V Rektifikace
- VI Výpočet disparity
- VII Podrobná rekonstrukce

Při **odstranění distorze** aplikace načte koeficienty distorzí ze zvoleného souboru s parametry kamery a provede úpravu zkoumaných fotografií (viz kapitola 2.2 na str. 19).

Během **získání bodů** jsou v obraze detekovány klíčové body pomocí metody SURF (kapitola 5 na str. 40). Nalezeným klíčovým bodům je přiřazen deskriptor. Díky porovnání deskriptorů jsou pak body spárovány mezi oběma obrazy.

Tento úsek vyžaduje od uživatele zadání hodnot pro práh a masku. Práh udává minimální velikost determinantu Hessianovy matice, pro který je bod ještě považován za klíčový. S klesající hodnotou je tedy detekováno více bodů, ale klesá spolehlivost správného spárování bodů mezi obrazy. Při příliš nízké hodnotě může dojít k pádu programu *MATLAB* z důvodu přetečení. Je tedy vhodné začít s vyšší hodnotou prahu (např. 500) a hodnotu prahu snížit při nízkém počtu bodů. Hodnota masky udává, šířku pásma v pixelech kolem hranic obrazu, ve kterém nechceme detekovat body.

Během **rozkognání inlierů** dojde k nalezení fundamentální matice, která splňuje rovnici (3.6) na str. 24 pro co největší počet detekovaných dvojic. Tyto dvojice jsou pak prohlášeny za inliery a ostatní dvojice jsou odstraněny. K nalezení fundamentální matice je použit 7-bodový algoritmus (část 3.2.1 na str. 24) v kombinaci s metodou RANSAC[10][20].

V této sekci uživatel nastavuje počet pokusů a práh. Při každém pokusu je zvoleno 7 náhodných bodů a následně určena fundamentální matice. Poté se zkoumá, kolik dvojic

bodů leží ve vzdálenosti menší než práh (uživatelé zadaná hodnota) od příslušných epipolárních přímek. Tyto dvojice jsou prohlášeny za inliery a spočítány. Výsledkem jsou dvojice a matice získané během pokusu s nejvyšším počtem inlierů.

Dalším krokem je **rekonstrukce scény**. Užitím kalibrační matice a fundamentální matice je určena matice esenciální a z ní jsou odvozeny matice obou kamer (část 3.3 na str. 26). Po nalezení kamer je již možná rekonstrukce bodů scény z nalezených dvojic bodů (část 3.4 na str. 28).

Tato rekonstrukce ovšem obvykle neobsahuje příliš mnoho bodů. Proto jsou dalšími kroky **rektifikace** (část 6.1 na str. 45) a **výpočet disparity** (část 6.2 na str. 46). Uživatel zde zadává velikost bloku a interval hodnot disparity. Velikost bloku představuje průměr čtvercového okolí pixelu, které je porovnáváno s okolím ostatních pixelů ležících na epipolární přímce v druhém obraze. Zadaná hodnota musí být liché celé číslo větší než čtyři. Interval hodnot disparity, pak omezuje počet prohledávaných pixelů na zadanou úsečku. Zadaný interval musí být dělitelný šestnácti.

Posledním krokem je **podrobná rekonstrukce**. Na odpovídající si body v rektifikovaných obrazech nalezené pomocí mapy disparity je aplikovaná zpětná transformace tak, abychom získaly odpovídající si body v původních obrazech.

S takto nalezenými body pracujeme stejně jako při rekonstrukci scény (bod č. 4), ale uživatel má řadu možností, jak ovlivnit výsledný počet a kvalitu rekonstrukce.

Údaj četnost bodů definuje vzdálenost dvou bodů v prvním obraze, které budou použity k rekonstrukci. Údaje poloměr a práh umožňují uživateli zamítnout body jejichž disparity se výrazně liší od průměrné disparity bodů v okolí (rozdíl nesmí být větší než práh, velikost okolí je dána poloměrem). Hodnota filtrace pozadí umožňuje zamítnout body, jejichž jas je menší než zadaná hodnota.

Jakmile uživatel provede rekonstrukci scény, zpřístupní se mu okno *plot3d.fig*, které mu umožní zobrazit a uložit rekonstrukci. Uživatel má možnost přepínat mezi rekonstrukcí získanou ze dvou obrazů a rekonstrukcí získanou z více obrazů. Dále má možnost volby mezi jednoduchou a podrobnou rekonstrukcí, případně mezi barevným a jednobarevným zobrazením bodů.

Dalšími možnostmi je nastavení zobrazovaného prostoru, nevykreslování příliš vzdálených bodů, vykreslení kamer, či překlopení scény v případě fotografií focených naležato. Uživatel má také kontrolu nad úhlem pohledu a přiblížením.

7.1.2. Implementace

V této části budou uvedeny dva konkrétní příklady kódů funkcí, které jsou implementovány v aplikaci a které využívají vzorců z předchozích kapitol. Tyto funkce byly vybrány k ukázce z důvodu jejich poměrné jednoduchosti a protože nevolají žádné další vlastní funkce. Dají se proto dobře porovnat s teoretickým postupem. Všechny ostatní kódy je možné nalézt na příloženém CD.

První uvedený kód se nachází v souboru `triang.m` a slouží k výpočtu bodů scény. Bod scény \mathbf{X}_i je řešením rovnice $A\mathbf{X}_i = \mathbf{0}$, kde A získáme užitím bodů obrazů \mathbf{x}_i , \mathbf{x}'_i a matic kamer P , P' .

$$A = \begin{pmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{pmatrix} \quad (3.38)$$

```

1 function [X] = triang(x1,x2,P1,P2)
2 % x1 - body v 1. obraze
3 % x2 - body v 2. obraze
4 % P1 - matice 1. kamery
5 % P2 - matice 2. kamery
6
7 X=zeros(4,size(x1,2));
8 for i=1:size(x1,2)
9     %sestaveni rovnice
10    A(1,:)=x1(1,i)*P1(3,:)-P1(1,:);
11    A(2,:)=x1(2,i)*P1(3,:)-P1(2,:);
12    A(3,:)=x2(1,i)*P2(3,:)-P2(1,:);
13    A(4,:)=x2(2,i)*P2(3,:)-P2(2,:);
14
15    % filtrace spatnych vysledku
16    x=isnan(A)+isinf(A);
17    x=max(max(x));
18    if(x~=1)
19        % reseni uzitim SVD rozkladu
20        [~,~,V]=svd(A);
21        X(:,i)=V(:,end);
22        X(:,i)=X(:,i)/(X(4,i));
23    else
24        X(:,i)=[0;0;0;1];
25    end
26 end

```

Druhý kód ze souboru `F_from_xx.m` implementuje 7 bodový algoritmus pro nalezení fundamentální matice F . Nejprve stručně připomeňme teoretický postup tohoto algoritmu (plně je vysvětlen v kapitole 3.2.1). Hledáme řešení rovnice

$$A\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}, \quad (3.11)$$

kde \mathbf{f} je tvořen prvky matice F . Při použití sedmi dvojic odpovídajících si bodů má matice A obecně hodnost rovnu sedmi. Řešením rovnic je množina vektorů tvaru

$$\mathbf{f} = \alpha \cdot \mathbf{f}_1 + (1 - \alpha) \cdot \mathbf{f}_2. \quad (3.13)$$

Pokud F_1 a F_2 jsou fundamentální matice odpovídající vektorům \mathbf{f}_1 , \mathbf{f}_2 pak pro hledanou matici F platí

$$F = \alpha \cdot F_1 + (1 - \alpha) \cdot F_2. \quad (3.14)$$

Dále využijeme toho, že matice F je singulární a tedy

$$\det(F) = 0. \quad (3.15)$$

Lze tedy sestavit kubickou rovnici a jejím vyřešením obdržet fundamentální matici F .

$$\det(\alpha \cdot F_1 + (1 - \alpha) \cdot F_2) = 0, \quad (3.16)$$

V aplikaci je tento algoritmus zpracován následovně:

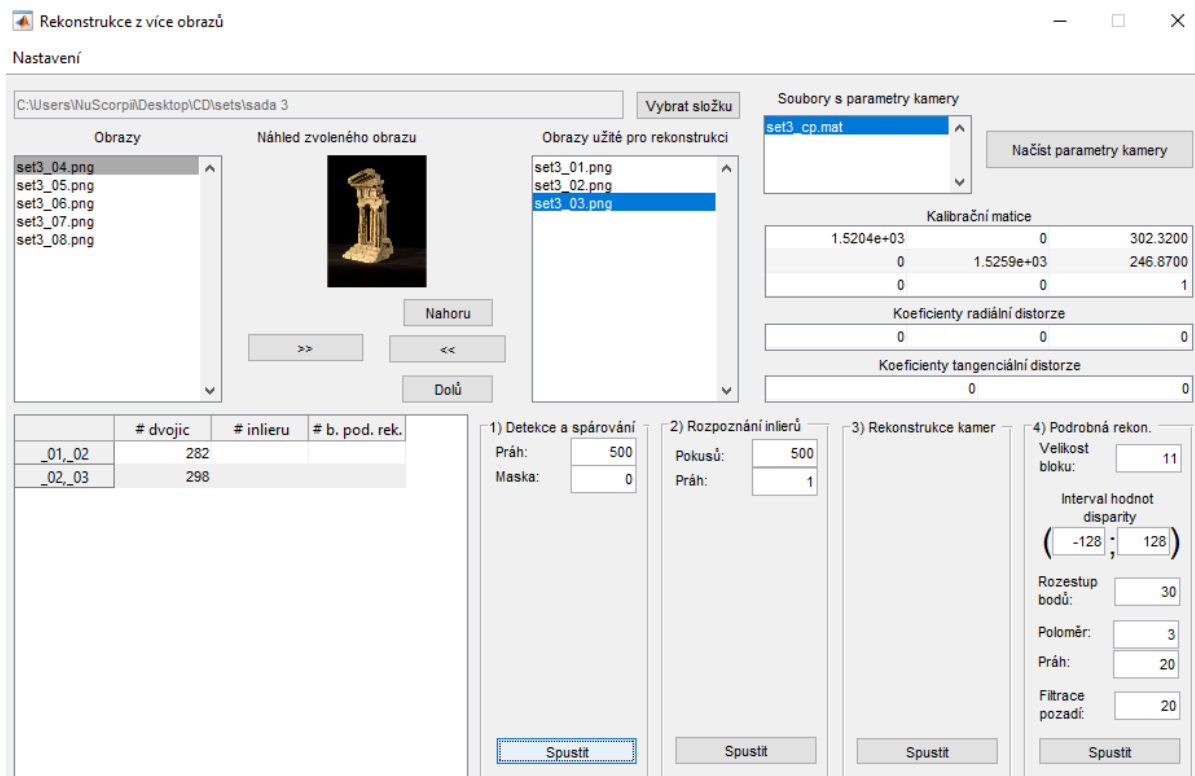
```

1 function F = F_from_xx(x1,x2)
2 % F – fundamentalni matice
3 % x1 – body v 1. pohledu
4 % x2 – body v 2. pohledu
5
6 % sestaveni matice A
7 A=zeros(7,9);
8 for i = 1:7
9     a = x2(:,i) * x1(:,i)';
10    A(i,:) = reshape(a',[9 1]);
11 end
12
13 % ziskani dvou reseni Af=0
14 f = null(A);
15 F1 = reshape(f(:,1),[3 3])';
16 F2 = reshape(f(:,2),[3 3])';
17
18 % reseni rovnice ( a*F{1} + (1-a)*F{2} ) = 0
19 FF{1}=F1; FF{2}=F2;
20
21 % predvypocet pro ziskani koeficientu polynomu
22 Det=zeros(2,2,2);
23 for i = 1:2
24     for j = 1:2
25         for k = 1:2
26             Det(i,j,k) = det([FF{i}(:,1), FF{j}(:,2), FF{k}(:,3)
27                                 ]);
28         end
29     end
30 end
31 % koreny rovnice
32 a = roots([ -Det(2,1,1)+Det(1,2,2)+Det(1,1,1)+Det(2,2,1)
33             +Det(2,1,2)-Det(1,2,1)-Det(1,1,2)-Det(2,2,2);...
34             Det(1,1,2)-2*Det(1,2,2)-2*Det(2,1,2)+Det(2,1,1)
35             -2*Det(2,2,1)+Det(1,2,1)+3*Det(2,2,2);...
36             Det(2,2,1)+Det(1,2,2)+Det(2,1,2)-3*Det(2,2,2);...
37             Det(2,2,2) ]);
38 % vyber realnych korenu
39 a = a(abs(imag(a))<10*eps);
40
41 % vypocet vysledne matice F
42 F=zeros(3,3,length(a));
43 for i =1:length(a)
44     F(:, :, i) = a(i)*F1 + (1-a(i))*F2;
45 end

```


7.1.3. Rekonstrukce z více obrazů

Rekonstrukce z více obrazů je prováděna ve vedlejším okně aplikace *multi.fig*. Náhled tohoto okna je na obr. 7.3. Toto okno je možné spustit z hlavního okna *main.fig* možností v horním menu: Otevřít rekonstrukci z více obrazů. Rozhraní okna umožňuje načtení vstupů a následné postupné spouštění jednotlivých kroků rekonstrukce uživatelem. Uživatel může některé kroky ovlivnit nastavením vybraných parametrů pomocí editovatelných textových oken.



Obrázek 7.3: Okno aplikace pro rekonstrukci z více pohledů *multi3d.fig*.

Vstupy a výstupy

Požadavky na vstupy jsou stejné jako u rekonstrukce ze dvou obrazů. Po načtení složky uživatel vybere obrazy pro rekonstrukci, přičemž při rekonstrukci záleží na jejich pořadí. Pořadí obrazů lze měnit pomocí tlačítek: Nahoru, Dolů.

Výsledky této rekonstrukce se vykreslují ve stejném okně jako výsledky rekonstrukce ze dvou obrazů a možnosti výstupů jsou stejné pro obě rekonstrukce.

Postup rekonstrukce

Celý proces rekonstrukce z více obrazů je přibližně stejný jako u rekonstrukce ze dvou obrazů. Uživatel má možnost rekonstrukci ovlivnit stejnými vstupy (práh, maska,...), proto zde už nebude opakován jejich význam.

Celý proces je rozdělen na čtyři úseky:

I Detekce a spárování

- II Rozpoznání inlierů
- III Rekonstrukce kamer
- IV Podrobná rekonstrukce

Detekce a spárování v sobě zahrnuje odstranění distorze, detekci klíčových bodů a jejich spárování mezi každými dvěma sousedními obrazy (pořadí obrazů může uživatel změnit před začátkem rekonstrukce).

Rozpoznání inlierů a výpočet fundamentální matice opět probíhá pro každou dvojici po sobě jdoucích obrazů.

Při **rekonstrukci kamer** nejdříve dojde k nalezení esenciální matice a následně matice kamer pro první dvojici obrazů. Poté je provedena rekonstrukce scény pro nalezené dvojice bodů v těchto dvou obrazech. V dalším kroku se vyberou všechny dvojice bodů druhé dvojice obrazů, které jsou již zrekonstruovány ve scéně (tj. součástí každé takové dvojice je bod ležící ve druhém obraze, který je spárován s bodem v prvním obraze i s bodem v třetím obraze).

Součástí vybraných dvojic jsou tak body ležící ve třetím obraze, ke kterým známe příslušné body scény – zrekonstruované z první dvojice obrazů. Užitím algoritmu DLT (část 3.5 na str. 29) pak obdržíme matici třetí kamery. Provedeme rekonstrukci scény pro druhou dvojici obrazů a celý postup zopakujeme pro získání matice čtvrté kamery. Obdobným způsobem pak získáme i matice dalších kamer.

Abychom získali natočení a umístění jednotlivých kamer, použijeme RQ rozklad (část 2.1.4 na str. 18).

Pro provedení **podrobné rekonstrukce** se pro každou dvojici obrazů provede rektifikace a následně se určí mapa disparity. Postup je již stejný jako pro rekonstrukci ze dvou obrazů.

Výše popsaný postup rekonstrukce scény z více obrazů s využitím algoritmu DLT se během programování a testování aplikace ukázal jako poměrně přesný. K jeho implementaci do aplikace byl použit kód[5]. Dalšími zvažovanými možnostmi bylo skládání dvojic, případně trojic obrazů. Skládání dvojic se ovšem ukázalo jako nepříliš spolehlivé, neboť je příliš zatíženo chybami v kalibrační matici, a skládání trojic je příliš náročné na implementaci a výpočetně složité.

7.2. Testování výsledků na reálných datech

Aplikace byla testována na dvou sadách pro rekonstrukci ze dvou fotografií (Sada 1, Sada 2) a na jedné sadě pro rekonstrukci z více fotografií (Sada 3). Sada 1 a Sada 2 jsou vlastnoručně pořízené. Pro tyto sady byla také provedena kalibrace fotoaparátu. Sadu 3 tvoří fotografie, které byly uvedeny v článku[23] a jsou volně dostupné včetně příslušné kalibrační matice. Výsledné rekonstrukce a vstupní fotografie všech tří sad lze vidět na obr. 7.7, obr. 7.8, obr. 7.9 a obr. 7.10 na konci této kapitoly na str. 62–65.

Testování aplikace probíhalo v prostředí programu *MATLAB* verze 2016b na operačním 64bitovém systému Microsoft Windows 10 Pro verze 10.0.17134. Počítač byl vybaven dvou-jádrovým procesorem Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2701 Mhz, operační paměť 8GB a SSD diskem. Během testování nebyl na počítači spuštěn žádný jiný program.

7.2.1. Rekonstrukce ze dvou obrazů

Fotografie v Sadě 1 a v Sadě 2 měly původní rozměr 3872x2592 pixelů. Rekonstrukce užitím takto velkých fotografií trvá často v řádech minut a navíc docházelo k občasným pádům programu *MATLAB* z důvodu nedostatečné paměti. Proto bylo nutné zmenšit rozměry fotografií v obou sadách.

Zmenšení vstupních fotografií

Podívejme se na dobu výpočtu v závislosti na velikosti vstupních obrazů. Pro analýzu byla použita Sada 1 vstupních obrazů. Rozměry obrazů byly zmenšeny na 75%, 50%, 25% a 10% původních rozměrů. Dále bylo nutné přepočítat kalibrační matice – ohniskové vzdálenosti i souřadnice principiálního bodu se také zmenšily na 75%, 50%, 25% a 10% původních hodnot. Poté byla každá takto upravená sada otestována na rekonstrukci ze dvou obrazů. Uživatelem zadávané hodnoty byly upravovány vzhledem k rozměrům fotografií, aby byl zamezen jejich vliv na výsledné měření. Pro hledání inlierů byl nastaven počet pokusů na 500 a práh na 1 pro všechna měření. Naměřené časy jsou uvedeny v Tabulce 7.1. Římské číslice značí úsek výpočtu:

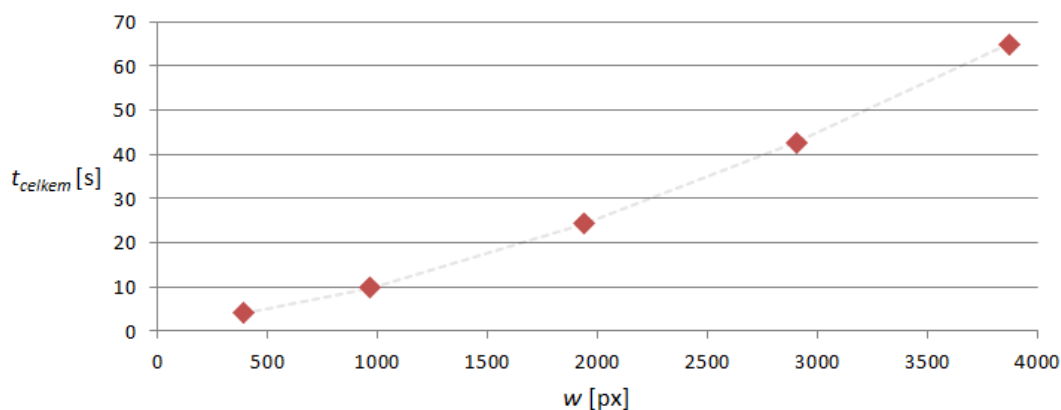
- I Odstranění distorze
- II Získání bodů
- III Rozpoznání inlierů
- IV Rekonstrukce scény
- V Rektifikace
- VI Výpočet disparity
- VII Podrobná rekonstrukce

Graf závislosti celkových časů rekonstrukce na šířce fotografie lze vidět na obr. 7.4. Je patrné, že zmenšením rozměrů fotografií dochází k výraznému zrychlení rekonstrukce. Dochází však také ke ztrátě informací uložených ve fotografii, což se projeví menším počtem detekovaných bodů, jak je vidět v Tabulce 7.2. Zajímavým úkazem je zvýšení celkového počtu bodů podrobné rekonstrukce pro některá zmenšení. Při zmenšení rozměrů fotografie totiž dochází k částečnému odstranění šumu a tento má vliv na přesnost výpočtu disparity. Při podrobné rekonstrukci se totiž odstraňují body s hodnotou disparity, která se výrazně liší od disparit bodů v okolí.

Při příliš velkém zmenšení (10%) však dochází k velkému poklesu počtu rekonstruovaných bodů. Proto dále budeme používat fotografie ze Sady 1 a Sady 2 v rozlišení 968x648 (25% původní velikosti). Tento rozměr pro dané sady umožňuje výrazně rychlejší výpočet se zachováním dostatečného množství informací pro správnou rekonstrukci.

Rozměry fotografií	časy výpočtů v jednotlivých úsecích výpočtu [s]							t_{celkem} [s]
	I	II	III	IV	V	VI	VII	
3872x2592 (100)%	2,643	23,965	15,544	0,747	1,758	19,338	0,880	64,876
2904x1944 (75)%	1,427	16,027	12,051	0,484	1,118	10,589	0,899	42,596
1936x1944 (50)%	0,734	10,148	6,908	0,309	0,585	4,785	0,855	24,324
968x648 (25)%	0,348	3,815	2,967	0,258	0,282	1,321	0,848	9,829
387x259 (10)%	0,218	0,744	1,002	0,239	0,247	0,718	0,980	4,140

Tabulka 7.1: Časy výpočtů dle rozměrů fotografií, Sada 1.



Obrázek 7.4: Graf závislosti času celého výpočtu t_{celkem} na šířce obrazu w , Sada 1.

Rozměry fotografií	nalezených dvojic	bodů rekonstrukce	bodů podrobné rek.
3872x2592 (100)%	2789	935	6059
2904x1944 (75)%	2219	859	6543
1936x1944 (50)%	1359	571	6891
968x648 (25)%	517	247	6131
387x259 (10)%	128	75	3494

Tabulka 7.2: Počty získaných dvojic a bodů během rekonstrukce dle rozměrů fotografií, Sada 1.

Tr	úsek výpočtu			
	I		II	
	počet dvojic	čas	počet inlierů	čas
100	559	4,103	287	12,636
200	420	3,050	230	9,728
300	321	2,457	194	7,505
400	284	2,078	165	6,655
500	245	1,811	148	5,797

Tabulka 7.3: Čas a počet detekovaných dvojic v závislosti na hodnotě prahu Tr , Sada 1.

Testování doby výpočtu

Označme hodnotu prahu zadávaného v úseku I jako Tr . Pro různé hodnoty tohoto prahu byly naměřeny časy výpočtů úseků I a II, počet nalezených dvojic a počet inlierů. V úseku II byla nastavena hodnota prahu na 1 a počet pokusů na 2000. Poměrně vysoký počet pokusů při nízkém počtu získaných dvojic napomáhá k opakovatelnosti měření. Naměřené hodnoty pro Sadu 1 jsou v Tabulce 7.2 a pro Sadu 2 v Tabulce 7.4. Z naměřených hodnot je patrné, že zvýšením prahu Tr dochází k nalezení menšího počtu dvojic a inlierů, ale urychlí se celý výpočet.

Dále bylo provedeno 5 měření doby výpočtu se stejným nastavením. Toto nastavení lze nalézt v Tabulce 7.5. Naměřené hodnoty jsou v Tabulce 7.6. Z vypočtených průměrných hodnot byl sestaven kruhový diagram na obr. 7.5. Tento graf udává podíl časů výpočtu jednotlivých úseků na celkovém čase výpočtu. Z grafu je patrné, že časově nejnáročnější pro toto nastavení jsou úseky II a III, tedy detekce a spárování bodů a následné hledání

Tr	úsek výpočtu			
	I		II	
	počet dvojic	čas	počet inlierů	čas
100	655	3,615	480	14,475
200	499	3,017	360	11,086
300	410	2,834	300	8,874
400	342	2,570	256	7,878
500	309	2,050	235	6,969

Tabulka 7.4: Čas a počet detekovaných dvojic v závislosti na hodnotě prahu Tr , Sada 2.

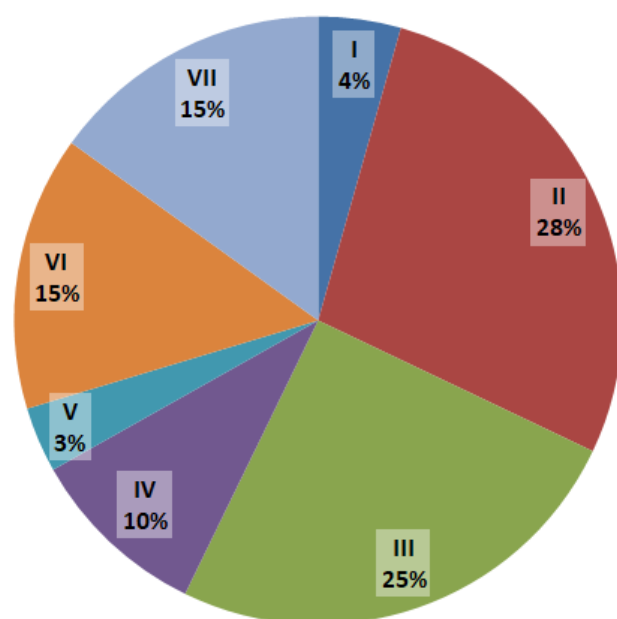
Úsek	Nastavení	Hodnota
II	Práh	300
	Maska	0
III	Pokusů	500
	Práh	1
VI	Velikost bloku	7
	Interval hodnot disparity	(-128;128)
VII	Rozestup bodů	8
	Poloměr	3
	Práh	10
	Filtrace pozadí	0

Tabulka 7.5: Nastavení programu pro měření v Tabulce 7.6.

inlierů. Jak ale bylo ukázáno, tyto úseky lze spočítat rychleji, pokud se zvýší hodnota prahu. Velice rychle jsou naopak spočítány úseky I (odstranění distorze) a V (rektifikace obrazů), které jsou pouze ovlivněny velikostí vstupních fotografií.

číslo spuštění	časy výpočtů v jednotlivých úsecích výpočtu [s]							$t_{celkem}[s]$
	I	II	III	IV	V	VI	VII	
1	0,373	2,403	2,078	0,919	0,337	1,305	1,351	8,766
2	0,398	2,513	2,244	0,887	0,290	1,250	1,352	8,934
3	0,413	2,430	2,230	0,904	0,295	1,314	1,369	8,955
4	0,376	2,520	2,159	0,781	0,310	1,360	1,300	8,806
5	0,384	2,418	2,440	0,778	0,312	1,258	1,311	8,901
průměr	0,388	2,456	2,230	0,853	0,308	1,297	1,336	8,872

Tabulka 7.6: Časy výpočtů jednotlivých úseků, Sada 1.



Obrázek 7.5: Podíl časů výpočtů jednotlivých úseků na celkovém čase výpočtu, Sada 1.

7.2.2. Rekonstrukce z více obrazů

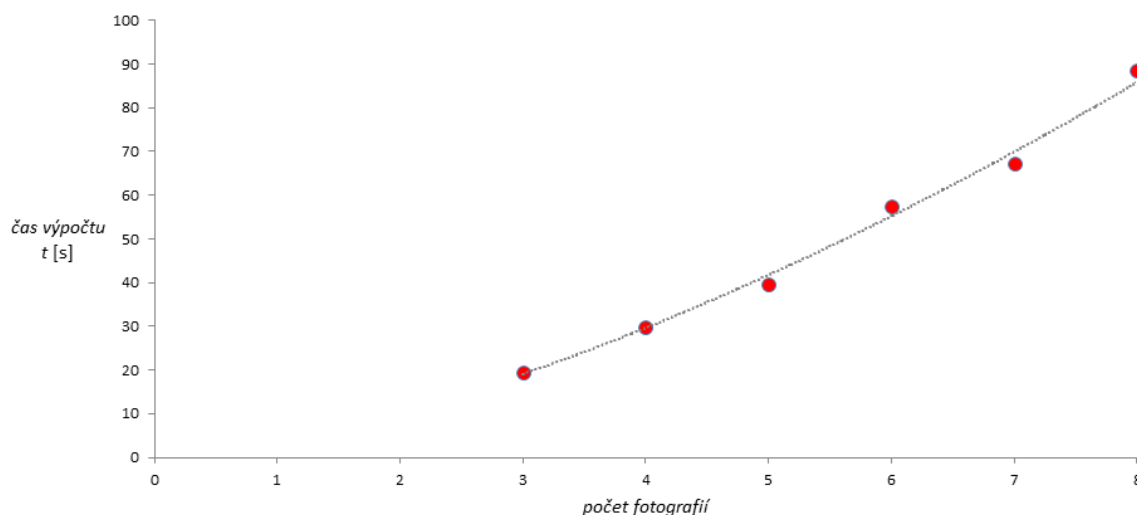
Rekonstrukce z více obrazů byla testována pouze z hlediska času potřebného na výpočet v závislosti na počtu vstupních fotografií. Závislost na jiných parametrech je podobná jako u rekonstrukce ze dvou obrazů. Oproti rekonstrukci ze dvou obrazů je lehce pozměněn algoritmus rekonstrukce, a proto i počet úseků. Úseky v rekonstrukci z více fotografií jsou

- I Detekce a spárování
- II Rozpoznání inlierů
- III Rekonstrukce kamer
- IV Podrobná rekonstrukce

Program postupně provedl rekonstrukci užitím tří až osmi fotografií ze Sady 3. Naměřené časy jsou zaznamenány v Tabulce 7.7 a použité hodnoty v Tabulce 7.5. Celková doba výpočtu je znázorněna na obr. 7.6. Body jsou v grafu proloženy křivkou $y = 3,5414x^{1,531}$, což naznačuje, že závislost není lineární. Z tabulky je patrné, že nejnáročnější úsek výpočtu je úsek IV.

počet fotografií	časy výpočtů v úsecích				t [s]
	I	II	III	IV	
3	0,554	3,242	0,842	14,683	19,320
4	3,242	4,651	0,914	23,689	29,889
5	0,842	6,101	0,920	31,881	39,657
6	14,683	7,513	0,969	48,262	57,606
7	19,320	9,033	1,000	56,123	67,164
8	0,635	10,519	1,063	75,721	88,431

Tabulka 7.7: Časy výpočtů jednotlivých úseků dle počtu vstupních obrazů, Sada 3.



Obrázek 7.6: Graf závislosti doby výpočtu t na počtu vstupních fotografií, Sada 3.

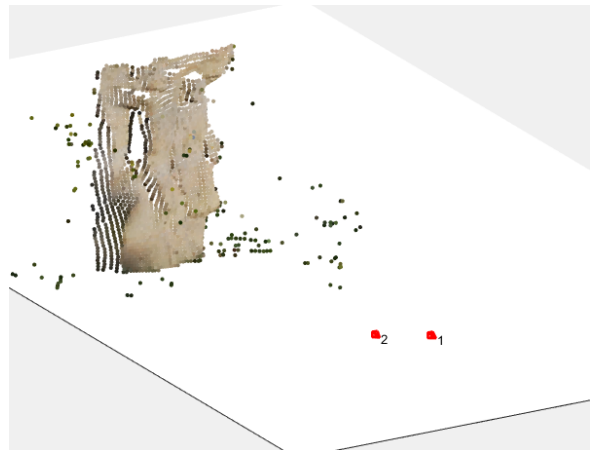
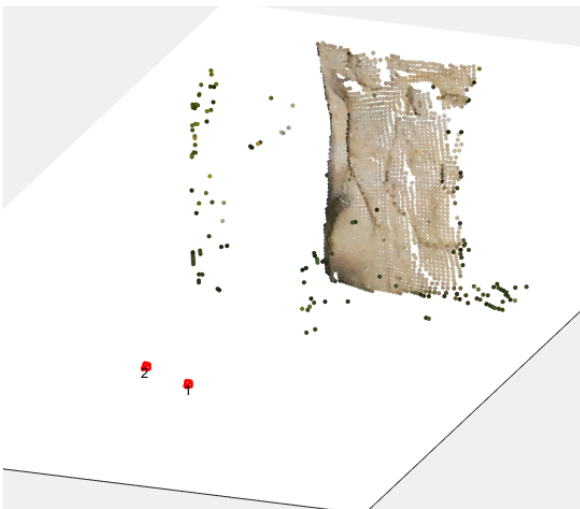
7.2.3. Výsledné rekonstrukce

Na následujících stranách se nachází vstupní fotografie a výstupní scény. Jak lze vidět, scény ze sady 1 a sady 2 byly zrekonstruovány poměrně věrně. Výsledné scény nabývají tvarů, které byly očekávány, přesto vzniklé rekonstrukce mají nedostatky jako například plochy bez zrekonstruovaných bodů (např. část vstupních dveří a jejich okolí v sadě 1).

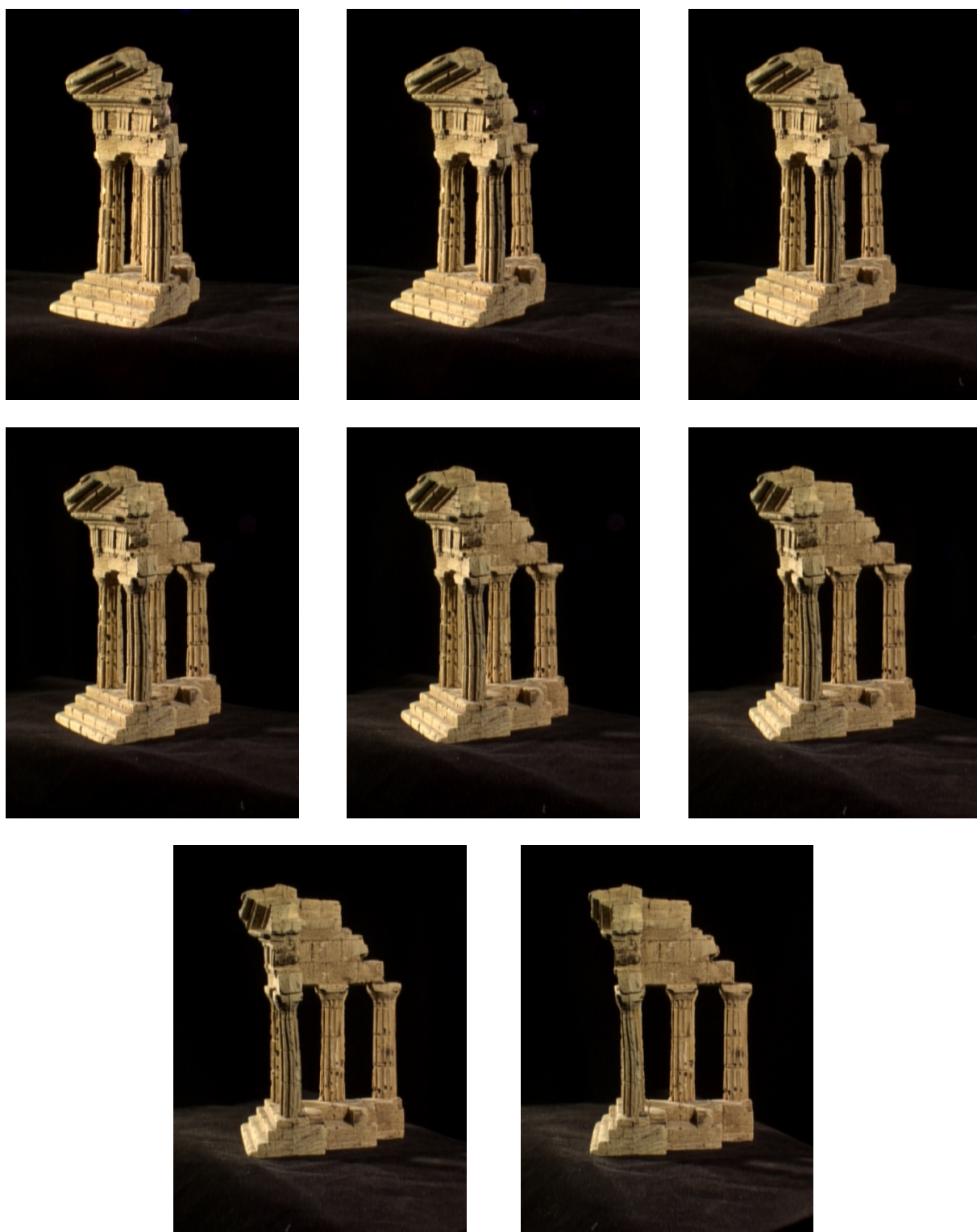
Ve scéně vzniklé rekonstrukcí ze sady 3 je možné pozorovat deformaci oproti reálnému objektu. Příkladem jsou sloupy, které jsou viditelně širší, než by měly být. Důvodem je zřejmě vyšší počet vstupních obrazů, neboť nepřesnosti ve výpočtu se s každým dalším obrazem zvyšují. Přesto je objekt ve scéně jasně rozeznatelný a jsou zachovány hlavní geometrické rysy.



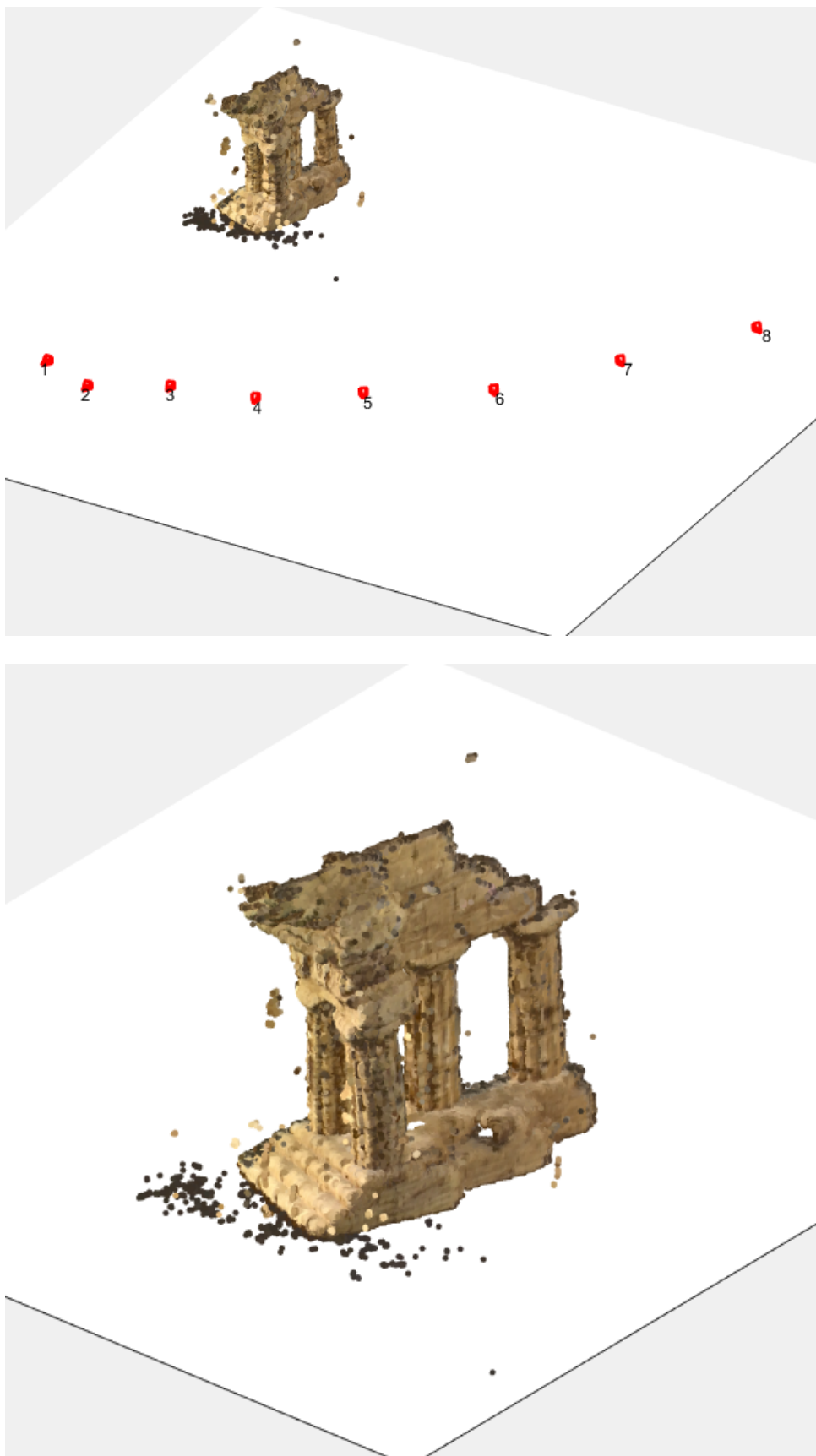
Obrázek 7.7: Rekonstrukce scény ze dvou obrazů, Sada 1.



Obrázek 7.8: Rekonstrukce scény ze dvou obrazů, Sada 2.



Obrázek 7.9: Fotografie pro rekonstrukci z více obrazů, Sada 3.



Obrázek 7.10: Rekonstrukce scény z více obrazů, Sada 3.

Závěr

Cílem této práce bylo popsat teorii a algoritmy vhodné pro rekonstrukci scény z více pohledů. Dále bylo za úkol vybrat vhodné metody na základě vstupů a naprogramovat vlastní aplikaci.

V této práci byly nejdříve zavedeny projektivní prostory, které umožňují zavedení homogenních souřadnic. Dále byly ukázány typy transformací v projektivních prostorech \mathbb{P}^2 a \mathbb{P}^3 . Byl tak položen teoretický základ pro další kapitoly. Poté byl popsán jednoduchý model kamery, který byl postupně rozšířen až na model, který odpovídá moderním digitálním fotoaparátům a kamerám.

Další část se týkala geometrie dvou pohledů, tzv. epipolární geometrie. Pro výpočet fundamentální matice byly uvedeny dvě metody lišící se v počtu použitých bodů. Poté byl ukázán způsob převodu fundamentální matice na matici esenciální a využití této matice k výpočtu matic kamer. Klíčovou roli pro rekonstrukci z více pohledů má algoritmus DLT, který umožňuje nalezení rotace a translace kamery pomocí nalezených bodů scény.

Na geometrii dvou pohledů navazuje geometrie pohledů tří. Pomocí trifokálního tenzoru byly zavedeny vztahy mezi body a přímkami ve třech pohledech. Na tuto část navazuje ukázka závislosti mezi epipolární geometrií a trifokálním tenzorem. K výpočtu trifokálního tenzoru poté slouží aparát založený na Carlsson-Weinshallové dualitě.

Nedílnou součástí rekonstrukce scény je i získání informací z fotografií. Metoda SURF umožňuje detekci a párování klíčových bodů mezi obrazy. Druhou možností je pak použití rektifikace a mapy disparity.

Vlastní přínos práce je v podrobném teoretickém popisu problému rekonstrukce 3D objektů z fotografií. Tyto obecné algoritmy byly zpracovány ve funkční aplikaci, která je schopná rekonstrukce ze dvou i více obrazů. Bylo popsáno jakým způsobem aplikace používá dané metody z teoretické části. Aplikace navíc kromě rekonstrukce umožňuje i ukládání zrekonstruovaných scén, takže je možno se získanými daty dále pracovat. Pro všechny části výpočtu byly provedeny měření časové závislosti celého procesu i dílčích kroků. Pro různé velikosti vstupních fotografií bylo vyhodnoceno optimální nastavení v závislosti na rychlosti a přesnosti rekonstrukce.

Na práci je možné navázat rozšířením o autokalibraci, která umožňuje provádět rekonstrukci i bez znalosti kalibrační matice. Dalším možným směrováním může také být například vytvoření sítě modelu ze získaných zrekonstruovaných bodů scény či doplnění nezrekonstruovaných oblastí.

Literatura

- [1] BARTOŠOVÁ, Barbora. *Projektivní geometrie v úlohách*. Olomouc, 2012. Diplomová práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, Katedra algebry a geometrie. Vedoucí práce Marek Jukl.
- [2] BAY, Herbert, Andreas ESS, Tinne TUYTELAARS a Luc VAN GOOL. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*. 2008, **110**(3), 346-359. DOI: 10.1016/j.cviu.2007.09.014. ISSN 10773142. Dostupné také z: <https://linkinghub.elsevier.com/retrieve/pii/S1077314207001555>
- [3] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, c2008. ISBN 978-0-596-51613-0.
- [4] CARLSSON, Stefan. Duality of reconstruction and positioning from projective views. *Proceeding VSR '95 Proceedings of the IEEE Workshop on Representation of Visual Scenes*. 1995.
- [5] CHIEN, Hsiang-Jen (Johnny). Direct Linear Transformation (DLT) Solver. *Mathworks* [online]. [cit. 2019-05-21]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/65030-direct-linear-transformation-dlt-solver>
- [6] DERPANIS, Konstatinos G. *Integral Image-base representations*. 2007. Dostupné také z: http://www.cs.yorku.ca/kosta/CompVis_Notes/integral_representations.pdf
- [7] DRUCKMÜLLER, Miloslav. *Numerické metody analýzy obrazů. (přednášky)*. Brno: VUT v Brně, Fakulta strojího inženýrství, 2018.
- [8] DRUCKMÜLLEROVÁ, Hana. *Phase-correlation based image registration*. Brno, 2010. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojího inženýrství, Ústav matematiky. Vedoucí práce Jana Procházková.
- [9] EUCLID, FITZPATRICK, Richard, ed. *Euclid's Elements of geometry*. 2008. ISBN 978-0-6151-7984-1.
- [10] FISCHLER, Martin A. a Robert C. BOLLES. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*. **24**(6), 381-395. DOI: 10.1145/358669.358692. ISSN 00010782. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=358669.358692>
- [11] HARTLEY, Richard a Andrew ZISSERMAN. *Multiple view geometry in computer vision*. 2nd ed. New York: Cambridge University Press, 2003. ISBN 978-0521540513.
- [12] HEIKKILA, J. a O. SILVEN. A four-step camera calibration procedure with implicit image correction. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 1997, , 1106-1112. DOI: 10.1109/CVPR.1997.609468. ISBN 0-8186-7822-4. Dostupné také z: <http://ieeexplore.ieee.org/document/609468/>

- [13] KALMAN, Dan. A Singularly Valuable Decomposition: The SVD of a Matrix. *The College Mathematics Journal*. 2018, **27**(1), 2-23. DOI: 10.1080/07468342.1996.11973744. ISSN 0746-8342. Dostupné také z: <https://www.tandfonline.com/doi/full/10.1080/07468342.1996.11973744>
- [14] KARÁSEK, Jiří a Ladislav SKULA. *Lineární algebra: teoretická část*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3100-8.
- [15] KRESSNER, Daniel. *Numerical methods for general and structured eigenvalue problems*. New York: Springer, 2005. ISBN 35-402-4546-4.
- [16] LANG, Ladislav. *Rekonstrukce objektu a pozice pozorovatele z 2D pohledů*. Plzeň, 2004. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Václav Skala.
- [17] LITWILLER, Dave. CCD vs. CMOS: facts and fiction. *Photonics Spectra*. 2001, **35**, 154-158.
- [18] MARTIŠEK, Dalibor. 02 - Euklidovská Geometrie. *Počítačová geometrie a grafika: Přednáška* [online]. 2012-11-20 [cit. 2018-10-10]. Dostupné z: <http://mathonline.fme.vutbr.cz/Prednaska/sc-1245-sr-1-a-261/default.aspx>
- [19] MARTIŠEK, Dalibor. 03 - Projektivní Geometrie. *Počítačová geometrie a grafika: Přednáška* [online]. 2012-11-20 [cit. 2018-10-10]. Dostupné z: <http://mathonline.fme.vutbr.cz/Prednaska/sc-1245-sr-1-a-261/default.aspx>
- [20] MRKVIČKA, Daniel. *Clustering mraku bodů*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Jana Procházková.
- [21] OYALLON, Edouard a Julien RABIN. An Analysis of the SURF Method. *Image Processing On Line*. 2015, **5**, 176-218. DOI: 10.5201/ipol.2015.69. ISSN 2105-1232. Dostupné také z: http://www.ipol.im/pub/art/2015/69/?utm_source=doi
- [22] SCHARSTEIN, D., R. SZELISKI a R. ZABIH. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*. IEEE Comput. Soc, 2001, , 131-140. DOI: 10.1109/SMBV.2001.988771. ISBN 0-7695-1327-1. Dostupné také z: <http://ieeexplore.ieee.org/document/988771/>
- [23] SEITZ, S.M., B. CURLESS, J. DIEBEL, D. SCHARSTEIN a R. SZELISKI. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. IEEE, 2006. DOI: 10.1109/CVPR.2006.19. ISBN 0-7695-2597-0. Dostupné také z: <http://ieeexplore.ieee.org/document/1640800/>
- [24] WEINSHALL, Daphna, Michael WERMAN a Amnon SHASHUA. Duality of multi-point and multi-frame geometry: Fundamental shape matrices and tensors. *Computer Vision - ECCV '96*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, 1996-6-2, , 217-227. Lecture Notes in Computer Science.

DOI: 10.1007/3-540-61123-1_141. ISBN 978-3-540-61123-3. Dostupné také z: http://link.springer.com/10.1007/3-540-61123-1_141

- [25] YANG, Guangyi, Deshi LI, Guobao RU, Jiahua CAO a Weizheng JIN. Body Height Estimation System Based on Binocular Vision. *International Journal of Online Engineering (iJOE)*. 2018, **14**(04), 177-192. DOI: 10.3991/ijoe.v14i04.8400. ISSN 1861-2121. Dostupné také z: <http://online-journals.org/index.php/i-joe/article/view/8400>
- [26] ZHANG, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, **22**(11), 1330-1334. DOI: 10.1109/34.888718. ISSN 01628828. Dostupné také z: <http://ieeexplore.ieee.org/document/888718/>
- [27] History of Camera Obscura - Who Invented Camera Obscura?. *Photography History Facts* [online]. c2018 [cit. 2018-12-22]. Dostupné z: <http://www.photographyhistoryfacts.com/photography-development-history/camera-obscura-history/>
- [28] Tsai grid to be used for camera calibration. In: <https://eyebug.wordpress.com> [online]. [cit. 2018-11-18]. Dostupné z: <https://eyebug.files.wordpress.com/2010/09/test2.png>

Seznam použitých symbolů a veličin

Označení	Popis
DLT	Direct Linear Transformation, viz kapitola 3.5
SSD	Sum of Squared Differences, viz kapitola 6.2.1
SURF	Speeded-Up Robust Features, viz kapitola 5
\det	determinant
\cdot	skalární součin
\times	vektorový součin
\mathbb{R}	množina reálných čísel
\mathbb{N}	množina přirozených čísel
\mathbb{P}^2	projektivní rovina
\mathbb{P}^3	projektivní prostor
ε	Levi-Civitův tenzor
\mathbf{e}	epipól
F	fundamentální matice
\mathcal{T}	trifokální tenzor
I	obraz
K	kalibrační matice
P	matice kamery
C	optický střed kamery
R	matice rotace
\mathbf{t}	vektor translace
$\mathbf{x} = (x, y, 1)^\top, \mathbf{y}$	bod obrazu, bod v \mathbb{P}^2
$\mathbf{X} = (X, Y, Z, 1)^\top$	bod scény, bod v \mathbb{P}^3
\mathbf{l}	přímka v obraze, přímka v \mathbb{P}^2
\mathbf{L}	přímka scény, přímka v \mathbb{P}^3
π	rovina v \mathbb{P}^3

A. Postup instalace knihovny mexopencv

Vytvořená aplikace potřebuje ke svému chodu nainstalovaný *Computer Vision Toolbox*. Pokud jej uživatel nemá k dispozici, je možné použít knihovnu *mexopencv* společně s knihovnou *openCV*. Instalace těchto knihoven není jednoduchá, a proto zde bude vysvětlen postup instalace *mexopencv* 3.0.0¹. K instalaci knihovny je nutné mít nainstalován program *Visual Studio 2015 Community*. Je možné použít i jiné verze, ale zde uvedený postup byl testován a upraven pro zmíněnou verzi.

Nejprve je nutné stáhnout knihovnu *openCV* verze 3.0.0 a doplňkovou knihovnu *openCV_contrib*, která obsahuje metodu SURF.

<https://github.com/opencv/opencv/tree/3.0.0>

https://github.com/opencv/opencv_contrib/tree/3.0.0.

Stažení se provede tlačítkem *Clone or download* a volbou *Download ZIP*. Oba stažené soubory rozbalíme například do složky `C:\dev`. Při správném postupu bychom pak měli nalézt tyto dva soubory:

- `C:\dev\opencv-3.0.0\README.md`
- `C:\dev\opencv_contrib-3.0.0\README.md`

V dalším kroku je nutné stažení programu CMake(<https://cmake.org/>). Tento program předpřipraví staženou knihovnu pro instalaci. Program spustíme pomocí souboru *cmake-gui.exe* nacházející se v adresáři *bin* složky programu.

Po spuštění CMake je potřeba v programu provést řadu úkonů:

- nastavit *source folder* na `C:/dev/opencv-3.0.0`
- nastavit *destination folder* na `C:/dev/build`
- stisknout *Configure* a v kolonkách nastavit
 - generator: Visual Studio 14 2015
 - optional platform: Win64
- potvrdit tlačítkem *Finish*, poté se spustí proces konfigurace
- po dokončení procesu je potřeba ve vzniklém seznamu odškrtnout tyto položky (pokud existují):
 - BUILD_DOCS
 - BUILD_EXAMPLES
 - BUILD_PACKAGE
 - BUILD_PERF_TESTS
 - BUILD_TESTS,
 - BUILD_JAVA,
 - BUILD_opencv_apps
 - BUILD_opencv_cuda*
 - BUILD_opencv_cudev

¹postup se často mění s verzí *mexopencv*, návody v angličtině pro jiné verze je možné najít na <https://github.com/kyamagu/mexopencv/wiki>, tyto návody obsahují navíc řadu náhledů obrazovky

- BUILD_opencv_js
 - BUILD_opencv_java*
 - BUILD_opencv_python*
 - BUILD_opencv_ts
 - BUILD_opencv_viz
 - BUILD_opencv_world
 - WITH_CUDA
 - WITH_CUFFT
 - WITH_CUBLAS
 - WITH_NVCUVID
 - WITH_MATLAB
 - WITH_VTK
- dále nastavit `OPENCV_EXTRA_MODULES_PATH` na `C:/dev/opencv_contrib-3.0.0/modules`
 - zaškrtnout `OPENCV_ENABLE_NONFREE`
 - znovu stisknout *Configure* a poté odškrtnout tyto položky (pokud existují):
 - BUILD_opencv_cvv
 - BUILD_opencv_freetype
 - BUILD_opencv_hdf
 - BUILD_opencv_matlab
 - BUILD_opencv_ovis
 - BUILD_opencv_sfm
 - stisknout *Configure*
 - stisknout *Generate*
 - zavřít program

Nyní je nutné spustit program *Visual Studio* a dále v něm otevřít soubor `C:/dev/build/OpenCV.sln` a provést následující:

- nastavit debugger do módu *Release* (z *Debug*)
- v podokně *Solution explorer* nalézt soubor `ALL_BUILD` (nejspíše ve složce `CMakeTargets`)
- na tento soubor kliknout pravým tlačítkem a zvolit *Build*, spustí se dlouhý proces
- v podokně *Solution explorer* nalézt soubor `INSTALL` (nejspíše ve složce `CMakeTargets`)
- na tento soubor kliknout pravým tlačítkem a zvolit *Build*
- zavřít program

Dalším krokem je přidání cesty `C:\dev\build\install\x64\vc14\bin` do proměnné prostředí `PATH`.

- v kontextovém menu *Můj počítač* kliknout na položku *Vlastnosti*
- v okně *Vlastnosti systému* zvolit záložku *Upřesnit*
- kliknout na *Proměnné prostředí*
- v okně *Systémové proměnné* najít proměnnou `PATH`, případně ji přidat.
- vložit do proměnné cestu k výše zmíněné složce, proměnná obsahuje seznam adresářů oddělených středníkem
- zavřít vlastnosti systému

Nyní je již možné stáhnout knihovnu *mexopencv* ze stránky

<https://github.com/kyamagu/mexopencv/tree/v3.0.0>

Postup stažení je stejný jako u knihovny *openCV*. Stažený soubor je nutné rozbalit, opět například do složky *C:\dev*. Po rozbalení by pak měl existovat soubor

- *C:\dev\mexopencv-3.0.0\README.markdown*

Posledním krokem je otevření programu *MATLAB* a spuštění tohoto kódu:

```
>> cd('C:\dev\mexopencv-3.0.0')
>> addpath('C:\dev\mexopencv-3.0.0')
>> addpath('C:\dev\mexopencv-3.0.0\opencv_contrib')
>> mexopencv.make('opencv_path','C:\dev\build\install','opencv_contrib',true)
```

Tím je knihovna nainstalována a připravena k použití. Při každém spuštění programu *MATLAB* pak pro využití knihovny stačí zadat

```
>> addpath('C:\dev\mexopencv-3.0.0')
>> addpath('C:\dev\mexopencv-3.0.0\opencv_contrib')
```

Alternativou je spuštění pomocí aplikace vytvořené v rámci této diplomové práce. V okně *main.fig* zvolíme *Nastavení/Cesta k MexOpenCV/Nastavit...* a v otevřeném dialogovém okně zadáme cestu ke složce *C:\dev\mexopencv-3.0.0*. Aplikace pak sama provede dva předchozí příkazy.

B. Obsah přiloženého CD

/	kořenový adresář přiloženého CD
program	kódy aplikace
funcFolder	pomocné funkce
dlt.m	
F_from_xx.m	
P_from_FK.m	
plotCam.m	
RANSAC_2views.m	
RecTrans_find.m	
RecTrans_run.m	
RQ.m	
triang.m	
x_from_PX.m	
guiFolder	vedlejší okna aplikace
multi.fig	
multi.m	
plot3d.fig	
plot3d.m	
main.fig	hlavní okno aplikace, umožňuje spuštění
main.m	
sets	testovací sady fotografií
sada 1	
set1_01.JPG	
set1_02.JPG	
set1_cp.mat	
sada 2	
set2_01.JPG	
set2_02.JPG	
set2_cp.mat	
sada 3	
set3_01.png	
set3_02.png	
set3_03.png	
set3_04.png	
set3_05.png	
set3_06.png	
set3_07.png	
set3_08.png	
set3_cp.mat	
readme.txt	soubor s popisem obsahu CD