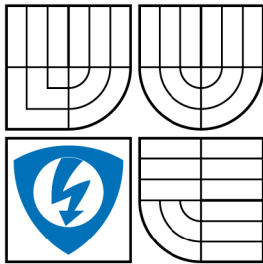


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

OPTOELEKTRONICKÉ A FOTOGRAMMETRICKÉ MĚŘICÍ SYSTÉMY

OPTOELECTRONIC AND PHOTOGRAMMETRIC MEASURING SYSTEMS

DOKTORSKÁ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE
AUTHOR

Ing. Petr Stančík

VEDOUCÍ PRÁCE
SUPERVISOR

Prof. Ing. Václav Říčný, CSc.

BRNO 2008

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Ing. Petr Stančík

Bytem:

Narozen/a (datum a místo): 29.4.1978, Vsetín

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 60200 Brno 2

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. RNDr. Vladimír Aubrecht, CSc.

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce

jiná práce, jejíž druh je specifikován jako

(dále jen VŠKP nebo dílo)

Název VŠKP: Optoelektronické a fotogrammetrické měřicí systémy

Vedoucí/školicel VŠKP: prof. Ing. Václav Řičný, CSc.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- tištěné formě - počet exemplářů 1
- elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Disertace se zabývá analýzou a návrhem optoelektronických a fotogrammetrických měřících systémů. Obsahuje konkrétní návrhy optoelektronických bezdotykových měřičů ploch rovinných objektů, případně plošné projekce 3D objektů včetně analýzy dosažitelné přesnosti měření. V další části věnované stereofotogrammetrii se zabývá principy rekonstrukce prostorových souřadnic snímaných objektů, metodami automatické kalibrace kamer, postupy vztotožňování bodů na snímcích spolu s analýzou dosažitelné přesnosti v určování sledovaných parametrů. Součástí práce je vyvinutý testovací program v němž jsou uvedené postupy implementovány a který umožňuje praktickou aplikaci stereofotogrammetrického systému pro pořizování prostorových souřadnic trojrozměrných objektů.

Abstract

Dissertation deals with analysis and design of optoelectronic and photogrammetric measuring systems. Specific design of optoelectronic contactless flat object area meters with analysis of attainable measurement accuracy is described. Next part is dedicated to stereophotogrammetry - principles of 3D reconstructions, methods of camera self-calibration and matching points in images are described. The analysis of attainable accuracy of monitored parameters is discussed too. Finally, the test program with implemented described routines is introduced. This test program enables practical application of stereophotogrammetric system for taking spatial coordinates of 3D objects.

Klíčová slova

Optoelektronické bezdotykové měřiče ploch, stereofotogrammetrie, kalibrace kamer, sesouhlasení bodů na snímcích, rekonstrukce prostorových souřadnic snímaných objektů

Keywords

Optoelectronic contactless area meters, stereophotogrammetry, camera self-calibration, matching points, 3D reconstruction

Bibliografická citace

STANČÍK, P. *Optoelektronické a fotogrammetrické měřicí systémy*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 89 s. Vedoucí dizertační práce prof. Ing. Václav Říčný, CSc.

Prohlášení

Prohlašuji, že jsem tuto disertační práci na téma „Optoelektronické a fotogrammetrické měřicí systémy“ vypracoval samostatně pod vedením svého vedoucího práce s použitím odborné literatury, kterou jsem citoval v seznamu literatury.

Poděkování

V úvodu této práce bych rád poděkoval Prof. Ing. Václavu Říčnému, CSc. nejen za odborné vedení a rady, ale i za prospěšné kritické připomínky v průběhu její realizace. Dále děkuji své ženě, rodičům a blízkým, za podporu v nelehkých chvílích a malému Míšovi za stálý úsměv a pomoc při mazání textu...

Obsah

1	ÚVOD	10
1.1	FORMULACE PROBLÉMU	10
1.2	SOUČASNÝ STAV.....	11
1.3	CÍLE DISERTACE	13
2	ANALÝZA A SYSTÉMOVÝ NÁVRH OPTOELEKTRONICKÝCH BEZDOTYKOVÝCH MĚŘIČŮ PLOCH	14
2.1	MĚŘIČ S ANALOGOVÝM ZPRACOVÁNÍM OBRAZOVÉHO SIGNÁLU	14
2.1.1	<i>Blokové schéma.....</i>	<i>14</i>
2.1.2	<i>Systémová relativní chyba měření</i>	<i>14</i>
2.1.3	<i>Vliv dílčích funkčních částí na výslednou chybu měření.....</i>	<i>15</i>
2.2	MĚŘIČ S ČÍSLICOVÝM ZPRACOVÁNÍM OBRAZOVÉHO SIGNÁLU	18
2.2.1	<i>Blokové schéma.....</i>	<i>18</i>
2.2.2	<i>Systémová relativní chyba měření a rozbor vlivu dílčích funkčních bloků.....</i>	<i>18</i>
2.2.3	<i>Odhad celkové chyby měření a porovnání obou metod.....</i>	<i>20</i>
2.3	PROGRAM AD-EVAL PRO MĚŘENÍ PLOCH A ROZMĚRŮ	21
2.3.1	<i>Vývojový diagram programu</i>	<i>21</i>
2.3.2	<i>Komunikační okna ovládacího programu.....</i>	<i>22</i>
2.3.3	<i>Základní ovládání programu AD-EVAL.....</i>	<i>24</i>
3	REKONSTRUKCE PROSTOROVÝCH SOUŘADNIC NA ZÁKLADĚ STEREOFOTOGAMMETRIE	25
3.1	MATEMATICKÉ TRANSFORMACE V PROSTORU	25
3.2	GEOMETRICKÝ MODEL KAMERY	27
3.3	REKONSTRUKCE ZE DVOU KALIBROVANÝCH SNÍMKŮ	30
3.3.1	<i>Epipolární geometrie, esenciální matice</i>	<i>30</i>
3.3.2	<i>Osmi-bodový lineární algoritmus</i>	<i>32</i>
3.4	REKONSTRUKCE ZE DVOU NEKALIBROVANÝCH SNÍMKŮ	33
3.4.1	<i>Epipolární geometrie pro nekalibrované kamery, fundamentální matice</i>	<i>33</i>
3.4.2	<i>Rozvrstvený přístup při rekonstrukci</i>	<i>34</i>
3.4.3	<i>Kalibrace založená na znalosti scény.....</i>	<i>37</i>
3.5	ZPŮSOBY ZTOTOŽŇOVÁNÍ BODŮ NA SNÍMCÍCH	40
3.5.1	<i>Globální metody.....</i>	<i>42</i>
3.5.2	<i>Lokální metody.....</i>	<i>43</i>
3.6	ANALÝZA PŘESNOSTI V URČENÍ PROSTOROVÉ SOUŘADNICE BODU.....	45
3.6.1	<i>Vliv kalibrace kamer.....</i>	<i>45</i>
3.6.2	<i>Vliv diskretizace obrazu.....</i>	<i>54</i>
3.7	TESTOVACÍ PROGRAM PRO REKONSTRUKCI PROSTOROVÝCH SOUŘADNIC SNÍMANÝCH OBJEKTŮ	54
3.7.1	<i>Vývojový diagram programu Recon3D</i>	<i>55</i>

3.7.2	<i>Komunikační okna programu Recon3D</i>	62
3.7.3	<i>Základní ovládání programu Recon3D</i>	64
4	ZÁVĚR	67
5	LITERATURA	69
6	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	74
7	SEZNAM PŘÍLOH	77
	CURRICULUM VITAE	78

1 Úvod

1.1 Formulace problému

V průmyslové praxi i dalších oborech často vyvstává potřeba bezdotykového měření rozměrů a ploch, přesněji vyjádřeno - plošných projekcí trojrozměrných objektů různých tvarů. Typickým příkladem mohou být např. úlohy v mikroelektronice, kde je měření ploch nezbytné pro technologické propočty. Řešení takového případu si lze jen stěží představit bez využití bezkontaktního měření. A právě zde se jako výhodné jeví využití optoelektronických metod měření ploch. V takovém případě postačí CCD (CMOS) kamera a zařízení, které dokáže analyzovat snímáný obraz a získat z něj hledané údaje.

S rostoucím výkonem výpočetní techniky a klesající cenou video karet pro připojení kamer, tzv. frame-grabberů, se naskytá možnost přejít k softwarovému vyhodnocení rozměrů snímáných objektů. Cílem práce tak bude vývoj algoritmů pro měření 2D rozměrů a ploch. Hlavním tématem však bude popis možných metod rekonstrukce 3D scény na základě stereoskopických snímků, optimalizace velmi náročných výpočtů a návrh programu, který by rekonstrukci prostorových souřadnic ze snímáné 3D scény prováděl automatizovaně.

Fotogrammetrie je vědní disciplína zabývající se rekonstrukcí trojrozměrných scén z dvou-rozměrných snímků. Její počátky se datují přibližně od poloviny minulého století. Dle českého Výkladového slovníku geodetických a kartografických názvosloví je fotogrammetrie obor, zabývající se rekonstrukcí tvaru, rozměru a polohy předmětů, zobrazených na fotografických snímcích. Základem fotogrammetrie jako měřické a mapovací metody je fotografický snímek. Od vytvoření definice tohoto oboru však uplynulo několik let a ve fotogrammetrii se stále více využívají digitální záznamy.

Poznámka: Speciální součástí fotogrammetrie je **stereofotogrammetrie**, ve které se vyšetřují geometrické rozměry 3D scény ze dvou (příp. i více) stereoskopických snímků. Právě této problematice bude věnována druhá část této práce.

Přechod od analogové k analytické fotogrammetrii nevedl k novým principům nebo výsledkům, ale pouze k určitým vylepšením. Přechod od analytické k digitální pak znamenal úplnou změnu hardwaru. V současné době se stereofotogrammetrické analýzy a výpočty uskutečňují převážně pomocí výpočetní techniky a vhodných speciálních programů a technické prostředky (hardware) slouží pro sejmutí a generování potřebných digitálních stereoskopických snímků. Kromě toho, že je možno implementovat již známé algoritmy k řešení úloh klasické fotogrammetrie, jako je triangulace, snímková orientace, ortoprojekce a stereoskopické měření, otevřela se cesta pro další metody a úlohy. Digitální fotogrammetrie tak v sobě obsahuje i některé metody zpracování obrazu (*image processig*) a počítačového vidění (*computer vision*). Jedná se například

o filtrování, ostření a změnu kontrastu snímku. Jsou to metody, které jsou při digitálním zpracování velmi jednoduše implementovatelné, ale podobných výsledků lze klasickými postupy dosáhnout pouze s obtížemi. Z oboru počítačového vidění se jedná například o automatické rozpoznávání objektu, což klasickými metodami není umožněno. Algoritmy pro porovnávání obrazu se dají využít při automatické orientaci snímku, triangulaci nebo při automatickém generování digitálního modelu terénu (DMT).

Mnohé současné vědní disciplíny (např. počítačové vidění, robotika a dálková měření atd.) užívají digitálních kamer k získávání informací o okolním světě. Jako rychlé, levné a přenosné zařízení je často užívána právě CCD, případně CMOS, kamera.

1.2 Současný stav

V oblasti 2D měření rozměrů objektů je současným trendem využívat poměrně nákladné komplexní měřicí systémy sestávající z průmyslových CCD (CMOS) kamer, digitalizačních karet a příslušného softwaru. Ten umožňuje řešení velkého počtu úloh ve vyhodnocování 2D rozměrů objektů. Jako příklad lze uvést systém EPIX[®] společnosti Neovision.

Základní principy stereofotogrammetrie jsou známy již dlouho a současný výzkum na nich staví. Problém rekonstrukce lze vlastně rozdělit do dvou rovin, zjištění prvků vnitřní a vnější orientace a nalezení odpovídajících si bodů na snímcích. Snahou je tyto operace co nejvíce zautomatizovat.

Kalibrace kamery spočívá v nalezení vnějších a vnitřních prvků orientace kamery, zahrnující ohniskovou vzdálenost a polohu středu promítání snímku. Dříve byla kalibrace prováděna ze znalosti prostorových bodů a jejich korespondence na snímku. Příkladem může být tenký plát s pravidelným vzorem, jehož 3D souřadnice jsou známy, umístěný do zorného pole kamery. Pak mohou být snímkové souřadnice z množství dobře rozeznatelných a měřitelných vlastností vzoru použity k určení parametrů kamery [80]. Tato metoda určuje parametry kamery velmi přesně, ale má několik nevýhod. První je, že kalibrační mřížka nemusí být vždy dostupná a druhá, že nelze kameru zkalibrovat pokud se její parametry mění, např. ostřením a přibližováním.

Kalibraci kamery je ovšem možné provádět i bez použití kalibrační mřížky. Faugeras, Luong a Maybank [12] uvedli ideu, že kamera může být kalibrována pomocí korespondence bodů mezi snímky a pojmenovali ji jako automatickou kalibraci. Fauregas [16] a Hartley [25] ukázali, že, i bez znalosti kalibrace kamer, můžeme získat rekonstrukci až ke projektivní transformaci ze dvou nekalibrovaných snímků. Od té doby bylo navrženo několik algoritmů, které se liší dovoleným posunem kamery a efektivním nalezením parametrů kamery. Na základě práce Kruppy bylo vytvořeno několik algoritmů pro kalibraci kamer [12],[39],[81],[55] stejně jako algoritmů pro robustní a přesný odhad fundamentální matice [5],[29],[77],[83],[85]. Některé metody kalibrují kameru přímo v jednom kroku, zatímco jiné využívají rozvrstvený přístup. Projektivní, afinní a euklidovské rozvrstvení při rekonstrukci bylo poprvé předloženo v [56] a později formulováno jako obecný matematický systém pro 3D rekonstrukci [14]. Nicméně euklidovská rekonstrukce není vždy realizovatelná pomocí těchto dílčích rekonstrukcí. Vnitřní nejednoznačnosti v rekonstrukci spojené se všemi typy kritických pohybů byly studovány v [68],[69],[87],[78],[43].

Částečná znalost snímané scény umožní snadnější kalibrační proces pro kamery, jedná se o kalibrační objekt [80], již zmiňovaný rovinný kalibrační obrazec [82], rovinné scény [79], orthogonalitu a paralelismus, pro něž jsou definovány tak zvané úběžné body (*vanishing points*). Postupy jak automaticky detekovat a vypočítat tyto body jsou uvedeny v [57], [8], [40].

Při rekonstrukci prostorových souřadnic snímaných objektů je stěžejní operací nalezení korespondence mezi dvojicí (trojicí, atd.) snímků. Korespondenční problém (*stereo matching*) byl zkoumán již před několika desetiletími. První důsledná teorie stereo vidění byla předložena v práci od Marr a Poggio [45]. Marr určil obecnou nejednoznačnost korespondenčního problému a definoval dvě základní omezení umožňující jasnou formulaci procesu sesouhlasení:

- jedinečnost: bod na jednom snímku může odpovídat nanejvýš jednomu bodu na druhém snímku (vycházející z předpokladu, že prostorový bod může být reprezentován v jednom snímku nejvýše jedním bodem)
- spjitost: disparita se všude mění téměř spojitě na povrchu objektů

Marr a Poggio navrhli přístup založený na ztotožnění základních rysů, odpovídajícím pozicím změn jasu – hrany, rohy, atd. (*feature matching*). Marrův princip, založený na předpokladu spjitosti povrchu byl zdokonalen pomocí PMF algoritmu navrženým Pollardem, Mayhewem a Frisbyem [54].

Na počátku devadesátých let minulého století se výzkumníci zaměřovali na modifikaci Marrovy teorie, kdy by se disparita určovala pro všechny pixely přímo. Takto byl vývoj algoritmů sesouhlasení rozdělen do dvou směrů. První směr, *globální metody*, reformuloval úlohu takovým způsobem, že byly disparita a všechny předpoklady o struktuře prostoru modelovány pomocí Bayesian proměnných a disparita pak určena pomocí Bayesian soustavy. Více o počátcích globálních metod a dynamického programování lze nalézt v [1], **Chyba! Nenalezen zdroj odkazů.**, [2], [17], [49]. Další směr představovaly *lokální metody*. Tyto metody byly na počátku 90let nahrazeny globálními metodami, ale na konci devadesátých let se přešlo zpět k metodám lokálním. Vycházejí z předpokladu, že není potřeba žádné globální optimalizace, ale naopak, lokální informace snímku je dostatečná pro správnou formulaci úlohy sesouhlasení. Při lokálních metodách je sesouhlasení prováděno na základě výběru nejvíce podobných elementů. Podstata lokálních metod spočívá v definici a výběru vhodného okna (na něž se metoda aplikuje) a statistice určené přes tuto lokální část snímku. Typické statistiky mohou být SSD (Sum of Squared Differences), SAD (Sum of Absolute Differences), NCC (normalized cross-correlation) atd. Hojně využívaná studie, zabývající se korespondenčním problémem, je dána v [61]. Autoři implementovali řadu algoritmů na testovacích stereo snímcích a uvedli porovnání těchto algoritmů.

1.3 Cíle disertace

Hlavním tématem disertační práce je analýza a vývoj optoelektronických a stereofotogrammetrických systémů pro měření 2D a 3D rozměrů objektů snímaných kamerou s CCD nebo CMOS snímačem.

V části zabývající se měřením velikosti ploch a rozměrů rovinných objektů představují cíle disertace:

- analytický rozbor a systémový návrh autonomních bezdotykových měřičů ploch pracujících na základě analogového i číslicového zpracování obrazového signálu kamery, snímající měřený objekt,
- analýza a z ní vyplývající odhad dosažitelné přesnosti měření na základě vyhodnocení systémových i dalších přídavných chyb dílčích funkčních bloků,
- vytvoření uživatelsky příjemného aplikačního softwaru pro vyhodnocení obrazových dat z obrazu měřeného objektu snímaného digitálním fotoaparátem nebo kamerou

Druhá část práce je věnována oblasti stereofotogrammetrie. Je zaměřena na analýzu a vývoj algoritmů pro rekonstrukci prostorových souřadnic snímaných objektů. Na základě podrobného zhodnocení doposud známých postupů automatické kalibrace kamery a ztotožňování bodů na stereosnímčích, představují cíle disertace v této oblasti:

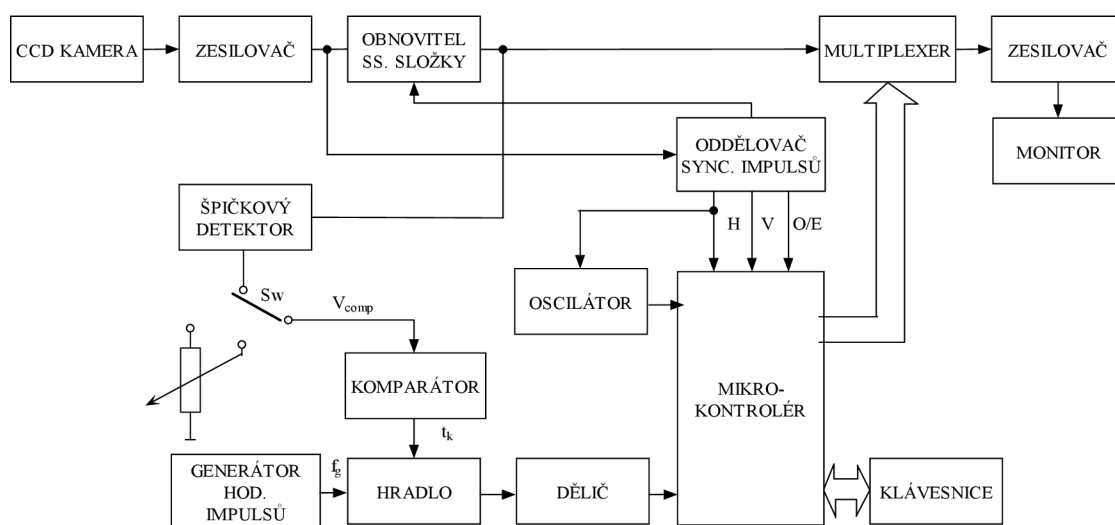
- návrhy originálních přístupů pro určování parametrů orientace kamery a vyhledávání totožných bodů na jednotlivých snímcích nutných pro rekonstrukci scény,
- analýza přesnosti určení prvků vnější a vnitřní orientace spolu s analýzou přesnosti při ztotožňování bodů na snímcích,
- Program pro rekonstrukci prostorových souřadnic snímaných 3D objektů na základě analýzy stereoskopických snímků,
- analýza přesnosti určení parametrů pro nejpoužívanější metody automatické kalibrace a způsoby sesouhlasení bodů na stereoskopických snímcích.

2 Analýza a systémový návrh optoelektronických bezdotykových měřičů ploch

2.1 Měřič s analogovým zpracováním obrazového signálu

Funkce tohoto měřiče vychází z principu rozkladu televizního obrazu, kdy se úroveň jasu na jednotlivých řádcích přeměňuje v časově proměnný signál. Měří se doba t_k , kdy je obrazový signál na úrovni, která překlopí komparátor. Čas je měřen digitálně pomocí čítače. Mírou plochy tedy je počet impulsů kalibračního oscilátoru, které projdou přes hradlo po dobu t_k . Jejich sečtením v čítačích a vzájemným podělením získáme údaj o měřené ploše v procentech referenční plochy. Na základě tohoto byl navržen mikropočítačem řízený měřič ploch, jehož blokové schéma je na Obr.1. Detailní popis lze nalézt v [63], modifikace zapojení synchronizační části pak v [67],[66].

2.1.1 Blokové schéma



Obr. 1. Blokové schéma měřiče s analogovým zpracováním obrazového signálu

2.1.2 Systémová relativní chyba měření

Systémová relativní chyba δ_{vq} vzniká vzorkováním svislého rozměru, tedy řádkovým rozkladem obrazu. Její maximální hodnota dosahuje hodnoty vzorkovacího intervalu, tedy přesně vzdáleností mezi dvěma řádky. Pak absolutní a relativní hodnotu kvantizační chyby

$$\Delta V_q = \frac{V_{\max}}{n_v}, \quad (1.1)$$

$$\delta_{vq} = \frac{\Delta V_q}{V} = \frac{1}{n_v} \frac{V_{\max}}{V}, \quad (1.2)$$

kde n_v je počet řádků na maximálním rozměru V_{\max} ,
 ΔV_q je absolutní chyba svislého rozměru V .

Obdobně lze vyjádřit systémovou relativní chybu δ_{hq} měření, způsobenou rozlišením kamery ve vodorovném směru, které je dáno počtem obrazových elementů a proto obdobně jako pro vertikální směr (rozměr V) bude pro rozměr H platit dle vztahu (1.2) pro relativní chybu

$$\delta_{hq} = \frac{\Delta H_q}{H} = \frac{1}{n_h} \frac{H_{\max}}{H}, \quad (1.3)$$

kde n_h je počet obrazových elementů CCD snímače na maximálním rozměru H_{\max} ,
 ΔH_q je absolutní chyba vodorovného rozměru H .

Systémové chyby δ_{hq} a δ_{vq} jsou nezávislé a proto lze vyjádřit výslednou relativní chybu δ_s (pro měření největší měřitelné obdélníkové plochy $S_{\max} = V \cdot H$) vztahem

$$\delta_s = \frac{\Delta S}{S_{\max}} = \frac{\sqrt{n_v^2 + n_h^2}}{n_h \cdot n_v}. \quad (1.4)$$

Například pro digitální kameru s rozlišením $n_v n_h = 6 \cdot 10^6$ pixelů a poměr stran $n_h/n_v = 1,5$ bude počet pixelů v horizontálním a vertikálním směru $n_v = 3 \cdot 10^3$ pixelu a $n_h = 2 \cdot 10^3$ pixelů. Poté s ohledem ke vztahu (1.4) dostáváme

$$\delta_s = \frac{\sqrt{(3 \cdot 10^3)^2 + (2 \cdot 10^3)^2}}{3 \cdot 10^3 \cdot 2 \cdot 10^3} = 0,37 \cdot 10^{-3}. \quad (1.5)$$

Při měření menších ploch $S_{\text{měř}}$ bude chyba větší (v prvním přiblížení bude lineárně závislá na poměru $S_{\max}/S_{\text{měř}}$). Její hodnota bude také závislá na tvaru a orientaci měřeného povrchu vzhledem k rozkladovému rastru.

2.1.3 Vliv dílčích funkčních částí na výslednou chybu měření

Vliv geometrického zkreslení snímače a použitého objektivu

V případě použití monolitického obrazového snímače CCD (BCCD) a špičkového objektivu je relativní chyba v řádu 10^{-6} a jsou tak zanedbatelné vůči systematické chybě zmíněné výše.

Vliv kolísání frekvence kalibračního oscilátoru f_g

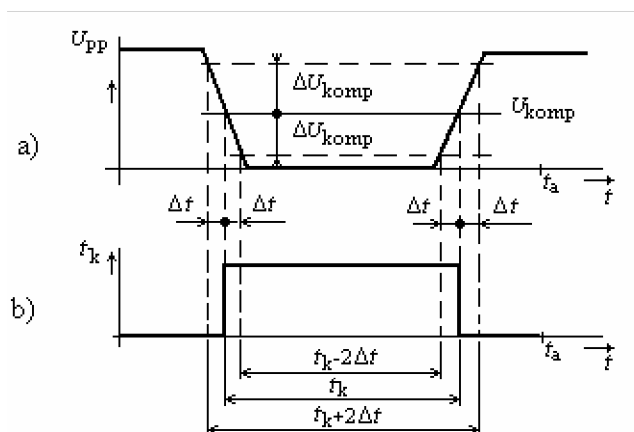
Pro frekvenci tohoto generátoru musí platit vztah

$$f_g \geq 2 \cdot \frac{n_h}{t_{\text{al}}}. \quad (1.6)$$

TV norma CCIR B,D udává pro aktivní dobu řádku $t_{al} = 52 \cdot 10^{-6}$ s. Pak např. pro počet obrazových elementů CCD snímače $n_h = 700$ vychází při dosazení do vztahu (1.6) $f_g \geq 27$ MHz. Uvážíme-li použití generátoru řízeného krystalem, je možné předpokládat relativní kmitočtovou stabilitu $\delta_{fg} = \Delta f_g / f_g \leq \pm 5 \cdot 10^{-5}$. Tento předpoklad byl ověřen měřením funkčního vzorku analogového měřiče.

Vliv kolísání komparačního napětí ΔU_{komp}

Pokud bude docházet k určitému driftu referenčního napětí pro komparátor ΔU_{komp} , bude se vlivem konečné délky náběžné a sestupné hrany impulsu na vstupu komparátoru měnit také doba t_k , po kterou bude sepnut komparátor. Předpokládáme-li dosti kontrastní měřenou scénu s ostrými jasovými přechody objektů v ní a přibližně lineární závislost náběžné a sestupné hrany, lze pro změnu doby Δt v jednom měřeném řádku psát vztah 1.7. Tuto situaci zachycuje obr. 2.



Obr. 2. a) videosignál na vstupu komparátoru, b) signál na výstupu komparátoru

$$\Delta t = \frac{\Delta U_{komp}}{0,8 U_{pp}} \cdot t_r \cong \frac{\Delta U_{komp}}{0,8 U_{pp}} \frac{0,35}{f_0}, \quad (1.7)$$

kde f_0 je horní mezní kmitočet kmitočtové charakteristiky kanálu (zesilovač, obnovitel ss. složky), např. $f_0 = 10$ MHz,

U_{pp} je mezivrcholové napětí za špičkovým usměrňovačem.

Pro celkovou relativní chybu měřeného časového intervalu bude platit v případě maximální měřené plochy

$$\delta_{tkomp} = \frac{\sum \Delta t}{\sum t_k} \cong \frac{2n_v \left(\frac{\Delta U_{komp}}{0,8 U_{pp}} \frac{0,35}{f_0} \right)}{n_v \cdot t_a} \quad (1.8)$$

Vliv kolísání stejnosměrné složky obrazového signálu na výstupu obnovitele ss. složky DCR

Pro změnu intervalu t_k o dobu Δt můžeme použít identicky výsledek jako pro vliv komparátoru, jen místo změny komparačního napětí ΔU_{komp} dosadíme změnu stejnosměrné složky ΔU_{ss} . Pak pro relativní chybu měřeného intervalu t_k , měnící se vlivem posunu ss. složky platí

$$\delta_{\text{tss}} \cong \frac{2n_v \left(\frac{\Delta U_{\text{ss}}}{0,8U_{\text{pp}}} \frac{0,35}{f_0} \right)}{n_v \cdot t_a} \quad (1.9)$$

Z důvodu toho, že jsou výše popsány vlivy opět vzájemně nezávislé, lze pro absolutní hodnotu celkové přídavné relativní chyby měření psát

$$|\delta_{\text{pa}}| = \sqrt{\delta_g^2 + \delta_{\text{tkomp}}^2 + \delta_{\text{tss}}^2} \quad (1.10)$$

Dosadíme-li za členy výrazy ze vztahů 1.8 a 1.9 a budeme uvažovat maximální měřenou plochu, a poměr $\Delta U_{\text{komp}}/U_{\text{pp}} = \Delta U_{\text{ss}}/U_{\text{pp}} = 10^{-2}$, což představuje 1% kolísání, dostaneme pro relativní chybu hodnotu

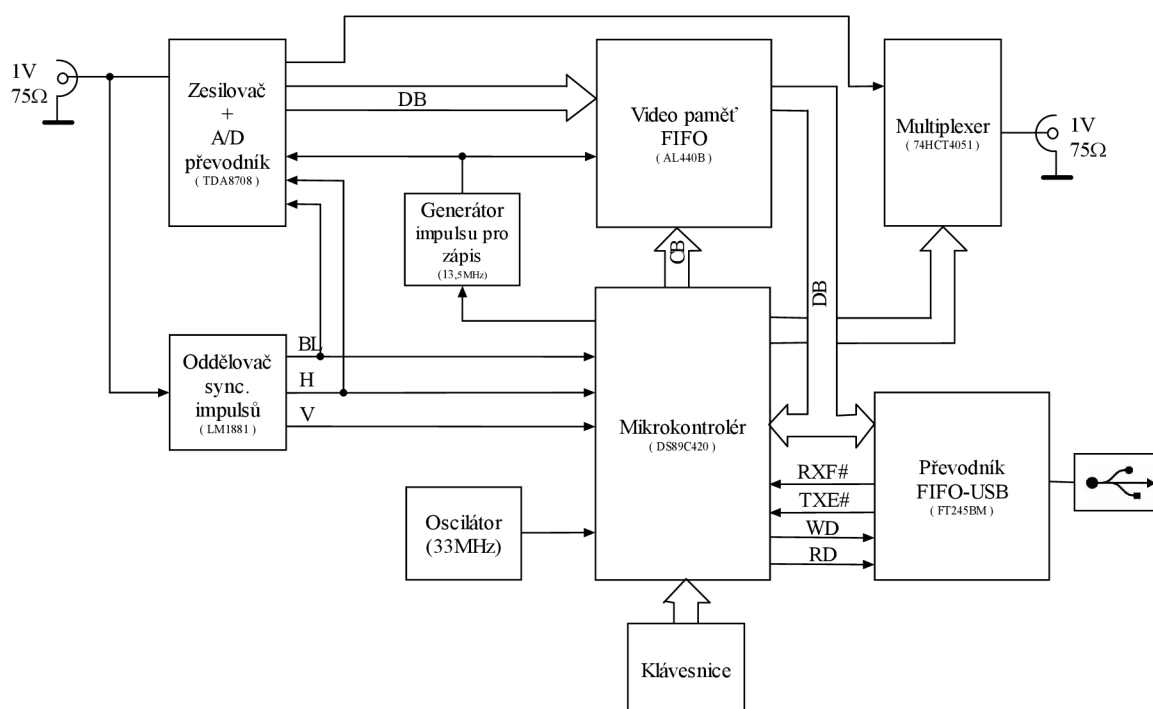
$$|\delta_{\text{pa}}| = 7,6 \cdot 10^{-5} \quad (1.11)$$

2.2 Měřič s číslicovým zpracováním obrazového signálu

Měření plochy se zde vyhodnocuje zcela jinak než v případě analogového zpracování [63]. Analogový obrazový signál z kamery je po obnově stejnosměrné složky (v obvodu vstupního zesilovače) digitalizován v A/D převodníku. Vzorkovaná data v rozlišení např. 8bit/vzorek se ukládají do pulsníkové nebo snímkové paměti. Jsou-li data zapsána v paměti, může procesor provést číslicovou komparaci a vyhodnocení plochy měřeného obrazce, případně uchovaná data upravit. Po získání výsledku měření poté procesor tyto informace předá do obvodu zobrazovače.

Poznámka: Použití digitální kamery je komplikováno skutečností, že výstupní digitální obrazový signál je obvykle komprimován v některém kompresním formátu, který obvykle snižuje informační obsah snímaného obrazu v oblasti vysokých prostorových kmitočtů.

2.2.1 Blokové schéma



Obr. 3. Blokové schéma měřiče s číslicovým zpracováním obraz signálu

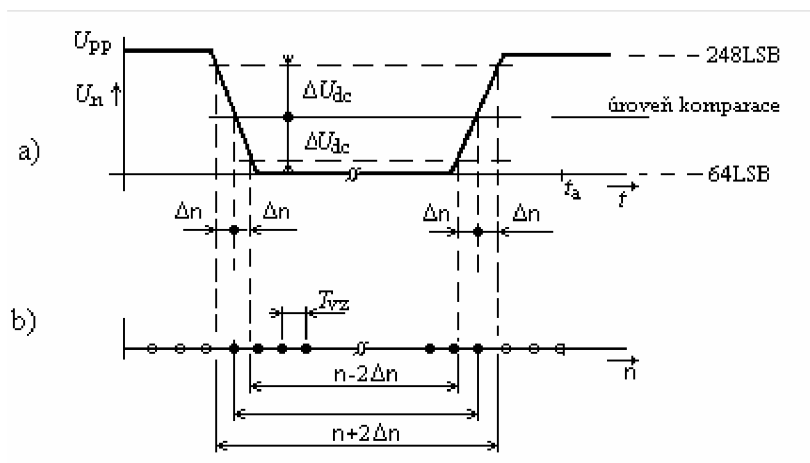
Detailnímu popisu blokového schématu se věnuje [64].

2.2.2 Systémová relativní chyba měření a rozbor vlivu dílčích funkčních bloků

Pro celkovou relativní systémovou chybu měření platí stejné vztahy popisované v kap. 2.1.1. Rovněž vliv kolísání frekvence vzorkovacího generátoru je totožný jako vliv kolísání frekvence kalibračního oscilátoru f_g popisovaného v 2.1.2.

Vliv kolísání stejnosměrné složky obrazového signálu a zesílení předzesilovače

Při kolísání ss. složky či úrovně černé dochází k posouvání videosignálu vůči hodnotě číselné komparace. Tak by mohlo být započítáváno více či méně vzorků do výsledné měřené plochy. Uvážíme-li vnitřní blokovou strukturu předzesilovače, zjistíme, že upínací hodnota napětí je udržována na konstantní hodnotě pomocí smyčky zpětné vazby vedené z výstupu A/D převodníku přes komparátor úrovně černé a odpovídá úrovni 64LSB. Kolísání této hodnoty lze očekávat v rozmezí $\pm 1\text{LSB}$. Situaci ilustruje obr.4. Horní část představuje jeden řádek videosignálu vstupujícího na A/D převodník a konečně spodní část ukazuje vzorky připočítávané do měřené plochy.


Obr. 4. Vliv kolísání stejnosměrné složky ΔU_{dc}

S ohledem na zobrazené můžeme psát pro hodnotu Δn

$$\Delta n = \frac{\Delta U_{dc}}{U_{pp}} \frac{2}{1} = \frac{1\text{LSB} \cdot 2}{248\text{LSB} - 64\text{LSB}} = 1,1 \cdot 10^{-2} \quad (1.12)$$

Celkovou chybu v počtu čítaných vzorků způsobenou kolísáním ss. složky můžeme pro největší měřitelnou plochu vyčíslit jako

$$\delta_{\text{tndc}} = \sum \Delta n / \Delta n = \frac{2n_v \left(\frac{\Delta U_{dc}}{U_{pp}} \right)}{n_v \cdot m} \quad (1.13)$$

Pro hodnoty $n_v = 575$, $f_{vz} = 27\text{MHz}$, $t_{al} = 52 \cdot 10^{-6}\text{ s}$, $m = f_{vz} \cdot t_{al} = 1404$ a kolísání $\Delta U_{dc} = 1\text{LSB}$ dostaneme po dosazení do vztahu hodnotu

$$\delta_{\text{tndc}} = 1,55 \cdot 10^{-5} \quad (1.14)$$

Podobný vztah se dá odvodit i pro vliv kolísání zesílení předzesilovače na počet započítaných vzorků do měřené plochy a proto můžeme počítat s chybou

$$\delta_{nAGC} = 1,55 \cdot 10^{-5} \quad (1.15)$$

Přesnost A/D převodu

Má hodnotu $\pm 1\text{LSB}$ a jsou v ní zahrnuty všechny významné chyby převodníku. Je tvořena kvantizační chybou, která je vlastní každému A/D převodu, jejíž hodnota je $\pm 1/2\text{LSB}$. Další chybou je diferenciální nelinearita, která představuje max. odchylku od ideálního kroku 1LSB, a má hodnotu $\pm 1/2\text{LSB}$.

Aplikujeme-li tuto přesnost na předchozí vztahy pro vliv kolísání parametru na celkový počet čítaných vzorků dospějeme k chybě $\delta_{\text{AD}} = 1,55 \cdot 10^{-5}$.

2.2.3 Odhad celkové chyby měření a porovnání obou metod

Absolutní hodnota přídatné celkové relativní chyby měření pro maximální možnou měřitelnou plochu může být vyjádřena s uvážením vlivů popsaných výše jako

$$|\delta_{\text{pd}}| = \sqrt{(\delta_{\text{fg}})^2 + (\delta_{\text{mdc}})^2 + (\delta_{\text{nAGC}})^2 + (\delta_{\text{AD}})^2} \quad (1.16)$$

A po dosazení

$$|\delta_{\text{pd}}| = \sqrt{(5 \cdot 10^{-5})^2 + (1,55 \cdot 10^{-5})^2 + (1,55 \cdot 10^{-5})^2 + (1,55 \cdot 10^{-5})^2} = 5,7 \cdot 10^{-5} \quad (1.17)$$

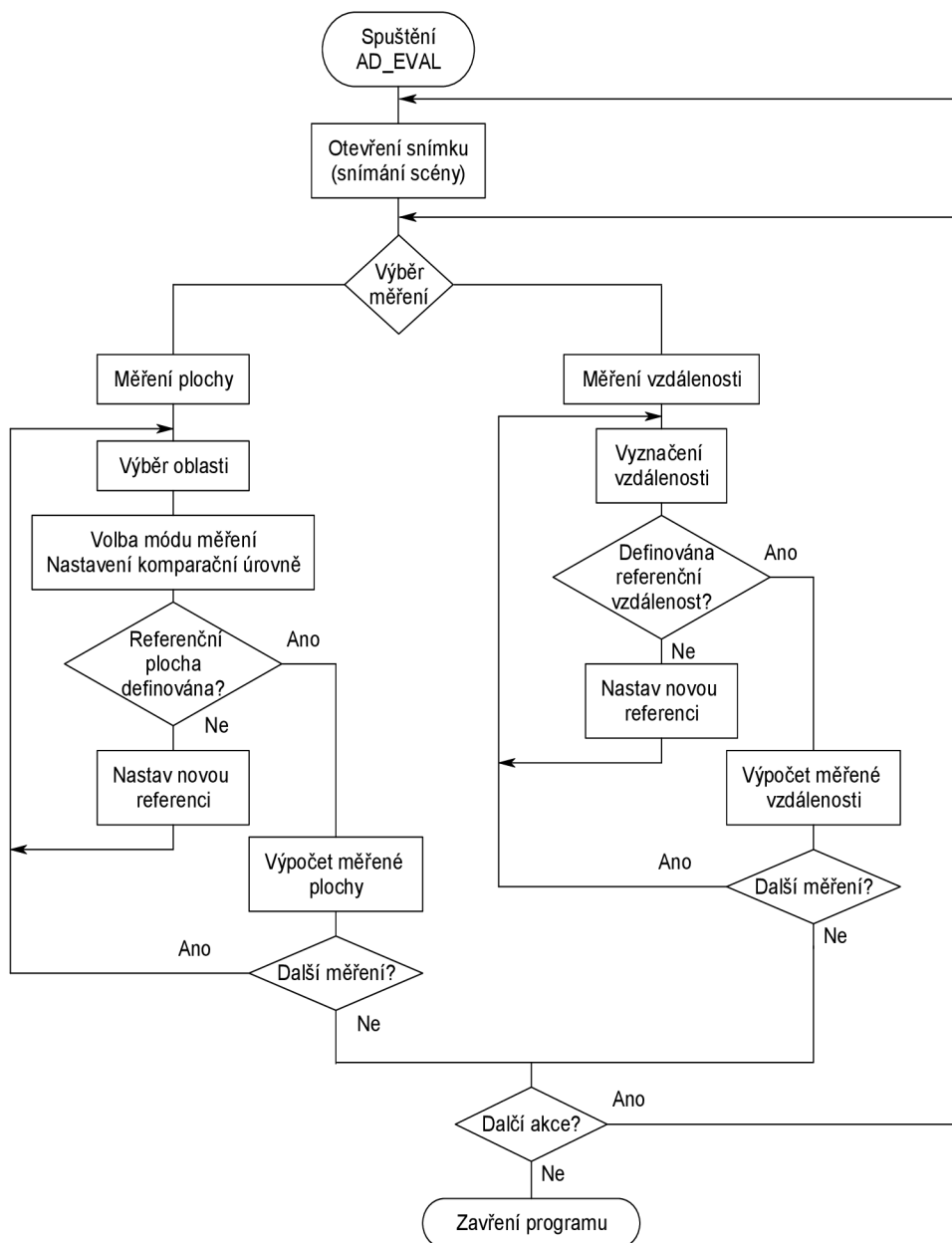
Je zřejmé, že přídatná relativní chyba vzrůstá při měření ploch zabírající menší plochu senzoru. Nicméně je tato chyba pro největší měřitelnou oblast stokrát menší než systematická relativní chyba δ_s . Optické zvětšení měřené oblasti je nastaveno prostřednictvím vhodného objektivu ve skutečných podmínkách tak, aby měřená oblast zakrývala 50% rastru snímače. V těchto podmínkách systematická relativní chyba δ_s představuje, evidentně, limit dosažitelného přesnosti měření. Žádané zvýšení přesnosti měření může být dosaženo jen aplikací zvláštní optoelektronického snímače s vysokým rozlišením.

Při snaze o vyčíslení celkové chyby měření u analogového i digitálního měřiče ploch docházíme k závěru, že digitalizace analogového videosignálu nepřináší výrazné zlepšení výsledné přesnosti měření. Výhoda spočívá především v číslicovém zpracování obrazu, v možnosti velkého počtu způsobů vyhodnocování velikosti měřené plochy v případě, že data by byla přenesena do PC. Tato možnost se jeví jako docela jednoduchá, při doplnění měřiče o sériové rozhraní a jeho využití pro zachytávání obrazu.

2.3 Program AD-EVAL pro měření ploch a rozměrů

Vytvořený program AD-EVAL slouží k softwarovému změření velikosti plochy, případně vzdálenosti mezi dvěma zvolenými body v plošné projekci snímané 2D nebo 3D scény, kterou představuje digitální snímek pořízený jakýmkoliv snímacím zařízením, např. digitálním fotoaparátem. Jelikož se využívá poměrového vyhodnocení, předpokládáme scénu s rovinnými objekty a optickou osu objektivu kolmo na ni. K vytvoření uživatelsky příjemného prostředí bylo vybráno vývojové prostředí Builder C++. Obrázek 5 představuje vývojový diagram programu AD-EVAL.

2.3.1 Vývojový diagram programu



Obr. 5 Vývojový diagram programu AD-EVAL

Spuštění programu

Po spuštění programu AD_EVAL lze otevřít snímek uložený v PC. Podporovaným formátem je BMP s 8 bitovou hloubkou na pixel. Při získávání dat z digitálního fotoaparátu či digitální kamery se využívá API (*application programming interface*) funkcí implementovaných v knihovnách DLL či LIB dodávaných k ovladačům či SDK (*Software Developer Kit*) jednotlivých zařízení. Pomocí těchto API funkcí se přes ovladač komunikuje s příslušným zařízením a je možné uložit snímanou scénu do souboru či paměti PC.

Po otevření snímku či naskenování scény volíme typ měření, tedy měření velikosti plochy či měření vzdálenosti mezi dvěma body. Defaultním módem je měření velikosti ploch. Není-li požadována další akce, lze program ukončit.

Měření velikosti ploch

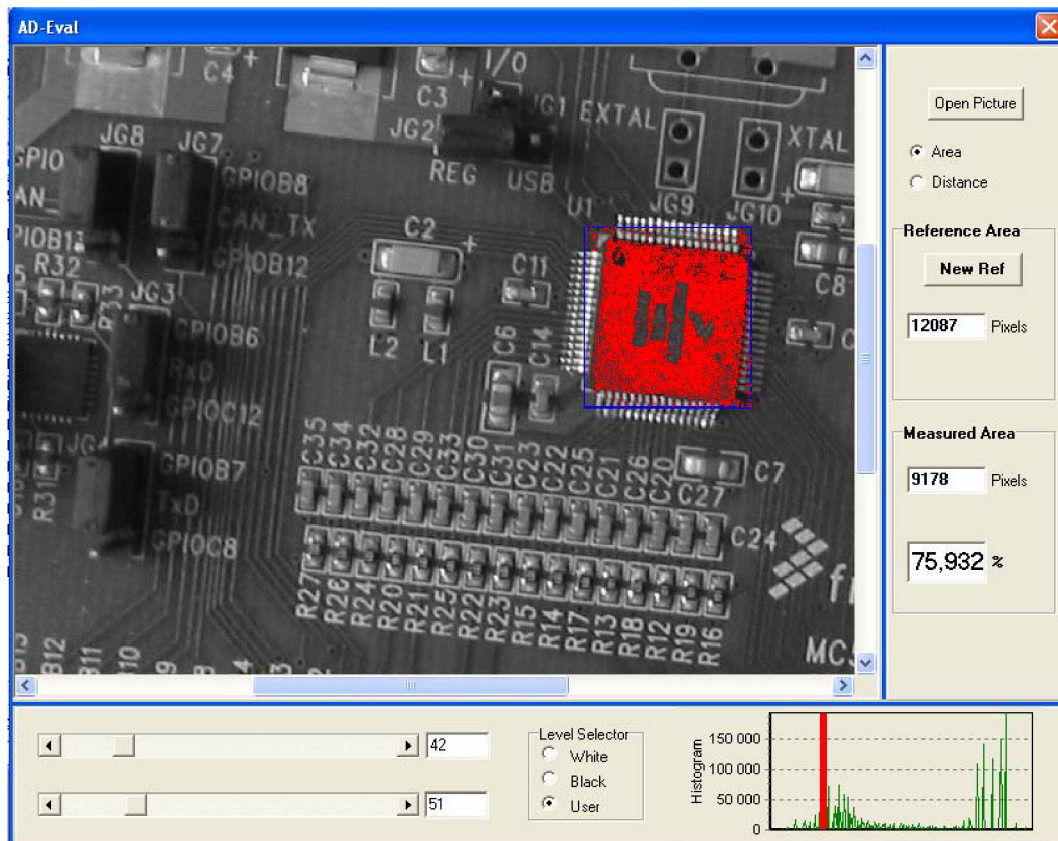
Při měření velikosti ploch je využito poměrové měření. Je proto nutné zvolit referenční plochu známé velikosti a k této je měřená plocha vztažena. Pro výběr části snímku se využívá nastavitelného rámečku. Zvolí se mód měření (bílá, černá, uživatel) a nastaví se prahová úroveň pro měření. Pokud ještě nebyla určena referenční plocha, nastaví se zvolená plocha jako referenční. Opět se provede selekce pro požadovanou část snímku, případně se upraví prahová úroveň a vykoná se kalkulace velikosti zvolené plochy poměrem k ploše referenční. Celý cyklus lze libovolně opakovat.

Měření vzdálenosti mezi 2 body

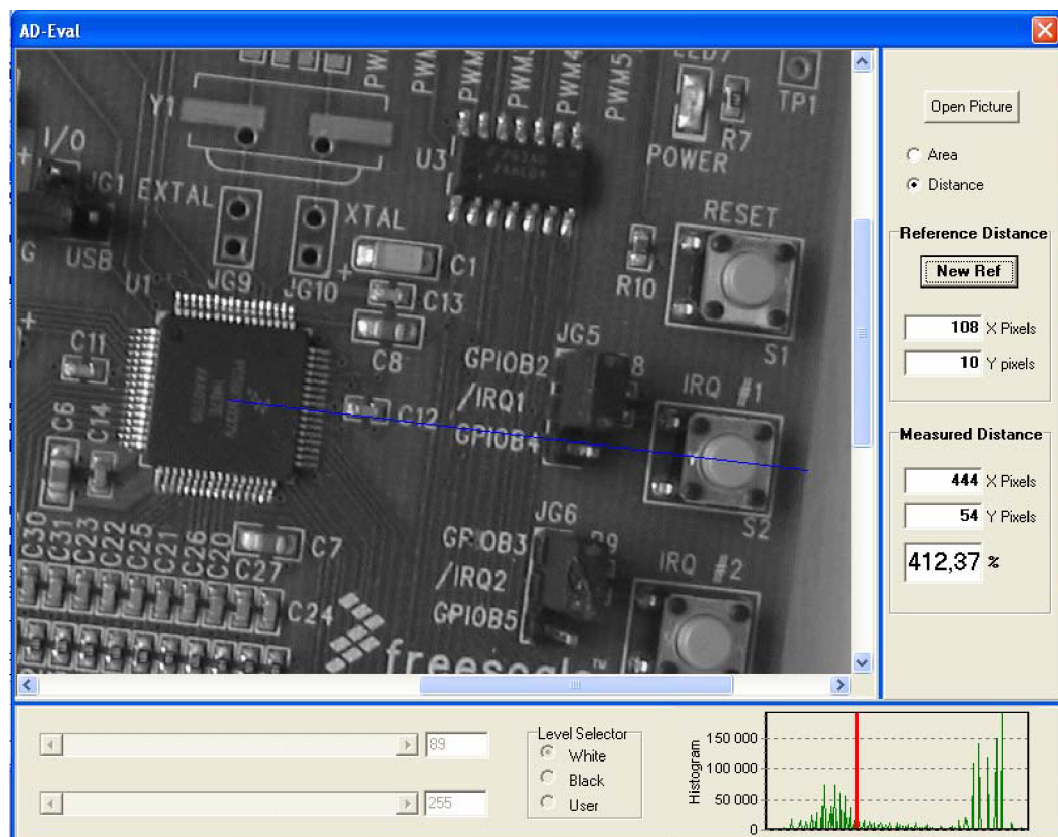
Měření vzdálenosti probíhá opět na základě poměrového měření. Znamou vzdálenost označíme na snímku dvěma body, nastavíme ji jako referenční vzdálenost. Neznámá vzdálenost je po opětovném označení vyhodnocena jako poměrný údaj vzhledem ke známé (referenční) vzdálenosti.

2.3.2 Komunikační okna ovládacího programu

Vzhled ovládacího programu je na Obr.6 a představuje situaci při měření velikosti plochy. Okno programu lze rozdělit do tří částí. První a největší část zabírá snímaná scéna. Zde lze po natažení obrázku pomocí myši volit oblast měření, ve které je automaticky zvýrazněna měřená plocha. Spodní část okna obsahuje přepínač módu měření (černá, bílá, uživatel) a posuvníky, sloužící k nastavení komparační úrovně pro měření. V pravé dolní části pak histogram zobrazeného obrázku. Poslední třetí část okna je vyčleněna pro zobrazení výsledku měření a obsahuje hodnoty „načtených“ pixelů referenční i měřené plochy a výslednou procentuální hodnotu měřené plochy k ploše referenční. V případě měření vzdálenosti je volba komparační úrovně a módu měření neaktivní. Zobrazované údaje se vztahují k měřené vzdálenosti mezi 2 body.




Obr. 6. Vytvořený ovládací program AD-EVAL, měření plochy.



Obr.7. Program AD-EVAL, měření vzdálenosti mezi 2 body

2.3.3 Základní ovládání programu AD-EVAL

Otevření snímku

Otevření snímku lze provést kliknutím na tlačítko . V následném okně používající náhled lze vybrat snímek. Povoleným formátem je BMP v odstínu šedi (8 bitová hloubka na pixel). Pokud je otevřený snímek větší než zobrazovací oblast, aktivují se posuvníky po stranách snímku, umožňující posunutí snímku.


Volba selektivního rámečku

Rámeček vytvoříme klasicky pomocí myši. Kliknutím na levé tlačítko zadáme první roh rámečku, při následném pohybu myši se rámeček vykresluje. Uvolněním levého tlačítka se zadá protější roh rámečku. Při najetí kurzoru na rámeček a stisku levého tlačítka je možné rámečkem pohybovat přes snímek a volit tak měřený objekt. Rovněž při najetí na okraj rámečku lze při současném stisku levého tlačítka a pohybu myši měnit jeho velikost.

Volba prahové úrovně

Volbu úrovně, při které se jednotlivé pixely ještě započítávají do měřené plochy, provádíme pomocí voliče módu měření (černá, bílá, uživatel) a nastavením prahové úrovně příslušným posuvníkem či zadáním konkrétní hodnoty v editačním poli. Nastavovaná prahová úroveň se vykresluje také na histogramu. Změna prahové úrovně je doprovázena zvýrazněním pixelů ve zvoleném rámečku, které jsou zahrnuty do měření.

Zadání referenční plochy

Je-li zvolen selektivní rámeček, lze pomocí tlačítka  uložit právě vybranou a odprahovanou část snímku jako referenční plochu, přičemž se v zobrazovacím poli objeví počet pixelů reprezentující tuto referenční plochu.

Vyhodnocení měřené plochy

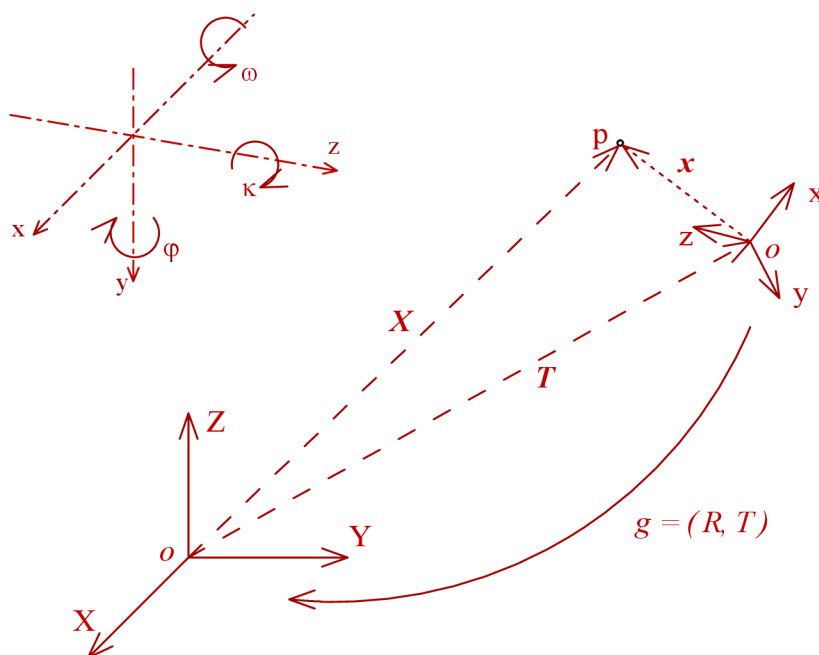
Při pohybu rámečkem či po zadání nového, se vypisuje počet pixelů měřené plochy do boxu *Measured area*. Pokud již byla zadána referenční plocha, je současně zobrazován údaj o relativním poměru měřené plochy ke zvolené referenci.

3 Rekonstrukce prostorových souřadnic na základě stereofotogrammetrie

Určení třírozměrných dat z dvourozměrných snímků je stěžejním problémem počítačového vidění a fotogrammetrie. Jednou z mnoha cest k získání prostorové scény ze snímků je stereovize. Ta má uplatnění v širokém okruhu oblastí, například tvorba třídímenzionálních map, prostorová vizualizace objektů, prostorová navigace robotů, a tak dále. Pro 3D rekonstrukci jsou charakteristické dvě základní operace – kalibrace kamery a sesouhlasení bodů na snímcích. Cílem procesu kalibrace kamery je nalezení projekce ze prostorových souřadnic do souřadnic snímku. K řešení korespondenčního problému může být využito velké množství omezení. Například, je-li známa kalibrace kamery, je možné využít epipolárního omezení a tímto proces sesouhlasení bodů zjednodušit. V následujících kapitolách budou popsány právě tyto dvě stěžejní operace, spolu s nezbytným matematickým popisem transformací v prostoru a geometrickým modelem kamery, přičemž je využito poznatků uvedených v [42].

3.1 Matematické transformace v prostoru

Tyto matematické transformace se zabývají popisem prostorových vztahů a změnou souřadnic bodu v prostoru při natočení či posunu kamery. Na Obr.8 je naznačena zvolená souřadnicová soustava. Úhel rotace kolem osy x je ω , úhel rotace kolem osy y je φ a konečně úhel rotace kolem osy z je κ



Obr. 8. Transformace bodu p mezi souřadným systémem xyz a XYZ

Rotace a posunutí bodu v prostoru

Mějme bod p se souřadnicemi $\mathbf{x}=[x,y,z]^T$ v prostoru xyz . Tyto souřadnice jsou rotovány třemi úhly relativně k systému XYZ . Pro souřadnici $\mathbf{X}=[X,Y,Z]^T$ bodu p v systému XYZ můžeme psát následující maticovou rovnici

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (3.1)$$

nebo vektorově

$$\mathbf{X} = \mathbf{R} * \mathbf{x}. \quad (3.2)$$

kde \mathbf{R} ... ortonormální matice rotace 3x3

Matici \mathbf{R} lze vyjádřit následovně pomocí rotačních úhlů

$$\mathbf{R} = \mathbf{R}_\omega \cdot \mathbf{R}_\varphi \cdot \mathbf{R}_\kappa, \quad (3.3)$$

kde

$$\mathbf{R}_\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \quad (3.4)$$

$$\mathbf{R}_\varphi = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \quad (3.5)$$

$$\mathbf{R}_\kappa = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Operaci posunutí můžeme napsat vektorově

$$\mathbf{X} = \mathbf{x} + \mathbf{T}, \quad (3.7)$$

kde \mathbf{T} je vektor translace a platí pro něj

$$\mathbf{T} = T_x + T_y + T_z = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (3.8)$$

Operaci současné rotace a translace je pak možné zapsat jako

$$\mathbf{X} = \mathbf{R} \mathbf{x} + \mathbf{T} \quad (3.9)$$

Nebo jednoduše $\mathbf{X} = g\mathbf{x}$, kde $g = (\mathbf{R}, \mathbf{T})$ a představuje právě transformaci souřadnic bodu mezi dvěma systémy. Je zřejmé, že na rozdíl od obyčejné rotace celková transformace není lineární, ale afinní. Nicméně je možné přepsat tuto afinní transformaci na lineární s použitím homogenních souřadnic:

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \bar{g} \bar{\mathbf{x}} \quad (3.10)$$

kde matice velikosti 4x4 značí homogenní reprezentaci transformace $g = (\mathbf{R}, \mathbf{T})$.

Obecně, jestliže je $g = (\mathbf{R}, \mathbf{T})$, pak jeho homogenní reprezentace je

$$\bar{g} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \quad (3.11)$$

3.2 Geometrický model kamery

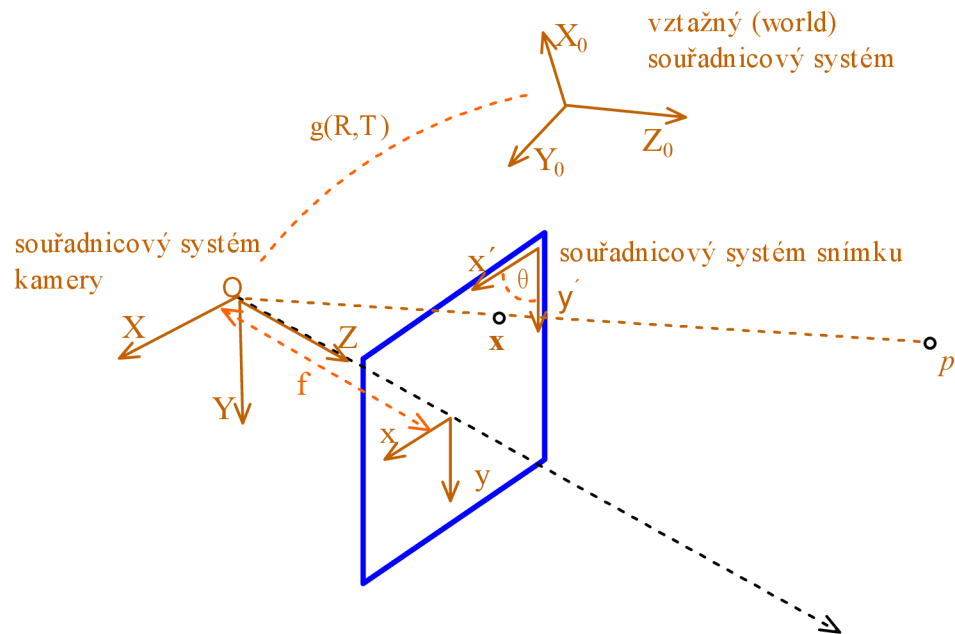
Pro popis vztahu mezi 2D a 3D prostorem je využíván geometrický model kamery, v tomto případě model označovány jako *pinhole camera model* (Obr.9). Pro tento model kamera provádí ideální perspektivní projekci. Předpokládejme obecný bod p se souřadnicemi $X_0 = [X_0, Y_0, Z_0]^T \in \mathbb{R}^3$ vztažený k referenční souřadné soustavě. Souřadnice $X = [X, Y, Z]^T$ téhož bodu vztažený k souřadné soustavě kamery je dána transformací $g = (\mathbf{R}, \mathbf{T})$ popsanou výše a sice

$$X = \mathbf{R}X_0 + \mathbf{T} \in \mathbb{R}^3. \quad (3.12)$$

Bod X se promítne do roviny kamery v bodě

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix}, \quad (3.13)$$

kde f je ohnisková vzdálenost.



Obr. 9 Geometrický model kamery

V homogenních souřadnicích lze tento vztah zapsat jako

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.14)$$

Předchozí vztah můžeme přepsat také jako

$$Z \mathbf{x} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}, \quad (3.15)$$

kde $\mathbf{X} = [X, Y, Z, 1]^T$ a $\mathbf{x} \doteq [x, y, 1]^T$ jsou teď v homogenním vyjádření.

Protože souřadnice Z není obvykle známa, můžeme ji zapsat jako libovolný pozitivní skalár $\lambda \in \mathbb{R}_+$. Matici z předešlého vztahu je možné rozložit

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3.16)$$

Definujme dvě matice

$$\mathbf{K}_f \doteq \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad \mathbf{\Pi}_0 \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 4}. \quad (3.17)$$

Matice $\mathbf{\Pi}_0$ se často označuje jako standardní matice projekce. Pomocí transformace souřadnic máme pro bod $\mathbf{X} = [X, Y, Z, 1]^T$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}. \quad (3.18)$$

S použitím výše uvedeného zápisu můžeme souhrnně psát pro celkový geometrický model ideální kamery

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \quad (3.19)$$

nebo v maticové formě

$$\lambda \mathbf{x} = \mathbf{K}_f \mathbf{\Pi}_0 \mathbf{X} = \mathbf{K}_f \mathbf{\Pi}_0 \mathbf{g} \mathbf{X}_0. \quad (3.20)$$

Jestliže je ohnisková vzdálenost f známa, může být normalizována, pak je tento

model redukován na euklidovskou transformaci g a následnou standardní projekcí $\mathbf{\Pi}_0$, tzn.

$$\lambda \mathbf{x} = \mathbf{\Pi}_0 \mathbf{X} = \mathbf{\Pi}_0 g \mathbf{X}_0. \quad (3.21)$$

Ideální model je specifikován vzhledem k referenčnímu snímku centrovanému v optickém středu a osou souběžnou s optickou osou. V praxi, když je pořízen snímek s digitální kamerou, je měření získáváno v pixelech s počátkem souřadnic snímku typicky v levém horním rohu snímku. Aby byl ideální model použitelný přepíšeme jej jako

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.22)$$

z čehož můžeme definovat

$$\mathbf{K} \doteq \mathbf{K}_s \mathbf{K}_f \doteq \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.23)$$

Sloučením \mathbf{K}_f a \mathbf{K}_s můžeme přepsat vztah pro projekci následovně,

$$\lambda \mathbf{x}' = \mathbf{K} \mathbf{\Pi}_0 \mathbf{X} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.24)$$

Matice \mathbf{K} obsahuje všechny parametry které jsou vlastní specifické kamery a proto se nazývá matice prvků vnitřní orientace nebo také kalibrační matice kamery. Jednotlivé prvky matice mají následující význam:

- f : ohnisková vzdálenost
- o_x : x-ová souřadnice středu snímku v pixelech
- o_y : y-ová souřadnice středu snímku v pixelech
- s_x : velikost pixelu v horizontálním směru
- s_y : velikost pixelu ve vertikálním směru
- s_θ : šikmost pixelu, často blízko nule.

Pokud je kalibrační matice známa, mohou být kalibrované souřadnice získány ze souřadnic v pixelech jednoduše inverzí matice \mathbf{K} ,

$$\lambda \mathbf{x} = \lambda \mathbf{K}^{-1} \mathbf{x}' = \mathbf{\Pi}_0 \mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.25)$$

Informace o matici \mathbf{K} je získána při procesu kalibrace kamery, která bude popsána později.

Geometrický vztah mezi bodem se souřadnicemi $X_0 = [X_0, Y_0, Z_0]^T$ vzhledem ke zvolenému souřadnému systému a odpovídajícím souřadnicím $x' = [x', y', 1]^T$ na snímku (v pixelech) závisí na transformaci (\mathbf{R}, \mathbf{T}) , někdy označovanou jako prvky vnější orientace, ideální projekci $\mathbf{\Pi}_0$ a prvkům vnitřní orientace \mathbf{K} . Celkový model pro vytvoření obrazu je proto zachycen následujícím vztahem

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\Theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \quad (3.26)$$

nebo v maticové formě

$$\lambda x' = \mathbf{K} \mathbf{\Pi}_0 X = \mathbf{K} \mathbf{\Pi}_0 g X_0. \quad (3.27)$$

3.3 Rekonstrukce ze dvou kalibrovaných snímků

Tato kapitola popisuje základní geometrii dvou snímků, zvanou epipolární geometrie. Počátky epipolární geometrie sahají do poloviny 19.století do práce Hesse, jenž studoval geometrii dvou snímků s použitím sedmi bodů (více v [47]). Kruppa dokázal že 5 bodů v obecné poloze je dostačující pro vyřešení geometrii dvou snímků až na konečný počet řešení. Jeho důkazy byly později zdokonaleny a byl prokázán konkrétní počet řešení s jednoduchým důkazem daným později v [26]. Praktický důkaz lze nalézt v [53] s konkrétním lineárním algoritmem používajícím šest sesouhlasených bodů.

Epipolární omezení se poprvé objevilo v [76]. Osmi bodový algoritmus byl dán v [37] a [28], který vyvolal v této oblasti široký zájem, což vedlo k vývoji řady lineárních a nelineárních algoritmů pro odhad pohybu ze dvou snímků. První práce lze nalézt v [15],[34],[48]. Zdokonalení osmi bodového algoritmu založeného na normalizovaných snímkových souřadnicích byl dán v [24].

3.3.1 Epipolární geometrie, esenciální matice

Předpokládejme dva snímky stejné scény pořízené z různých bodů. Jestliže předpokládáme, že kamera je zkalibrována (kalibrační matice \mathbf{K} je jednotková), homogenní snímkové souřadnice x a prostorová souřadnice X bodu p jsou dány vztahem

$$\lambda x = \mathbf{\Pi}_0 X, \quad (3.28)$$

kde $\mathbf{\Pi}_0 = [\mathbf{I}, 0]$. To znamená, že se bod na snímku liší od skutečné prostorové souřadnice bodu o neznámý rozměr λ . Jestliže uvážíme body x_1, x_2 jako totožné body na dvou snímcích, pak budou tyto body popsány přesným geometrickým vztahem.

Vztah mezi 3-D souřadnicí bodu ve vztažném souřadném systému a souřadným systémem kamery může být vyjádřen transformací $g = (\mathbf{R}, \mathbf{T})$. Předpokládejme vztažný systém totožný s jednou kamerou. Označíme-li prostorové souřadnice bodu p vzhledem ke snímkům kamer $X_1 \in \mathbb{R}^3$ a $X_2 \in \mathbb{R}^3$ pak je vztah mezi nimi dán transformací

$$X_2 = \mathbf{R}X_1 + \mathbf{T}. \quad (3.29)$$

Nyní předpokládejme $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ jako homogenní souřadnice projekce stejného bodu p na snímek kamery. Protože $\mathbf{X}_i = \lambda_i \mathbf{x}_i, i=1,2$, může být vztah přepsán

$$\lambda_2 \mathbf{x}_2 = \mathbf{R} \lambda_1 \mathbf{x}_1 + \mathbf{T}. \quad (3.30)$$

Pokud eliminujeme rozměr λ_i vynásobením obou stran rovnice vektorem \mathbf{T} , dostáváme

$$\lambda_2 \hat{\mathbf{T}} \mathbf{x}_2 = \hat{\mathbf{T}} \mathbf{R} \lambda_1 \mathbf{x}_1. \quad (3.31)$$

Jelikož vektor $\hat{\mathbf{T}} \mathbf{x}_2 = \mathbf{T} \times \mathbf{x}_2$ je kolmý k \mathbf{x}_2 , vnitřní produkt $\langle \mathbf{x}_2, \hat{\mathbf{T}} \mathbf{x}_2 \rangle = \mathbf{x}_2^T \hat{\mathbf{T}} \mathbf{x}_2$ je nulový. Vynásobení předchozího vztahu \mathbf{x}_2^T dává nulový taktéž $\mathbf{x}_2^T \hat{\mathbf{T}} \mathbf{R} \lambda_1 \mathbf{x}_1$. Jelikož $\lambda_1 > 0$ dokázali jsme následující.

Předpokládejme dvě souřadnice na snímku $\mathbf{x}_1, \mathbf{x}_2$ téhož bodu ze dvou pozic kamery se vtahem (\mathbf{R}, \mathbf{T}) , pak $\mathbf{x}_1, \mathbf{x}_2$ splňují podmínku

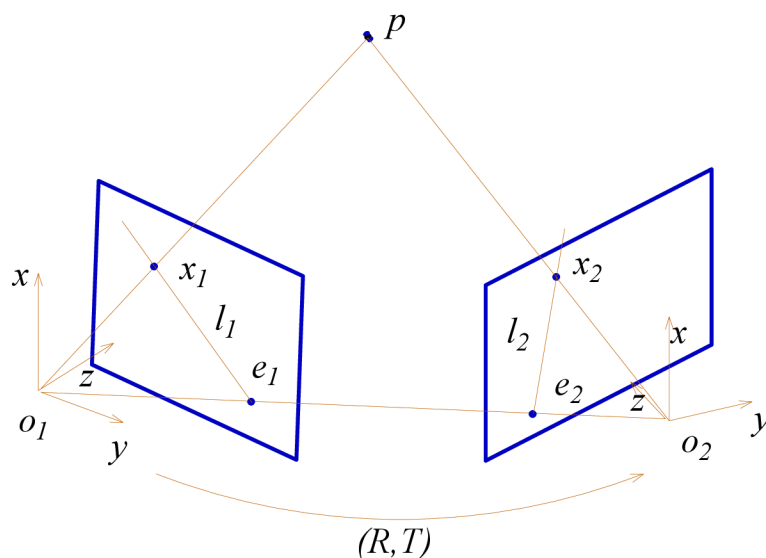
$$\langle \mathbf{x}_2, \mathbf{T} \times \mathbf{R} \mathbf{x}_1 \rangle = 0, \text{ nebo } \mathbf{x}_2^T \hat{\mathbf{T}} \mathbf{R} \lambda_1 \mathbf{x}_1 = 0. \quad (3.32)$$

Matice

$$\mathbf{E} = \hat{\mathbf{T}} \mathbf{R} \in \mathbb{R}^{3 \times 3} \quad (3.33)$$

se ve vztahu určujícím epipolární omezení nazývá *esenciální* matice. Zakódovává relativní pozici mezi dvěma kamerami.

Toto omezení vychází přímo z geometrických úvah a je naznačeno na Obr. 10.



Obr.10. Epipolární geometrie

S přihlédnutím k tomuto obrázku, definujeme následující geometrické objekty:

- rovina (o_1, o_2, p) určená dvěma centry projekce a bodem p se nazývá epipolární rovina. Existuje jedna rovina pro každý bod p ,
- projekce e_1 (e_2) středu projekce jedné kamery na snímek druhé kamery se nazývá epipola,
- průsečík epipolární roviny s rovinou snímku je přímka l_1 (l_2), nazývaná epipolární přímka.

Epipolární omezení je dobře známé ve stereovizi: pro každý bod \mathbf{x}_1 na prvním snímku, totožný bod \mathbf{x}_2 na druhém snímku leží na epipolární přímce l_2 .

3.3.2 Osmi-bodový lineární algoritmus

Nechť $\mathbf{E} = \hat{\mathbf{T}}\mathbf{R}$ je esenciální matice související s epipolárním omezením. Jednotlivé prvky matice jsou označeny jako

$$\mathbf{E} \doteq \begin{bmatrix} \mathbf{e}_{11} & \mathbf{e}_{12} & \mathbf{e}_{13} \\ \mathbf{e}_{21} & \mathbf{e}_{22} & \mathbf{e}_{23} \\ \mathbf{e}_{31} & \mathbf{e}_{32} & \mathbf{e}_{33} \end{bmatrix} \quad (3.34)$$

a srovnány do vektoru $\mathbf{E}^8 \in \mathbb{R}^9$, pak

$$\mathbf{E} = [\mathbf{e}_{11}, \mathbf{e}_{21}, \mathbf{e}_{31}, \mathbf{e}_{12}, \mathbf{e}_{22}, \mathbf{e}_{32}, \mathbf{e}_{13}, \mathbf{e}_{23}, \mathbf{e}_{33}]^T \in \mathbb{R}^9. \quad (3.35)$$

Dále definujeme Kroneckeruv součin \otimes dvou vektorů \mathbf{x}_1 a \mathbf{x}_2 jako

$$\mathbf{a} \doteq \mathbf{x}_1 \otimes \mathbf{x}_2. \quad (3.36)$$

Pokud $\mathbf{x}_1 = [x_1, y_1, z_1]^T \in \mathbb{R}^3$ a $\mathbf{x}_2 = [x_2, y_2, z_2]^T \in \mathbb{R}^3$ pak

$$\mathbf{a} = [x_1x_2, x_1y_2, x_1z_2, y_1x_2, y_1y_2, y_1z_2, z_1x_2, z_1y_2, z_1z_2]^T \in \mathbb{R}^9. \quad (3.37)$$

Jelikož je epipolární omezení $\mathbf{x}_2^T \mathbf{E} \mathbf{x}_1 = 0$ lineární vzhledem k prvkům matice \mathbf{E} , použitím výše uvedeného vztahu jej můžeme přepsat jako vnitřní produkt matic \mathbf{a} a \mathbf{E}^8

$$\mathbf{a}^T \mathbf{E}^8 = 0. \quad (3.38)$$

Nyní daný soubor sesouhlasených bodů na snímcích $(\mathbf{x}_1^j, \mathbf{x}_2^j)$, pro $j = 1, 2, \dots, n$, definuje matici $\mathbf{D} \in \mathbb{R}^{n \times 9}$ jako

$$\mathbf{D} = [\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n]^T, \quad (3.39)$$

kde j -tý řádek \mathbf{a}^j je kroneckeruv součin jednotlivých párů $(\mathbf{x}_1^j, \mathbf{x}_2^j)$.

V případě absence šumu vektor \mathbf{E}^8 splňuje rovnici

$$\mathbf{D} \mathbf{E}^8 = 0. \quad (3.40)$$

Tato lineární rovnice může být vyřešena pro vektor \mathbf{E}^8 . Aby řešení bylo jednoznačné musí být hodnota matice $\mathbf{D} \in \mathbb{R}^{n \times 9}$ přesně osm.

Celkový algoritmus, popsáný v [37], pak vypadá takto:

Pro daný soubor sesouhlasených bodů na snímcích $(\mathbf{x}_1^j, \mathbf{x}_2^j)$, pro $j = 1, 2, \dots, n$ ($n > 8$) tento algoritmus obnoví (\mathbf{R}, \mathbf{T}) odpovídající

$$\mathbf{x}_2^{jT} \hat{\mathbf{T}} \mathbf{R} \lambda_1 \mathbf{x}_1^j = 0, \quad j = 1, 2, \dots, n \quad (3.41)$$

1. Sestavíme $\mathbf{D} = [\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n]^T \in \mathbb{R}^{n \times 9}$ z korespondenčních bodů $(\mathbf{x}_1^j, \mathbf{x}_2^j)$.

Nalezneme vektor \mathbf{E}^8 jednotkové délky tak, aby $\|\mathbf{D} \mathbf{E}^8\|$ byl minimalizován následovně:

Vypočítáme SVD $\mathbf{D} = \mathbf{U}_D \mathbf{\Sigma}_D \mathbf{V}_D^T$ a definujeme \mathbf{E}^8 jako devátý sloupec \mathbf{V}_D .

Sestavíme zpět devět elementů vektoru \mathbf{E}^8 do matice 3×3 .

2. Vypočteme \mathbf{E} jako $\mathbf{E} = \mathbf{U} \text{diag}\{\sigma_1, \sigma_2, \sigma_3\} \mathbf{V}^T$.

3. Obnovíme posunutí z esenciální matice. Je potřeba pouze \mathbf{U} a \mathbf{V} k získání \mathbf{R} a \mathbf{T} z esenciální matice jako

$$\mathbf{R} = \mathbf{U} \mathbf{R}_z^T \left(\pm \frac{\pi}{2} \right) \mathbf{V}^T, \quad \hat{\mathbf{T}} = \mathbf{U} \mathbf{R}_z \left(\pm \frac{\pi}{2} \right) \mathbf{\Sigma} \mathbf{V}^T,$$

$$\text{kde } \mathbf{R}_z^T \left(\pm \frac{\pi}{2} \right) \doteq \begin{bmatrix} 0 & \pm 1 & 0 \\ \pm 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

3.4 Rekonstrukce ze dvou nekalibrovaných snímků

V předchozí kapitole bylo ukázáno, že projekce bodu v prostoru se souřadnicemi \mathbf{X} do snímku kamery má souřadnice \mathbf{x}' , které vyhovují rovnici

$$\lambda \mathbf{x}' = \mathbf{K} \mathbf{\Pi}_0 g \mathbf{X}_0 = \mathbf{K} [\mathbf{R}, \mathbf{T}] \mathbf{X}. \quad (3.42)$$

Matice \mathbf{K} definovaná jako

$$\mathbf{K} = \begin{bmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad (3.43)$$

popisuje vnitřní parametry kamery, jako je ohnisková vzdálenost f , pozice středu snímku (o_x, o_y) , velikost pixelů (s_x, s_y) a jejich šikmost. Nazývá se maticí vnitřních parametrů, nebo jednoduše kalibrační matice. Převádí metrické souřadnice na snímkové souřadnice. Transformace $g = (\mathbf{R}, \mathbf{T})$ reprezentuje vnější parametry kamery, tzn. její polohu a orientaci vůči zvolené vztažné souřadné soustavě.

V případě kalibrovaných kamer je kalibrační matice jednotková, $\mathbf{K} = \mathbf{I}$ a dvojice snímků s dostatečným počtem sesouhlasených bodů postačí k rekonstrukci polohy kamer a následně bodu v prostoru. Jestliže \mathbf{K} jednotková není, přesto je však známa, je možné problém přeformulovat tak, aby mohl být kalibrovaný stav opět aplikován. A to sice vynásobením rovnice z obou stran \mathbf{K}^{-1} a s uvažováním, že $\mathbf{x} \doteq \mathbf{K}^{-1} \mathbf{x}'$. Pak se dostáváme zpět ke kalibrované verzi $\lambda \mathbf{x} = \mathbf{\Pi}_0 g \mathbf{X}$. Proto je znalost kalibrační matice podstatná pro rekonstrukci 3D euklidovského prostoru.

3.4.1 Epipolární geometrie pro nekalibrované kamery, fundamentální matice

Epipolární omezení definované v kapitole 3.3.1 pro kalibrované kamery může být rozšířeno pro nekalibrované kamery jednoznačným způsobem. Toto omezení vyjadřuje vlastnost, že prostor určený třemi vektory $\mathbf{x}_1, \mathbf{x}_2$ a \mathbf{T} je nulový. Jinými slovy tyto tři vektory jsou ve stejné rovině. Tento prostor je tak dán smíšeným součinem tří vektorů; $\mathbf{x}'_2, \mathbf{T}' = \mathbf{K} \mathbf{T}$ a $\mathbf{K} \mathbf{R} \mathbf{x}_1 = \mathbf{K} \mathbf{R} \mathbf{K}^{-1} \mathbf{x}'_1$ a sice

$$\mathbf{x}'_2{}^T \hat{\mathbf{T}}' \mathbf{K} \mathbf{R} \mathbf{K}^{-1} \mathbf{x}'_1 = 0. \quad (3.44)$$

Tato rovnice je nekalibrovanou verzí epipolárního omezení popsaného v kapitole 3.3.1. Jiný způsob jak získat epipolární omezení pro nekalibrované kamery je přímá substituce $\mathbf{x} \doteq \mathbf{K}^{-1}\mathbf{x}'$ do $\mathbf{x}_2^T \hat{\mathbf{T}}\mathbf{R}_1\mathbf{x}_1 = 0$, tedy

$$\mathbf{x}_2^T \underbrace{\mathbf{K}^{-T} \hat{\mathbf{T}} \mathbf{R} \mathbf{K}^{-1}}_{\mathbf{F}} \mathbf{x}_1 = 0. \quad (3.45)$$

Matice \mathbf{F} definována v předešlém se nazývá fundamentální matice.

$$\mathbf{F} = \mathbf{K}^{-T} \hat{\mathbf{T}} \mathbf{R} \mathbf{K}^{-1} = \hat{\mathbf{T}}' \mathbf{K} \mathbf{R} \mathbf{K}^{-1} \in \mathbb{R}^{3 \times 3}. \quad (3.46)$$

Jestliže $\mathbf{K} = \mathbf{I}$, matice \mathbf{F} je identická s maticí \mathbf{E} . Matice \mathbf{F} může být odhadnuta ze souboru osmi a více sesouhlasených bodů na dvou snímcích. Algoritmus použitý v případě kalibrovaných kamer lze využít i zde, nicméně, zatímco esenciální matice poskytuje vše nezbytné pro rekonstrukci polohy kamer a tudíž i rekonstrukci 3D euklidovského prostoru, pro případ nekalibrovaných kamer nelze snadno získat \mathbf{R} a \mathbf{T} z fundamentální matice \mathbf{F} , což vede k využití rozvrstveného přístupu při rekonstrukci

3.4.2 Rozvrstvený přístup při rekonstrukci

Určení úplné kalibrace a euklidovská rekonstrukce 3D scény z nekalibrovaných snímků je komplexní postup, jehož úspěch závisí na řešení obtížných nelineárních rovnic, jakožto i dalších faktorů nad nimiž máme malou kontrolu. Toto přivádí celý problém na rozdělení postupu na jednotlivé kroky, kdy je rekonstrukce nejprve stanovena pro obecnou transformaci a poté postupně upravována, než je možné získat euklidovskou rekonstrukci. Postup bývá obvykle z projektivní přes afinní transformaci až po transformaci euklidovskou.

3.4.2.1 Geometrické rozvrstvení

Vezměme v úvahu vztah (3.42), zapsaný nyní pro dva pohledy

$$\lambda_i \mathbf{x}'_i = \mathbf{K}_i \mathbf{\Pi}_0 \mathbf{g}_{ie} \mathbf{X}_{ie}, \quad i = 1, 2. \quad (3.47)$$

Zde je přímo rozlišena euklidovská struktura \mathbf{X}_e od struktur, které budou definovány později při rozvrstveném přístupu rekonstrukce. \mathbf{g}_{ie} , pro $i = 1, 2$, představuje euklidovskou polohu i -tého systému kamery vztažený k referenční soustavě. Související projektivní matice je poté

$$\mathbf{\Pi}_{ie} = \mathbf{K}_i \mathbf{\Pi}_0 \mathbf{g}_{ie}, \quad i = 1, 2. \quad (3.48)$$

Jelikož je výběr referenční souřadné soustavy libovolný, zvolí se takovým způsobem, aby \mathbf{g}_{ie} měla jednoduchý tvar pro jeden ze snímků. Například, chceme aby $\mathbf{g}_{1e} = (\mathbf{I}, 0)$, pak vztah (3.42) přechází na

$$\lambda_1 \mathbf{x}'_1 = \mathbf{K}_1 \mathbf{\Pi}_0 \mathbf{X}_e. \quad (3.49)$$

Což znamená, že se referenční souřadný systém shoduje s euklidovským souřadným systémem s počátkem v optickém středu první kamery. Přesto jsou zde stále nejednoznačnosti, můžeme zvolit obecnou 4x4 matici \mathbf{H} tak, aby

$$\mathbf{K}_1 \mathbf{\Pi}_0 \mathbf{X}_e = \mathbf{K}_1 \mathbf{\Pi}_0 \mathbf{H}^{-1} \mathbf{H} \mathbf{X}_e = \mathbf{\Pi}_{1p} \mathbf{X}_p, \quad (3.50)$$

kde $\mathbf{\Pi}_{1p} = \mathbf{\Pi}_{1e} \mathbf{H}^{-1}$ a $\mathbf{X}_p = \mathbf{H} \mathbf{X}_e$.

Matice \mathbf{H} představuje libovolnou lineární transformaci homogenních souřadnic, nazývaná projektivní transformace. Z možných matic \mathbf{H} vybereme takovou, která umožní jednoduchý model projekce. Např. $\mathbf{\Pi}_{1p} = \mathbf{\Pi}_0 = [\mathbf{I}, 0]$, což odpovídá nastavení referenčního systému do prvního snímku. Obecně nelze nalézt euklidovskou transformaci g_{1e} aby $\mathbf{\Pi}_{1e}$ odpovídal standardní kameře. Nicméně můžeme nalézt obecnou invertibilní matici \mathbf{H} , např jako

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{K}^{-1} & 0 \\ \mathbf{v}^T & v_4 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{v}^T & v_4 \end{bmatrix} = \mathbf{H}_a^{-1} \mathbf{H}_p^{-1} \in \mathbb{R}^{4 \times 4}, \quad (3.51)$$

kde první matice na pravé straně je vlastně afinní transformace souřadnic, zatímco druhá matice reprezentuje projektivní transformaci.

Pomocí těchto dvou referenčních systémů dostáváme

$$\begin{aligned} \lambda_1 \mathbf{x}'_1 &= \mathbf{\Pi}_0 \mathbf{X}_p, \\ \lambda_2 \mathbf{x}'_2 &= \mathbf{\Pi}_{2p} \mathbf{X}_p = \mathbf{K}_2 \mathbf{\Pi}_0 g_{2e} \mathbf{H}_a^{-1} \mathbf{H}_p^{-1} \mathbf{X}_p, \end{aligned} \quad (3.52)$$

kde

$$\mathbf{X}_p = \mathbf{H}_p \overbrace{\mathbf{H}_a \underbrace{g_e \mathbf{X}}_{\mathbf{X}_e}}^{\mathbf{X}_a}.$$

\mathbf{X} představuje skutečnou 3D strukturu, \mathbf{X}_e se od ní liší euklidovskou transformací $g_e(\mathbf{R}_e, \mathbf{T}_e)$, \mathbf{X}_a se liší obecnější afinní transformací $\mathbf{H}_a g_e$, a konečně \mathbf{X}_p obecnou lineární (projektivní) transformací $\mathbf{H}_p \mathbf{H}_a g_e$. Obdobně definujeme projektivní matice pro euklidovskou kameru jako $\mathbf{\Pi}_{ie} = \mathbf{K}_i \mathbf{\Pi}_0 g_{ie}$, afinní kameru jako $\mathbf{\Pi}_{ia} = \mathbf{K}_i \mathbf{\Pi}_0 g_{ia} \mathbf{H}_a^{-1}$ a projektivní kameru jako $\mathbf{\Pi}_{ip} = \mathbf{K}_i \mathbf{\Pi}_0 g_{ip} \mathbf{H}_a^{-1} \mathbf{H}_p^{-1}$.

Souhrn vztahů je v následující tabulce. Předpokládá se, že se parametry kamery nemění, tedy $\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K}$.

Tab.1 Vztah mezi třemi typy rekonstrukce

	Projekce kamery	3D struktura
Euklidovská	$\mathbf{\Pi}_{1e} = [\mathbf{K}, 0], \mathbf{\Pi}_{2e} = [\mathbf{KR}, \mathbf{KT}]$	$\mathbf{X}_e = g_e \mathbf{X} = \begin{bmatrix} \mathbf{R}_e & \mathbf{T}_e \\ 0 & 1 \end{bmatrix} \mathbf{X}$
Afinní	$\mathbf{\Pi}_{2a} = [\mathbf{KRK}^{-1}, \mathbf{KT}]$	$\mathbf{X}_a = \mathbf{H}_a \mathbf{X}_e = \begin{bmatrix} \mathbf{K} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X}_e$
Projektivní	$\mathbf{\Pi}_{2p} = [\mathbf{KRK}^{-1} + \mathbf{K} \mathbf{T} \mathbf{v}^T, \mathbf{v} \quad \mathbf{K} \mathbf{T}]$	$\mathbf{X}_p = \mathbf{H}_p \mathbf{X}_a = \begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{v}^T \mathbf{v}_4^{-1} & \mathbf{v}_4^{-1} \end{bmatrix} \mathbf{X}_a$

3.4.2.2 Projektivní rekonstrukce

Problém rekonstrukce projektivní struktury \mathbf{X}_p a projekce kamery $\mathbf{\Pi}_{ip}$ lze řešit pomocí několika totožných bodů na snímcích $(\mathbf{x}'_1, \mathbf{x}'_2)$. Princip je jednoduchý, ze

ztotožněných bodů sestavíme fundamentální matici \mathbf{F} , z níž pak získáme projekci kamery $\mathbf{\Pi}_{2p}$ a nakonec projektivní strukturu \mathbf{X}_p triangulací.

Projektivní matice $\mathbf{\Pi}_{1p}, \mathbf{\Pi}_{2p}$ popisují vztah daného páru nekalibrovaných snímků $(\mathbf{x}'_1, \mathbf{x}'_2)$ do neznámé struktury \mathbf{X}_p , která souvisí s euklidovskou strukturou projektivní transformací

$$\begin{aligned}\lambda_1 \mathbf{x}'_1 &= \mathbf{\Pi}_{1p} \mathbf{X}_p = [\mathbf{I}, 0] \mathbf{X}_p \\ \lambda_2 \mathbf{x}'_2 &= \mathbf{\Pi}_{2p} \mathbf{X}_p = \left[(\hat{\mathbf{T}}')^T \mathbf{F}, \mathbf{T}' \right] \mathbf{X}_p.\end{aligned}\quad (3.53)$$

Tato rovnice je lineární pro všechny neznámé. Proto je možné určit \mathbf{X}_p přímo. Eliminací neznámých velikostí λ_1 a λ_2 vynásobením obou stran rovnic $\hat{\mathbf{x}}'_1$ a $\hat{\mathbf{x}}'_2$ dostáváme $\mathbf{x}'_i \mathbf{\Pi}_{ip} \mathbf{X}_p = 0, i = 1, 2$, což nám dá tři skalární rovnice pro jeden snímek. Pouze dvě z nich jsou lineárně nezávislé. Proto dva totožné body $\mathbf{x}_1 = [x_1, y_1, 1]^T$ a $\mathbf{x}_2 = [x_2, y_2, 1]^T$ dávají čtyři lineárně nezávislé omezení pro \mathbf{X}_p

$$\begin{aligned}(x_1 \boldsymbol{\pi}_1^{3T}) \mathbf{X}_p &= \boldsymbol{\pi}_1^{1T} \mathbf{X}_p, & (y_1 \boldsymbol{\pi}_1^{3T}) \mathbf{X}_p &= \boldsymbol{\pi}_1^{2T} \mathbf{X}_p \\ (x_2 \boldsymbol{\pi}_2^{3T}) \mathbf{X}_p &= \boldsymbol{\pi}_2^{1T} \mathbf{X}_p, & (y_2 \boldsymbol{\pi}_2^{3T}) \mathbf{X}_p &= \boldsymbol{\pi}_2^{2T} \mathbf{X}_p,\end{aligned}\quad (3.54)$$

kde $\mathbf{\Pi}_{1p} = [\boldsymbol{\pi}_1^1, \boldsymbol{\pi}_1^2, \boldsymbol{\pi}_1^3]^T$ a $\mathbf{\Pi}_{2p} = [\boldsymbol{\pi}_2^1, \boldsymbol{\pi}_2^2, \boldsymbol{\pi}_2^3]^T$ jsou projektivní matice. Projektivní struktura může být získána metodou nejmenších čtverců ze systému lineárních rovnic

$$\mathbf{M} \mathbf{X}_p = 0. \quad (3.55)$$

Pro převedení projektivní struktury \mathbf{X}_p na euklidovskou \mathbf{X}_e je třeba určit neznámou transformaci $\mathbf{H}^{-1} = \mathbf{H}_a^{-1} \mathbf{H}_p^{-1}$ tak, aby $\mathbf{X}_e = \mathbf{H}_a^{-1} \mathbf{X}_a = \mathbf{H}_a^{-1} \mathbf{H}_p^{-1} \mathbf{X}_p$. To může být provedeno ve dvou krocích. Prvně určit afinní strukturu \mathbf{X}_a nalezením matice \mathbf{H}_p^{-1} a poté určením euklidovské struktury \mathbf{X}_e nalezením matice \mathbf{H}_a^{-1} (nebo analogicky \mathbf{K}).

3.4.2.3 Afinní rekonstrukce

Jak již bylo uvedeno v předešlém, v případě obecného pohybu a při absenci znalosti scény nebo parametrů kamery, vše co je možné určit je fundamentální matice a související projektivní struktura \mathbf{X}_p . K tomu abychom rozšířili projektivní strukturu na afinní, musíme najít transformaci

$$\mathbf{H}_p^{-1} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{v}^T & v_4 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \quad (3.56)$$

tak, že $\mathbf{X}_a = \mathbf{H}_p^{-1} \mathbf{X}_p$ a $\mathbf{\Pi}_{ia} = \mathbf{\Pi}_{ip} \mathbf{H}_p^{-1}, i = 1, 2$. \mathbf{H}_p^{-1} má speciální vlastnost, a to sice že zobrazuje všechny body \mathbf{X}_p , splňující vztah $[\mathbf{v}^T, v_4] \mathbf{X}_p = 0$, do bodů s homogenními souřadnicemi $\mathbf{X}_a = [X, Y, Z, 0]^T$. V afinním prostoru tyto body odpovídají bodům nekonečně daleko od středu zvoleného referenčního systému. Geometrické místo těchto bodů se nazývá nevlastní rovina v projektivním prostoru, označovaná P_∞ . Je charakterizovaná vektorem

$$\boldsymbol{\pi}_\infty^T \doteq [\boldsymbol{v}^T, v_4] = [v_1, v_2, v_3, v_4] \in \mathbb{R}^4, \quad (3.57)$$

jako $\boldsymbol{X}_p \in P_\infty$ jestliže a pouze tehdy když $\boldsymbol{\pi}_\infty^T \boldsymbol{X}_p = 0$. Poznamenejme, že tento $\boldsymbol{\pi}_\infty^T$ je definován pouze na libovolný skalární koeficient. Proto je nejdůležitější poměr mezi \boldsymbol{v} a v_4 . Obvyklá volba pro $\boldsymbol{\pi}_\infty^T$ je jednoduše $[\boldsymbol{v}^T / v_4, 1]$ pokud $v_4 \neq 0$. V každém případě je nalezení této roviny ekvivalentní k nalezení vektoru \boldsymbol{v} a v_4 a proto ekvivalentní k nalezení rozšiřující matice \mathbf{H}_p^{-1} (nebo \mathbf{H}_p) z projektivního do afinního prostoru.

3.4.2.4 Euklidovská rekonstrukce

Pokud se parametry kamery mezi jednotlivými snímky nemění nebo pokud se mění jen několik parametrů, pak pomocí více snímků pořízených z rozdílných poloh můžeme rekonstruovat euklidovskou strukturu a polohu kamer spolu s kalibrací kamery. Existuje metoda, pomocí níž je možné nalézt přímo matici $\mathbf{H} = \mathbf{H}_p \mathbf{H}_a$ bez nutnosti procházení přes všechny kroky geometrického rozvrstvení..

Předpokládejme strukturu projektivní matice jako

$$\boldsymbol{\Pi}_{ie} \boldsymbol{X}_e = \boldsymbol{\Pi}_{ie} \mathbf{H}^{-1} \mathbf{H} \boldsymbol{X}_e \doteq \boldsymbol{\Pi}_{ip} \boldsymbol{X}_p, \quad i = 1, 2, \dots, m. \quad (3.58)$$

Zároveň předpokládáme, že je známa projektivní matice $\boldsymbol{\Pi}_{ip}$ a struktura \boldsymbol{X}_p . Zvolíme první snímek jako referenční, tj. $\boldsymbol{\Pi}_{ie} = [\mathbf{K}_1, 0]$ a $\boldsymbol{\Pi}_{ip} = [\mathbf{I}, 0]$ a transformační matici ve tvaru

$$\mathbf{H} = \begin{bmatrix} \mathbf{K}_1 & 0 \\ -\boldsymbol{v}^T \mathbf{K}_1 & 1 \end{bmatrix}. \quad (3.59)$$

Řešení lze hledat ze vztahu

$$\boldsymbol{\Pi}_{ip} \mathbf{Q} \boldsymbol{\Pi}_{ip}^T \cong \mathbf{S}_i^{-1}, \quad (3.60)$$

kde

$$\mathbf{Q} \doteq \mathbf{H} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{H}^T \in \mathbb{R}^{4 \times 4} \quad (3.61)$$

je symetricky pozitivní semidefinitní matice hodnosti 3. Vztah (3.60) se označuje jako *absolutní kvadratické omezení*.

3.4.3 Kalibrace založená na znalosti scény

3.4.3.1 Dílčí znalost scény

V běžném prostředí se informace, jako je ortogonalita a rovnoběžnost mezi liniemi (např. budovy, ulice), může výhodně využít. Samozřejmě nelze ověřit, jestli je tomu tak, avšak uživatel by si toho měl být vědom a vytvořit algoritmus, který tento předpoklad využívá.

Soubor paralelních linií se protíná v jednom bodě nekonečně vzdáleném. Projekce tohoto bodu do obrazové roviny se nazývá úběžný bod (*vanishing point*). Jestliže dva vektory $l^1, l^2 \in \mathbb{R}^3$ představují obrazy dvou paralelních přímek pak jejich úběžný bod je dán

$$v \approx l^1 \times l^2 = \hat{l}^1 \hat{l}^2. \quad (3.62)$$

Pokud jsou si přímky navzájem kolmé, můžeme předpokládat, při vhodné volbě souřadného systému, že se jejich 3D směr shoduje se třemi hlavními osami $e_1 = [1,0,0]^T$, $e_2 = [0,1,0]^T$, $e_3 = [0,0,1]^T$. Pak úběžné body, odpovídající souboru paralelních přímek, budou

$$v_1 = \mathbf{K} \mathbf{R} e_1, \quad v_2 = \mathbf{K} \mathbf{R} e_2, \quad v_3 = \mathbf{K} \mathbf{R} e_3. \quad (3.63)$$

Otrogonální vztah mezi e_1, e_2, e_3 poskytuje snadno omezení pro kalibrační matici \mathbf{K} . Přesněji

$$v_i^T \mathbf{S} v_j = v_i^T \mathbf{K}^{-T} \mathbf{K}^{-1} v_j = e_i^T \mathbf{R}^T \mathbf{R} e_j = e_i^T e_j = 0, \quad i \neq j. \quad (3.64)$$

Kde $\mathbf{S} = \mathbf{K}^{-T} \mathbf{K}^{-1}$ je opět symetrická matice spojená s nekalibrovanou kamerou

$$\mathbf{S} = \mathbf{K}^{-T} \mathbf{K}^{-1} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (3.65)$$

Když jsou tři úběžné body nalezeny, poskytují tři nezávislé omezení pro matici \mathbf{S} .

$$v_1^T \mathbf{S} v_2 = 0, \quad v_1^T \mathbf{S} v_3 = 0, \quad v_2^T \mathbf{S} v_3 = 0. \quad (3.66)$$

Obecně má symetrická matice \mathbf{S} pět stupňů volnosti. Bez dalšího omezení nemůžeme matici \mathbf{S} obnovit úplně. Pokud předpokládáme nulovou šikmost pixelů ($f_{s_0} = 0$) a známý poměr stran snímku ($f_{s_x} = f_{s_y}$), můžeme pak dostat jediné řešení pro matici \mathbf{K} z jednoho snímku.

Detailní popis využívající tento typ omezení může být nalezen v [36].

3.4.3.2 Kalibrace pomocí kalibračního objektu

Kalibrace pomocí kalibračního objektu je metoda, kdy máme přístup ke kameře a můžeme umístit známý objekt na scénu. Při těchto podmínkách můžeme použít objekt s přesným počtem bodů (*calibration rig*), jejichž souřadnice jsou známy vzhledem k obecnému referenčnímu snímku s velkou přesností.

Nechť $\mathbf{X} = [X, Y, Z, 1]^T$ jsou souřadnice bodu p na objektu. Jeho obraz má snímkovou souřadnici $\mathbf{x}' = [x', y', 1]^T$, která vyhovuje rovnici (3.42). Necht' $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_s \in \mathbb{R}^4$ jsou tři řádkové vektory projektivní matice $\boldsymbol{\Pi} = \mathbf{K} \boldsymbol{\Pi}_0 \mathbf{g} \in \mathbb{R}^{3 \times 4}$, pak může být rovnice (3.42) zapsána pro každý bod p^i na objektu jako

$$\lambda^i = \begin{bmatrix} x''^i \\ y''^i \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\pi}_1^T \\ \boldsymbol{\pi}_2^T \\ \boldsymbol{\pi}_3^T \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{bmatrix}. \quad (3.67)$$

Ze třetího řádku dostáváme $\lambda^i \boldsymbol{\pi}_3^T \mathbf{X}^i = 0$. Proto pro každý bod dostáváme následující dvě rovnice

$$\begin{aligned} x''^i \boldsymbol{\pi}_3^T \mathbf{X}^i &= \boldsymbol{\pi}_1^T \mathbf{X}^i, \\ y''^i \boldsymbol{\pi}_3^T \mathbf{X}^i &= \boldsymbol{\pi}_2^T \mathbf{X}^i. \end{aligned} \quad (3.68)$$

Na rozdíl od předchozích formulací, jsou zde X^i, Y^i a Z^i známé a proto jsou známé i x''^i a y''^i . Můžeme tedy seskupit všechny neznámé části projekční matice $\boldsymbol{\Pi}$ do vektoru a přepsat výše uvedené rovnice jako systém lineárních rovnic

$$\mathbf{M}\boldsymbol{\Pi}^8 = 0, \quad (3.69)$$

kde $\boldsymbol{\Pi}^8$ je přeskupená projekční matice $\boldsymbol{\Pi}$

$$\boldsymbol{\Pi}^8 = [\pi_{11}, \pi_{12}, \pi_{13}, \pi_{21}, \pi_{22}, \pi_{23}, \pi_{31}, \pi_{32}, \pi_{33}, \pi_{41}, \pi_{42}, \pi_{43}]^T \in \mathbb{R}^{12} \quad (3.70)$$

a řádky matice \mathbf{M} jsou funkcemi (x''^i, y''^i) a (X^i, Y^i, Z^i) . Lineární odhad $\boldsymbol{\Pi}^8$ může být získán minimalizací kritéria nejmenších čtverců, a sice $\min \|\mathbf{M}\boldsymbol{\Pi}^8\|^2$ s podmínkou $\|\boldsymbol{\Pi}^8\|^2 = 1$. Toto může být provedeno, jako obvykle pomocí SVD. Po získání odhadu projekční matice

$$\boldsymbol{\Pi} = \mathbf{K}[\mathbf{R} | \mathbf{T}] = [\mathbf{KR} | \mathbf{KT}] \quad (3.71)$$

můžeme rozložit první 3x3 submatici do kalibrační matice \mathbf{K} a matici rotace \mathbf{R} použitím QR dekompozice

$$\mathbf{KR} = \begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} \\ \pi_{21} & \pi_{22} & \pi_{23} \\ \pi_{31} & \pi_{32} & \pi_{33} \end{bmatrix}, \quad (3.72)$$

která poskytne odhad vnitřních parametrů v kalibrační matici \mathbf{K} a tak i rotační složky vnějších parametrů. Odhad posunutí ukončí celý kalibrační proces

$$\mathbf{T} = \mathbf{K}^{-1} \begin{bmatrix} \pi_{14} \\ \pi_{24} \\ \pi_{34} \end{bmatrix}. \quad (3.73)$$

Tento proces vyžaduje, aby měly body v prostoru známé souřadnice X^i a také aby byly v obecné pozici, to znamená že neleží například na rovině. Nicméně nejjednodušší a nejpoužívanější kalibrační objekty jsou právě rovinné šachovnicové obrazce.

3.4.3.3 Kalibrace pomocí plošného vzoru

Jelikož si můžeme zvolit referenční souřadný systém libovolně, zvolíme jej zarovnaný s plošným obrazcem, takže body na něm budou mít souřadnice $\mathbf{X} = [X, Y, 0, 1]^T$. Pak s ohledem na souřadný systém kamery obraz \mathbf{x}' bodu \mathbf{X} na obrazci je dán vztahem

$$\lambda^i = \begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{r}_1, \mathbf{r}_2, \mathbf{T}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad (3.74)$$

kde $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^3$ jsou první a druhý sloupec matice rotace \mathbf{R} .

Všimněme si, že matice

$$\mathbf{H} \doteq \mathbf{K}[\mathbf{r}_1, \mathbf{r}_2, \mathbf{T}] \in \mathbb{R}^{3 \times 3} \quad (3.75)$$

je lineární transformací homogenních souřadnic $[X, Y, 1]^T$ do homogenních souřadnic $[x, y, 1]^T$, jedná se o homografii mezi plošným obrazcem a rovinou snímku.

Vynásobením obou stran rovnice pomocí $\hat{\mathbf{x}}'$ pro eliminaci λ dostáváme

$$\hat{\mathbf{x}}' \mathbf{H} [X, Y, 1]^T = 0. \quad (3.76)$$

V této rovnici známe \mathbf{x}' i $[X, Y, 1]^T$. Pokud máme dostatečný počet bodů na obrazci, může být z této rovnice matice \mathbf{H} řešena lineárně.

Pokud už známe \mathbf{H} , a víme, že první dva sloupce $[\mathbf{h}_1, \mathbf{h}_2]$ odpovídají $\mathbf{K}[\mathbf{r}_1, \mathbf{r}_2]$, jelikož $\mathbf{r}_1, \mathbf{r}_2$ jsou kolmé vektory dostáváme dvě rovnice, pro něž kalibrační matice musí splňovat

$$\mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0, \quad \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2. \quad (3.77)$$

Tyto rovnice jsou kvadratické vzhledem k prvkům matice \mathbf{K} . Nicméně, pokud jsme schopni zanedbat strukturu $\mathbf{S} = \mathbf{K}^{-T} \mathbf{K}^{-1} \in \mathbb{R}^{3 \times 3}$, můžeme ji pak určit lineárně použitím SVD.

3.5 Způsoby ztotožňování bodů na snímcích

Ztotožňování bodů na snímcích je další stěžejní operací při rekonstrukci prostorové scény. Máme-li k dispozici sesouhlasený pár, je výpočet prostorové souřadnice bodu snadný. Nicméně právě nalezení korespondence na snímcích je poměrně obtížná úloha. Obvykle se však předpokládá, že je geometrie kamer známá a tak lze aplikovat epipolární omezení. Toto omezení zaručuje, že příslušný bod na

jednom snímku leží na odpovídající epipolární přímce na druhém snímku a naopak. Použitím epipolárního omezení mohou být snímky upraveny (rektifikovány), tzn. transformovány tak, aby se epipolární přímka shodovala s odpovídajícím řádkem snímku. Tato transformace redukuje korespondenční proces z 2D úlohy na 1D úlohu, kdy je prozkoumáván pouze odpovídající řádek. Jakmile je korespondence nalezena, lze rekonstruovat 3D model scény. Z souřadnice, nebo-li též hloubka, je nepřímo úměrná disparitě – rozdílu snímkových souřadnic odpovídajících si bodů na dvou snímcích.

K tomu, aby se určila korespondence mezi dvěma snímky, je nezbytné definovat základní prvky, elementy, které se mají sesouhlasovat. V postupech sesouhlasení mohou být rozlišeny dvě hlavní třídy základních prvků: tvary (rysy) a jas pixelu. Tvary jsou typicky snadno rozlišitelné elementy či regiony ve snímcích a vyhledávání je prováděno pouze pro tyto tvary. Metody používající tyto základní rysy náleží do skupiny tzv. *feature-based stereo* metod. Na druhé straně metody zpracovávající hodnoty jasu pixelu provádějí vyhledávání přes pixely. Tyto metody náleží do skupiny označované jako *area-based stereo* metody.

Feature-based stereo

Marr definoval tvary na pozicích ve snímku, kde se jas ostře mění (hrany, rohy), protože prohlásil, že tyto rozdíly v jasu jsou zachovány v obou snímcích, zatímco jas pixelu se může velmi snadno měnit v důsledku rozličných geometrických zkreslení, vzorkování obrazu, atd. Tvary jsou získávány různými detektory [7],[44],[74],[22].

Vyhledávání je prováděno pouze mezi získanými tvary. Proto je disparita vyhodnocena pouze v pozicích těchto tvarů, tudíž velmi řídké. Proto je pro sesouhlasení ze dvojice snímků nevýhodná. Pro získání mapy disparity je nezbytné provést určitý postprocessing, jako např. interpolaci přes plynulou plochu. Nicméně tento proces může přinést spoustu chyb. Dalším problémem je detekce tvarů samotná. Ta může být také snadno zdrojem chyb a proto celý proces sesouhlasení nemusí být správný.

Ačkoliv výzkumy opustily testování *feature-based* metod pro sesouhlasení ve dvojici snímků, zájem byl opětovně obnoven v odlišné oblasti vývoje, např. pro *wide-baseline stereo*. Problém v této oblasti spočívá v nalezení geometrie kamer z daného souboru snímků. Pro tuto aplikaci malé množství hodnot disparity nevádí. Definice tvarů se odlišuje od předešlých a jsou definovány jako význačné regiony – nejvíce se odlišující části snímku vzhledem k okolí.

Area-based stereo

Základní elementy pro sesouhlasení v *area-based* metodách jsou jednotlivé pixely, přes které se vyhledávání provádí. Metody se soustředí na hledání korespondence mezi pixely s výsledkem ve formě mapy disparity. Vyhledávání vychází pouze z funkcí jasu pixelu, a proto není potřeba žádné určení tvarů a rysů.

Proces vyhodnocení vzájemného sesouhlasení je vyčíslen na základě podobnostních statistik. Nejjednodušší statistikou je porovnávání jasu pixelu [10],[11], které je ovšem velice citlivé na šum a zkreslení snímku. Robustnější přístupy používají porovnávací okno předdefinované velikosti, které určuje okolí pixelu zahrnutého do vyhodnocení. Nejvíce používanými statistikami jsou např. SSD (*Sum of Squared Differences*) [11],[51],[71], SAD (*Sum of Absolute Differences*) [19],[27], NCC (*normalized cross-correlation*) [30].

Area-based metody mají nevýhodu oproti předešlé metodě v tom, že porovnávají funkci jasu pixelu, která může kolísat vlivem zkreslení.

Jak už bylo uvedeno výše, podstata digitální stereovize spočívá v nalezení korespondence mezi základními elementy. Kvalita výsledku přímo souvisí s použitým algoritmem pro vyhledávání a sesouhlasení. Tyto algoritmy lze rozdělit do již výše definovaných globálních a lokálních metod. Globální metody optimalizují určitou globální funkci, zatímco lokální metody využívají pouze část snímku kolem daného pixelu.

3.5.1 Globální metody

Mnoho globálních metod je formulováno jako systém minimalizující globální funkci (energii). Podstatou je nalezení funkce disparity d , která minimalizuje globální energii

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d). \quad (3.78)$$

Výraz $E_{data}(d)$ představuje, do jaké míry se funkce disparity shoduje se vstupní dvojicí snímků.

$$E_{data}(d) = \sum_{(x,y)} C(x, y, d(x, y)), \quad (3.79)$$

kde (x, y) reprezentuje souřadnice pixelu

$C(x, y, d(x, y))$ hodnotící funkci pro vyhledávání.

Výraz $E_{smooth}(d)$ zahrnuje předpoklad spojitosti daný použitým algoritmem. Aby bylo možné optimalizaci výpočetně provést, je často omezen pouze na měření rozdílu mezi sousedními disparitami pixelů,

$$E_{smooth}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x+1, y)) + \rho(d(x, y) - d(x, y+1)), \quad (3.80)$$

kde ρ je určitá monotónně vzrůstající funkce rozdílu disparit.

Jakmile je globální energie definována, je možné využít množství algoritmů hledající lokální minimum. Některé metody hledají minimum odděleně pro každý řádek a tak provádějí minimalizaci 1D funkce [3],[9]. Jiné metody minimalizují přímo 2D funkci, představující prostor disparit [31],[35],[59].

Dynamické programování

Zatímco 2D optimalizace rovnice (3.78) může být výpočetně náročné pro obecné třídy spojitých funkcí, dynamické programování může nalézt globální minimum pro samostatné přímky vyhledáváním v akceptovatelném časovém intervalu. Problém optimalizace je formulován jako výpočet minimální dráhy přes matici všech sesouhlasených bodů mezi dvěma korespondujícími vyhledávacími přímkami. Překrytí jsou modelována přiřazením skupiny pixelů v jednom snímku do jednoho pixelu na snímku druhém. Problémy dynamického programování spočívají ve zvolení správné míry překrytí, obtížností s uplatněním konsistence mezi vzájemnými prohledávacími přímkami [49]. Dalším problémem je, že přístup v dynamickém programování vyžaduje uplatnění monotónnosti, nebo-li omezení v uspořádání. Toto omezení vyžaduje, aby vzájemné uspořádání pixelů na skenovací přímce zůstalo stejné mezi dvěma snímky.

3.5.2 Lokální metody

Lokální metody neoptimalizují žádnou globální funkci, zde se vypočítává mapa disparit na základě části obrazu, okolí bodu. Výsledná disparita je pro každý pixel určena pouze okolím tohoto pixelu, přes které se aplikuje některá z metod.

Předpokládáme-li area-based stereo, je každý pixel označen signaturou. Tato signatura popisuje pixel a snahou je definovat takové signatury, které umožní pixely jednoduše rozpoznávat jedny od druhých, a to právě na základě těchto signatur. Nicméně pro obecné snímky je toto nemožné, z důvodu různých zkreslení a proto je cílem definovat signatury co nejvíce rozpoznatelné.

Typická definice signatur je určité okolí (okno předdefinovaného rozměru) vyhodnocovaného pixelu na snímku. Proto je správné zvolení tohoto okna velmi důležité, neboť přímo ovlivňuje rozlišitelnost signatury.

Nad signaturami jsou definovány vyhledávací statistiky vyhodnocující míru podobnosti. Typickými představiteli jsou *Sum of Squared Differences* (SSD), *Sum of Absolute Differences* (SAD), *normalized cross-correlation* (NCC). Algoritmy sesouhlasení jsou aplikovány z důvodu nalezení korespondencí mezi snímky. V následující tabulce je představeno několik statistik používaných v korelačních metodách.

Tab. 2: Definice nejběžnějších statistik

Statistika	Definice
SAD	$C(x, y, d) = \sum_{i,j} I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j) $
ZSAD	$C(x, y, d) = \sum_{i,j} \left (I_1(x+i, y+j) - \overline{I_1(x, y)}) - (I_2(x+d_x+i, y+d_y+j) - \overline{I_2(x+d_x, y+d_y)}) \right $
SSD	$C(x, y, d) = \sum_{i,j} (I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j))^2$
ZSSD	$C(x, y, d) = \sum_{i,j} \left[(I_1(x+i, y+j) - \overline{I_1(x, y)}) - (I_2(x+d_x+i, y+d_y+j) - \overline{I_2(x+d_x, y+d_y)}) \right]^2$
SSD-N	$C(x, y, d) = \frac{\sum_{i,j} (I_1(x+i, y+j) - I_2(x+d_x+i, y+d_y+j))^2}{\sqrt{\sum_{i,j} I_1(x+i, y+j)^2 \cdot \sum_{i,j} I_2(x+d_x+i, y+d_y+j)^2}}$
CC	$C(x, y, d) = \sum_{i,j} (I_1(x+i, y+j) - \overline{I_1(x, y)}) \cdot (I_2(x+d_x+i, y+d_y+j) - \overline{I_2(x+d_x, y+d_y)})$
NCC	$C(x, y, d) = \frac{\sum_{i,j} (I_1(x+i, y+j) - \overline{I_1(x, y)}) \cdot (I_2(x+d_x+i, y+d_y+j) - \overline{I_2(x+d_x, y+d_y)})}{\sqrt{\sum_{i,j} (I_1(x+i, y+j) - \overline{I_1(x, y)})^2 \cdot \sum_{i,j} (I_2(x+d_x+i, y+d_y+j) - \overline{I_2(x+d_x, y+d_y)})^2}}$

kde $I_k(x, y)$, $\overline{I_k(x, y)}$ je intenzita, resp. střední intenzita bodu (x, y) ve snímkových souřadnicích pro k -tý snímek,
 i, j jsou indexy definující velikost okna
 d_x, d_y hodnoty disparit ve směru os x a y .

Klasické lokální algoritmy sesouhlasení jsou takzvané winner-takes-all (WTA) [50],[72], které vybírají jako sesouhlasený pár ten s nejlepším výsledkem statistické funkce při předpokladu jedinečnosti. Tento algoritmus získá nejlepší výsledek sesouhlasení, ovšem nemusí zaručit kvalitu sesouhlasení. Jiný přístup dávající přesnější algoritmus sesouhlasení, Confidently-Stable-Matching (CSM), byl představen v [73]. tento přístup je založen na vlastnosti stability řešení, která zaručuje nejlepší jednoznačné řešení.

Volba správné velikosti a tvaru okna je stěžejní pro celkový úspěch lokálních metod. Okno musí být dostatečně veliké, aby se postihly změny intenzity pro spolehlivé vyhledávání a zároveň i dostatečně malé z důvodu zamezení efektů projektivního zkreslení. Kromě toho vhodně zvolené okno může zlepšit rozlišitelnost signatury. Vlivem geometrického zkreslení, při obdélníkovém oknu pevné velikosti mající sledovaný pixel uprostřed okna mohou nastat problémy. Ty vznikají na rozhraních objektů, kde se hloubka scény mění. Okno pak pokrývá nekorespondující části snímku a tudíž jsou signatury těžko porovnatelné. Během doby byly publikovány nejružnější přístupy zvládající tento problém.

Posunující se okno. Tyto metody nemodifikují ani velikost ani tvar okna. Zde je snahou posunovat pozici okna přes zkoumaný pixel z důvodu pokrytí větší totožné části na snímku. Zvolená signatura označující pixel reprezentuje okno na pozici s nejlepší mírou sesouhlasení. Takto by signatury, získané tímto procesem, měly být více spolehlivé. Odpovídající přístupy byly publikovány v[4],[18],[52]. I když první dva řeší problém vyhledávání pomocí dynamického programování, kladou důraz na zlepšení matching costs. V [4] se využívá posunutí okna ve dvou směrech, zatímco [18] přináší metodu, kde se prozkoumává devět pozic okna. Poslední přístup náleží čistě do skupiny lokálních metod. Autoři se zde předkládají řešení s posunujícím se oknem. Okno velikosti 5x5 se posouvá do devíti pozic a zvolena je pozice s nejmenší hodnotou SSD. Posunující se okna umožňují detekovat hranice objektu přesněji, ovšem mají tendenci zničit spojitost povrchu disparit. Proto uvedli metodu kde je posunující okno aplikováno pouze na oblasti možných překročení hranic, které jsou jasně detekované. Ve zbytku snímku je již využito standardního okna.

Okno s přispůsobující se velikostí. Přístupy využívající adaptivní okno se snaží zvládnout problém správné velikosti okna. V částech snímku bez textur může být okno veliké, nicméně v nespojitosti hloubky by okno mělo být malé. Proto následující řešení typicky zvětšují či zmenšují velikost okna v závislosti na prozkoumávané části snímku. Kanade & Okutomi [32] navrhli metodu, kde se vybere vhodné okno na základě vyhodnocení změn intenzity a disparity. Předpoklad je, že v místě diskontuit je kolísání intenzity i disparity veliké, odlišné od pozic na povrchu objektu ve scéně. Kanade & Okutomi navrhli statistický model distribuce disparity uvnitř okna, umožňující sledování vlivu kolísání intenzit a disparit na nejistotu odhadu disparity v centru okna. Počínaje prvotním odhadem disparity, algoritmus opakovaně obnovuje disparitu pro

každý pixel zvolením tvaru a velikostí okna. Okno je zvětšováno krokově ve všech směrech. Opakování je zastaveno když odhadu disparity v centru okna konverguje.

Okno využívající segmentaci obrazu. Pro zajištění, že okna pokrývají relevantní části snímku, je snahou zajistit informaci o pozici objektu pomocí segmentace. Snímek je rozdělen na základě funkce intenzity. Tato metoda zajišťuje, že okno nepřekročí rozhraní segmentu. I zde však mohou nastat problémy. A to se správností segmentace (problém se stíny). Tao a spol [75] uvedli postup využívající segmentaci obrazu. Na počátku je disparita, mající nejlepší skóre vyhledávání přes všechny ztotožněné páry, zvolena jako počáteční disparita pro každý pixel. Poté je referenční snímek segmentován. Pro každý segment je přiřazena disparita odpovídající průměru počátečních disparit uvnitř segmentu. Druhý snímek je předpovídán z referenčního snímku použitím informace o hloubce. Vyhledávání bodů je prováděno na základě globální vyhledávací funkce, která definuje podobnost mezi predikovaným a druhým snímek. Cílem je nalézt mapu hloubek maximalizující globální vyhledávací funkci. Segmentace umožňuje lepší reprezentaci informace o hloubce, nicméně může přinést různé chyby.

Okno založené na spojených komponentech konstantních disparit. Boykov a spol. [6] publikovali postupy, kdy jsou okna definována tak, aby pokrývala scénu se stejnou disparitou, odtud tedy komponenty konstantních disparit. V [6] se počítá pro každý pixel soubor možných disparit. Komponenty disparit jsou vytvořeny spojením pixelů majících stejnou možnou disparitu. Jako výsledná disparita pro jednotlivý pixel je zvolena ta naležící komponente s největší disparitou. Navržený algoritmus má tendenci vytvořit po částech konstantní mapu disparit, z čehož vyplývají možné problémy s různými šikmými objekty.

Okna modelované v prostoru disparit. V současných postupech se ukazuje, že vlivem geometrických zkreslení způsobených rozdílnou pozicí kamer, je nemožné vytvořit okno na vstupních snímcích a očekávat, že budou pokrývat stejnou část scény. Proto se výzkum zaměřil na návrh metody, kde by mohla být reálná scéna modelována s výsledkem ve formě prostoru obrazu0 disparit, tedy 3D struktury složené ze všech možných sesouhlasených párů. Protože prostor disparit obsahuje všechny možné páry, je snadné jej modelovat. Szeliski & Scharstein [71] navrhli metodu pro sesouhlasení bodů v prostoru disparit. Definovali obraz prostoru disparit (DSI), což je 3D struktura bodů (x,y,d) , kde (x,y) představuje pixel v prvním snímku a d všechny možné disparity.

3.6 Analýza přesnosti v určení prostorové souřadnice bodu

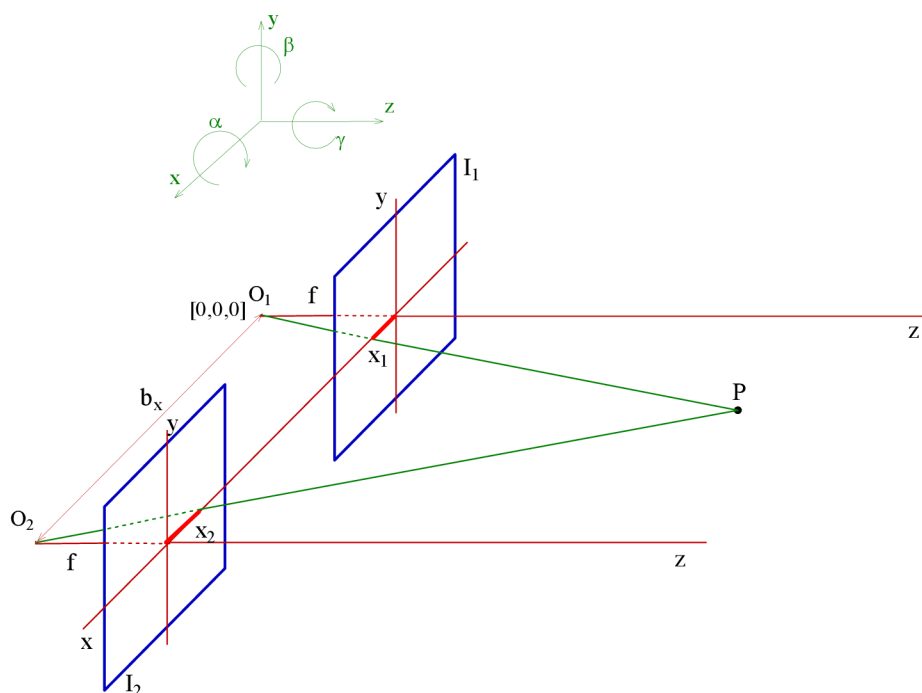
3.6.1 Vliv kalibrace kamer

Kalibrace kamer je velice důležitým krokem při určování 3D souřadnice ze získaných stereosnímků. Pro danou dvojici snímků, jakmile je vyřešen problém korespondence bodů, přesnost v určení 3D souřadnice závisí na celé geometrii kamer, t.j. přesnosti v určení prvků vnitřní a vnější orientace. V následujícím textu bude sledována předpokládaná chyba v určení Z souřadnice bodu v závislosti na nedokonalém určení polohy kamer vůči sobě. Přesné určení vzájemné polohy ovlivňuje epipolární předpoklad používaný v procesu korespondence bodů, jakožto i ustanovení oblasti korespondence na počátku.

Z praktického hlediska je teda poloha kamer důležitá pro 3D měření založeném na stereosnímčích. I když je kalibrace dokonalá, parametry kamer se mohou měnit během užívání. To naskytá několik otázek. Jakým způsobem ovlivní nedokonalá kalibrace určení 3D souřadnice? Jak velkou chybu jednotlivých parametrů při kalibraci je možné akceptovat pro zadanou přesnost měření s použitím reálného měřicího systému.

V této kapitole je tak analyzován efekt různých chyb v kalibraci na celkovou přesnost měřicího systému. Tato analýza je důležitá neboť jakákoliv sebepřesnější metoda kalibrace kamer nebo sesouhlasení bodů přináší určitou větší či menší chybu. A proto je důležité pochopit, která s jednotlivých chyb má největší vliv na přesnost určení 3D souřadnice. Při této analýze je využíváno poznatků uvedených v [86].

Předpokládejme základní stereoskopický systém popsany [65] a zobrazený na Obr.11.



Obr.11. Princip stereofotogrammetrie

Při určování chyb kalibrace bude nejprve brán v potaz vždy jen jeden zdroj chyby, ostatní parametry považujeme za správné. Následně bude vliv jednotlivých chyb sloučen. Předpokládejme, že je korespondenční problém vyřešen a je nalezená správná dvojice bodů na snímcích. Zdroj chyb v určení Z souřadnice je možné rozdělit do několika kategorií.

3.6.1.1 Chyba v určení Z souřadnice vlivem pootočení kamer v osách snímku

Předpokládejme že je první kamera perfektně zkalibrována a její optická osa představuje osu z souřadného systému s počátkem v ohnisku (Obr.11). Optická osa kamery 2 je paralelní k ose objektivu první kamery. Je taktéž perfektně zkalibrována až na určitý úhel otočení γ kolem optické osy (obr.12). Pomocí trigonometrie dostaneme tyto vztahy

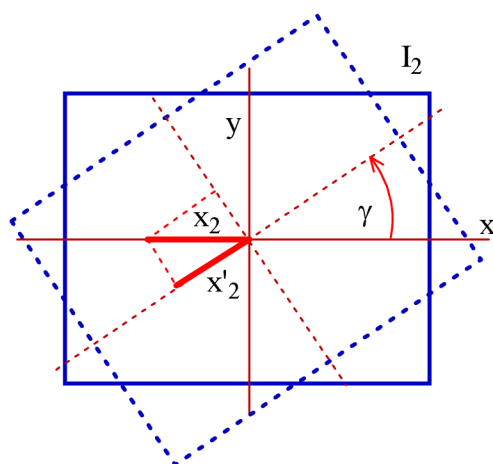
$$Z_{\text{teor}} = f \left(\frac{b_x}{x_1 - x_2} \right), \quad (3.81)$$

$$Z_{\text{skut}} = f \left(\frac{b_x}{x_1 - x_2'} \right),$$

kde Z_{skut} je sledovaná Z -tová souřadnice bodu P ,
 Z_{teor} pak správná Z -tová souřadnice.
 b_x je báze kamer.

Na základě toho, že je první kamera kalibrována, můžeme pro absolutní chybu v určení Z souřadnice psát

$$\begin{aligned} \Delta Z &= Z_{\text{teor}} - Z_{\text{skut}} = \\ &= f \left(\frac{b_x}{x_1 - x_2} \right) - f \left(\frac{b_x}{x_1 - x_2'} \right) \\ &= f b_x \frac{x_2 - x_2'}{(x_1 - x_2)(x_1 - x_2')} \\ &= Z_{\text{skut}} \left(\frac{x_2 - x_2'}{x_1 - x_2} \right). \end{aligned} \quad (3.82)$$

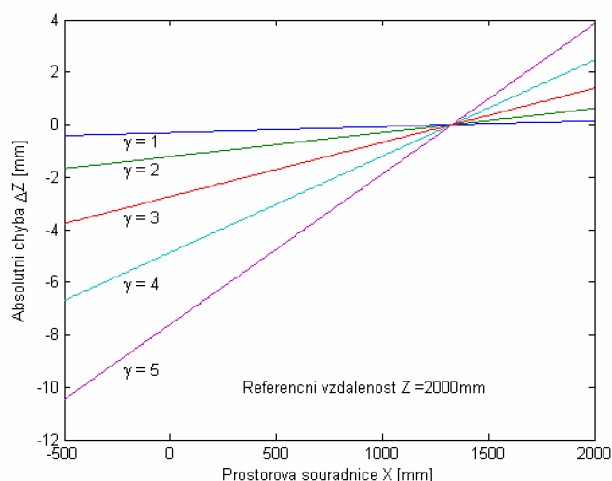


Obr.12. Natočení druhé kamery o chybový úhel γ

Dosazením za $x_2' = x_2 \cos \gamma$, kde γ je úhel natočení kamery vůči sobě, a za předpokladu, že $x_1 = f \cdot Y / X$ a $x_2 = f \cdot (X - b_x) / Z$ dostáváme

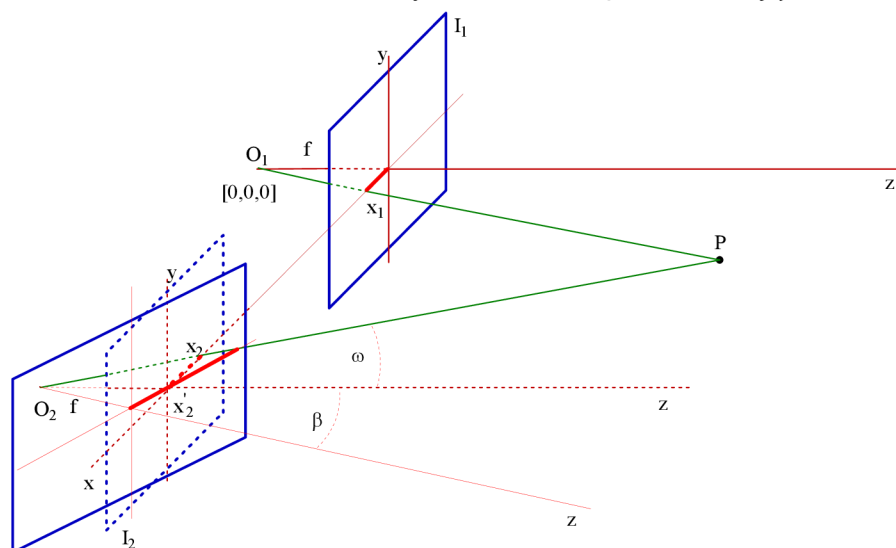
$$\Delta Z = Z_{\text{teor}} \frac{(X - b_x)(1 - \cos \gamma)}{b_x}. \quad (3.83)$$

Obrázek 13 ilustruje závislost chyby Z souřadnice na úhlu otočení γ a souřadnici X .



Obr.13. Chyba souřadnice Z v závislosti na úhlu natočení γ a souřadnici X

Předpokládejme, že je první kamera perfektně zkalibrována a její optická osa představuje osu z souřadného systému s počátkem v ohnisku (Obr.14). Druhá kamera je taktéž perfektně zkalibrována až na určitý úhel otočení β kolem osy y.



Obr.14. Natočení kamery o úhel β

Podobně jako v předchozím určíme souřadnici x_2' . Z Obr. 14 vidíme, že $\tan(\beta + \omega) = x_2' / f$ a $\tan \omega = x_2 / f$. Uvážíme-li trigonometrický vztah $\tan(\beta + \omega) = (\tan \beta + \tan \omega) / (1 - \tan \beta \tan \omega)$ dostáváme

$$\frac{x_2'}{f} = \frac{\tan \beta + x_2 / f}{1 - \tan \beta \cdot x_2 / f} \quad (3.84)$$

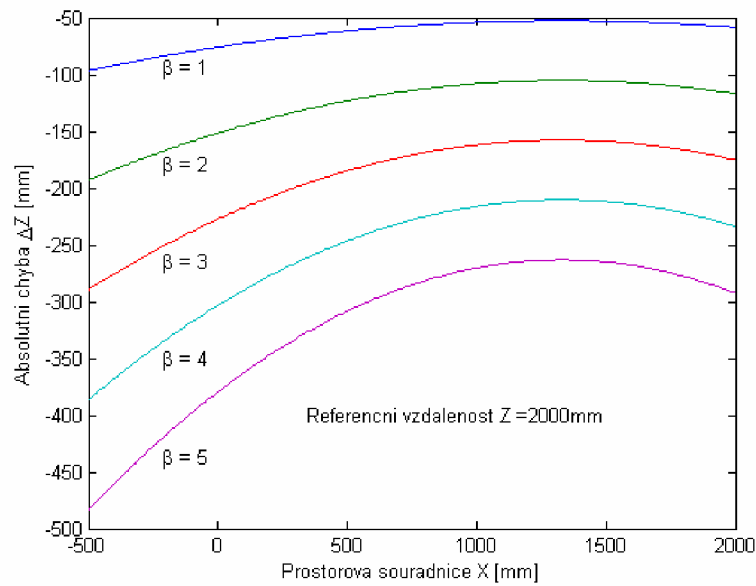
Pak může být x_2' vyjádřeno jako

$$x_2' = f \frac{x_2 + f \tan \beta}{f - x_2 \cdot \tan \beta} \quad (3.85)$$

kde β je chybový úhel natočení kamer kolem osy y .
dosazením vztahu (3.85) do (3.82) pro určení chyby ΔZ dostáváme

$$\begin{aligned} \Delta Z &= Z_{skut} \left(\frac{x_2 - f \frac{x_2 - f \tan \beta'}{x_2 \tan \beta + f_2}}{x_1 - x_2} \right) = \\ &= Z_{skut} \frac{\tan \beta \cdot f^2 \left[1 + \left(\frac{x_2}{f} \right)^2 \right]}{(x_1 - x_2)(x_2 \tan \beta + f)} \\ &\cong \frac{\tan \beta \left[Z_{skut}^2 + (X - b_x)^2 \right]}{b_x} \end{aligned} \quad (3.86)$$

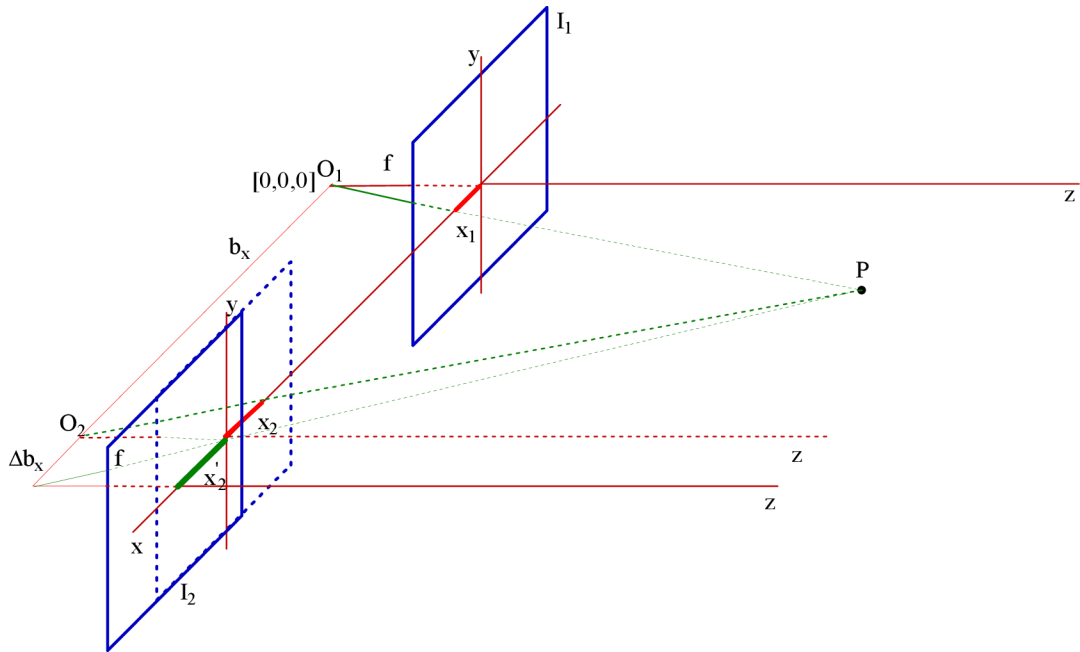
Obrázek 15 ilustruje závislost chyby Z souřadnice na úhlu otočení β a souřadnici X .



Obr.15. Chyba souřadnice Z v závislosti na úhlu natočení β a souřadnici X

3.6.1.2 Chyba v určení Z souřadnice vlivem posunutí kamer v osách snímku

Předpokládejme opět situaci jako v předchozím případě, jen uvažujme určité posunutí Δb_x druhé kamery v ose x . Situaci je znázorněna na Obr. 16.

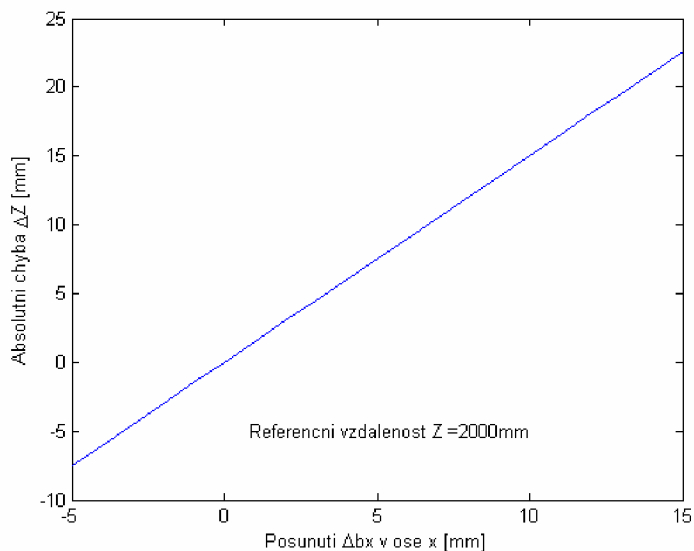


Obr.16. Situace při posunutí druhé kamery o Δb_x v ose x

Z Obr. 16 vidíme, že pro souřadnici x_2' platí $x_2' = f(X_2 + \Delta b_x) / Z$. Pokud doplníme tento vztah do výrazu pro určení chyby ΔZ dostáváme

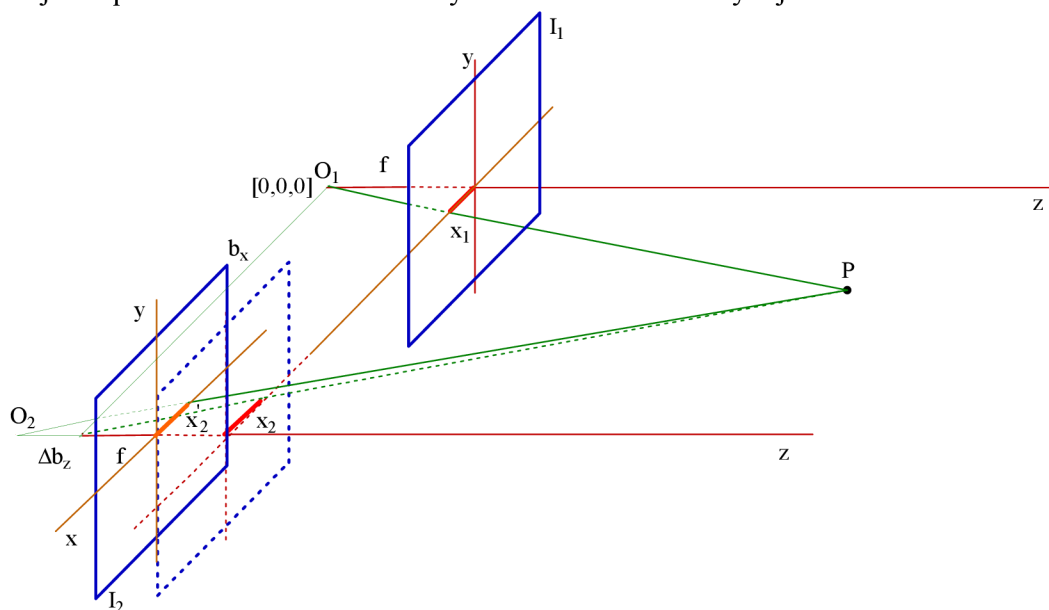
$$\begin{aligned} \Delta Z &= Z_{\text{skut}} \left(\frac{x_2 - (X_2 + \Delta b_x) / Z}{x_1 - x_2} \right) = \\ &= Z_{\text{skut}} \frac{f \frac{X_2}{Z} - f \frac{X_2}{Z} - f \frac{\Delta b_x}{Z}}{f \frac{X_1}{Z} - f \frac{X_1 - b_x}{Z}} \\ &= Z_{\text{skut}} \frac{\Delta b_x}{b_x}. \end{aligned} \quad (3.87)$$

Obrázek 17 ilustruje závislost chyby Z souřadnice na posunutí Δb_x kamer vůči sobě.



Obr.17. Chyba souřadnice Z v závislosti na posunutí kamer vůči sobě o Δb_x .

Nyní předpokládejme podobnou situaci jako v předešlém s tím rozdílem, že uvažujeme posunutí Δb_z druhé kamery v ose z. Situaci zachycuje Obr. 18.

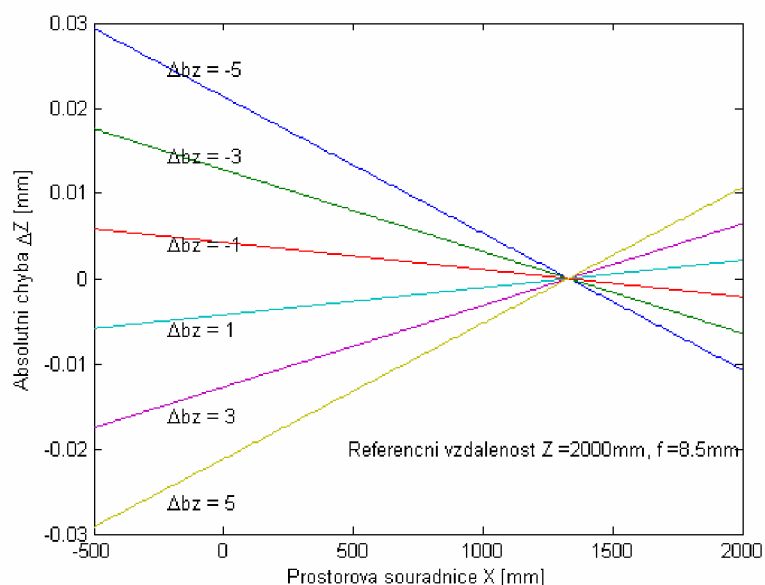


Obr.18. Situace při posunutí druhé kamery o Δb_z v ose z

Pro souřadnici x'_2 platí $x'_2 = fX_2 / (Z + \Delta b_z)$. Nyní po dosazení do základního vztahu dostáváme pro určení chyby ΔZ

$$\begin{aligned} \Delta Z &= Z_{\text{skut}} \frac{f \frac{X_2}{Z} - f \frac{X_2}{Z + \Delta b_z}}{b_x} = \\ &= \frac{fX_2 \Delta b_z}{b_x (Z + \Delta b_z)}. \end{aligned} \quad (3.88)$$

Obrázek 19 ilustruje závislost chyby Z souřadnice na posunutí Δb_z kamer vůči sobě a souřadnici X



Obr.19. Chyba souřadnice Z v závislosti na posunutí Δb_z kamer vůči sobě.

3.6.1.3 Vliv jednotlivých parametrů na chybu měření

Na základě dílčích výsledků uvedených v předchozích odstavcích je snadné říci, který z parametrů vyžaduje větší pozornost při kalibraci kamer. Nejkritičtější je chyba v určení natočení kamer kolem osy y , dále posunutí v ose x a dále otočení kolem optické osy z .

Rovněž je možné spočítat toleranci jednotlivých parametrů pro danou přesnost měření celého systému. Tak například pro danou dvojici snímků s parametry $b_x=1500\text{mm}$, $f=8\text{mm}$, jestliže chceme, aby chyba Z -tové souřadnice byla menší než 40mm , tak pro bod s $Z=2000\text{mm}$ a $X=1000\text{mm}$ může být tato přesnost dosažena, jestliže $\gamma \leq 16,2^\circ$ přičemž ostatní parametry jsou určeny přesně. Podobně pokud uvažujeme pouze chybový úhel β , tak pro danou přesnost musí být $\beta \leq 0,8^\circ$. Pro posunutí v ose x pak Δb_x musí být menší jak 30mm .

V případě, že nebudeme uvažovat pouze jeden zdroj chyby, což je v praxi pravděpodobnější, můžeme zkombinovat vliv otočení kolem os a a posunutí v osách dohromady. Tak například pokud jsou kamery kalibrovány s chybovými parametry Δb_x , β a γ můžeme sloučit následující vztahy

$$x_2' = x_2 Z / (Z + \Delta b_z),$$

$$x_2' = f \frac{x_2 + f \tan \beta}{f - y \cdot 2 \cdot \tan \beta}, \quad (3.89)$$

$$x_2' = x_2 \cos \gamma,$$

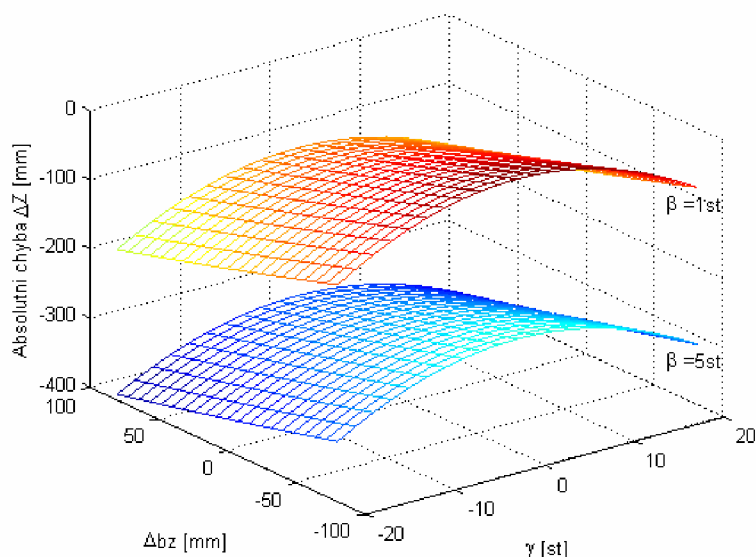
do výsledného vztahu pro x_2'

$$x_2' = \frac{fZ}{Z + \Delta b_z} \frac{x_2 \cos \gamma + f \tan \beta}{f - x_2 \cos \gamma \cdot \tan \beta} \quad (3.90)$$

Zde je důležité pořadí, v jakém jsou vztahy kombinovány do sebe. Vezmeme-li v úvahu rozdílné pořadí dostáváme odlišný výsledek. Pro absolutní chybu Z souřadnice, můžeme psát

$$\Delta Z = Z \left(\frac{x_2 - \frac{fZ}{Z + \Delta b_z} \frac{x_2 \cos \gamma + f \tan \beta}{f - x_2 \cos \gamma \cdot \tan \beta}}{x_1 - x_2} \right) \quad (3.91)$$

Obr.20 ilustruje závislost absolutní chyby Z souřadnice v případě aplikace více chybových parametrů, zde posunutí v ose z , otočení kolem osy y a z . Při výpočtu bylo použito: $b_x = 1500\text{mm}$, $f=8\text{mm}$, referenční bod s $X=1000\text{mm}$ a $Z = 2000\text{mm}$.



Obr.20. Chyba souřadnice Z v závislosti na Δb_x , β a γ .

Stejně jako v předchozím by bylo vhodné vypočítat toleranci parametrů při uvažování více zdrojů chyb. Pro danou toleranci Z souřadnice bychom rádi znali s jakou maximální možnou chybou mohou být kalibrační parametry určeny. Řešení není tak snadné, neboť máme jednu rovnici o více neznámých. Přibližně však platí, že celková chyba je dána součtem chyb pro jednotlivé případy, které byly popsány výše. Můžeme tedy psát pro předchozí případ

$$\Delta Z(\Delta b_z, \beta, \gamma) \cong \Delta Z(\Delta b_z) + \Delta Z(\beta) + \Delta Z(\gamma) \quad (3.92)$$

Takto je problém zjednodušen na řešení více rovnic vždy o jedné neznámé.

3.6.2 Vliv diskretizace obrazu

Z důvodu diskretizace obrazu se snímkové souřadnice mění skokově s kvantizační chybou $\pm 1/2$ pixelu. Pro souřadnice v ose x to znamená, že jsou získány s chybou $\pm \delta/2$, kde δ představuje vzorkovací interval snímku. Proto disparita, definovaná jako rozdíl snímkových souřadnic, může nabývat minimální chyby $\pm \delta$. Definujeme-li chybu disparity jako $\Delta d = d' - d$, kde d' je disparita vzniklá diskretizací, pak může být tato chyba v rozsahu $-\delta$ až δ .

Uvažujeme-li stereoskopický systém znázorněný na Obr.11, můžeme napsat podobný vztah pro určení absolutní chyby Z souřadnice jako v kapitole 3.6.1,

$$\begin{aligned}
 \Delta Z &= Z_{\text{disk}} - Z_{\text{skut}} = \\
 &= \frac{b_x f}{d'} - Z_{\text{skut}} \\
 &= \frac{b_x f}{d + \Delta d} - Z_{\text{skut}} \\
 &= \frac{b_x f}{b_x f / Z_{\text{skut}} + \Delta d} - Z_{\text{skut}} \\
 &= \frac{-Z_{\text{skut}} \Delta d}{b_x f + Z_{\text{skut}} \Delta d}.
 \end{aligned} \tag{3.93}$$

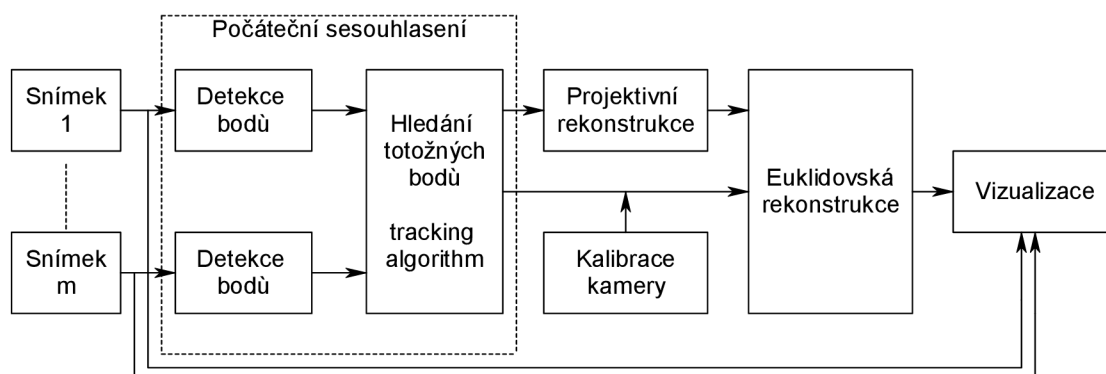
Ze vztahu je patrné, že chyba z -tové souřadnice vzrůstá, se zmenšujícím se členem $b_x f$ a tedy pro případ přibližujících se kamer. Vezmeme-li v úvahu konkrétní hodnoty, např. $b_x = 1000 \text{ mm}$, $f = 8 \text{ mm}$, $Z = 4000 \text{ mm}$, pak změna x -ové souřadnice o jeden pixel (tedy chyba disparity $\Delta d = 0,0033 \text{ mm}$ pro 2Mpixelový snímač velikosti $1/2,7''$) způsobí chybu Z souřadnice $\Delta Z = 6,2 \text{ mm}$. Z tohoto lze rovněž usuzovat, že pokud je chyba, způsobená nepřesným určením parametrů kamery, menší než $3,1 \text{ mm}$, pak již další zpřesnění parametrů kamery neovlivní výslednou chybu Z souřadnice. Přesnost snímacího systému je tak limitována rozlišením snímače.

3.7 Testovací program pro rekonstrukci prostorových souřadnic snímaných objektů

Vytvořený testovací program Recon3D slouží pro ověření výše uvedených postupů pro rekonstrukci prostorových souřadnic ze dvou či tří snímků, tedy kalibraci kamery a nalezení korespondujících bodů jakožto i konečnou rekonstrukci prostorové souřadnice bodu. Program používá digitální fotoaparát k pořízení stereosnímků. Komunikace probíhá přes USB rozhraní a snímky jsou ukládány do PC v nekomprimovaném formátu BMP s 8bitovou hloubkou na pixel (barevný nebo v odstínu šedé). K vytvoření uživatelsky příjemného prostředí bylo opět využito vývojového prostředí Builder C++. Při vývoji bylo rovněž využito OpenCV[®] knihovny pro urychlení řešení některých úloh rekonstrukce prostorové souřadnice. Tato knihovna nabízí programátorům celou řadu algoritmů, které mohou být využity pro velké množství úloh počítačového vidění. Rovněž obsahuje ucelenou knihovnu pro maticový počet, který je zde hojně využíván.

3.7.1 Vývojový diagram programu Recon3D

Testovací program v této verzi vyhodnocuje 3D strukturu snímané scény na základě informace získané ze sesouhlasených bodů ve dvojici či trojici snímků. Obecný přehled systému je zobrazen na Obr. 21.



Obr. 21 Postup při rekonstrukci 3D prostorové scény

a) Pořízení sekvence snímků

Vytvořený program Recon3D nabízí dvě možnosti získání snímku. První možností je otevření dříve uložených snímků v PC. Podporovanými formáty jsou 8bitové BMP (barevné nebo v odstínu šedé) formáty. Jestliže program komunikuje přímo s digitálním fotoaparátem či kamerou, data jsou získávána pomocí API rozhraní a funkcí implementovaných v knihovnách DLL nebo LIB. Tyto knihovny jsou součástí ovladačů nebo vývojového prostředí dodávaného k jednotlivým zařízením. Pomocí těchto API funkcí program komunikuje s fotoaparátem nebo kamerou a takto je možné uložit snímanou scénu do souboru či paměti PC.

b) detekce bodů na snímcích

Automatické nalezení základních geometrických prvků (rohů, atd.) je využito v procesu zjištění projektivní geometrie. Body jsou nalezeny pro jednotlivé snímky a v následném kroku přes dané snímky sesouhlaseny.

Kvalita bodu se souřadnicí $\mathbf{x} = [x, y]^T$, uvažovaného jako kandidát pro výběr, bývá obvykle měřena pomocí Harrisova kritéria [22]

$$C(\mathbf{x}) = \det(\mathbf{G}) + k \times \text{trace}^2(\mathbf{G}) \quad (3.94)$$

počítaného přes určitou oblast $W(\mathbf{x})$ na snímku. Ta může být například obdélníkové okno centrované v bodě \mathbf{x} o velikosti 9x9 pixelů. Ve vztahu je konstanta k volena uživatelem, často jako empiricky určená hodnota 0,03. \mathbf{G} je matice 2x2, která závisí na souřadnici \mathbf{x} následovně

$$\mathbf{G} = \begin{bmatrix} \sum_{W(\mathbf{x})} I_x^2 & \sum_{W(\mathbf{x})} I_x I_y \\ \sum_{W(\mathbf{x})} I_x I_y & \sum_{W(\mathbf{x})} I_y^2 \end{bmatrix} \mathbf{R}^{2 \times 2}, \quad (3.95)$$

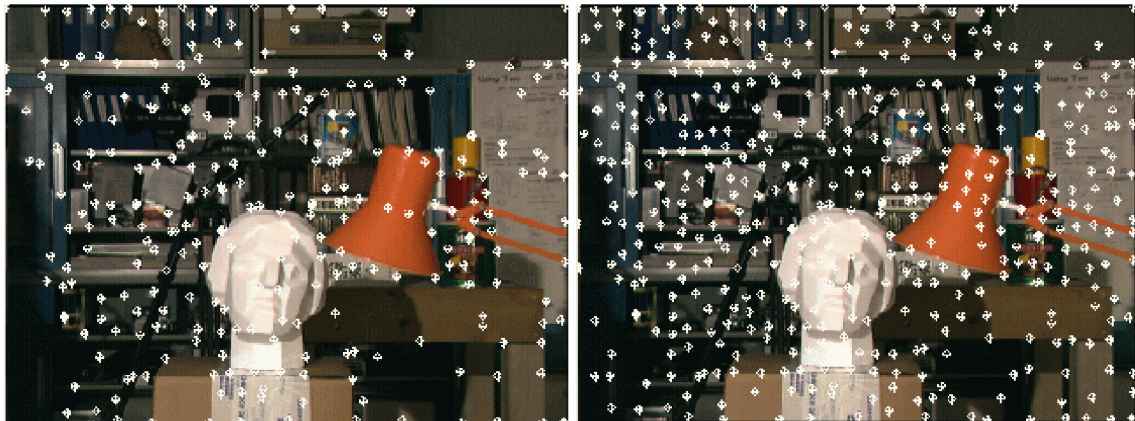
kde I_x, I_y jsou gradienty získané konvolucí snímku I s derivacemi dvojic Gaussovských filtrů.

Bod x je na snímku zvolen, jestliže kritérium $C(x)$ překročí určitou mez τ . Protože určitá část snímku může obsahovat větší množství objektů, které vyvolají selekci bodu a na druhé straně oblasti, které jsou více homogenní, jsou bez nalezených bodů, je doporučeno rozdělit snímek na větší počet podoblastí a body nalézat v těchto podoblastech. Tímto způsobem dojde k nalezení významných bodů rovnoměrně rozložených přes celý snímek.

Algoritmus detekce bodů vypadá následovně

1. Vypočteme gradient snímku $\nabla I = [I_x, I_y]^T$.
2. Vybereme velikost okna $W(x)$, např. 9×9 . Vypočteme kvalitu $C(x)$ pro každý bod x pomocí vztahu (3.94).
3. Zvolíme mez τ a přerovnáme podle kvality $C(x)$ body x překračující danou mez
4. Zvolíme dělení snímku na více oblastí. Pro každou oblast zvolíme minimální oddělovací prostor a maximální počet bodů vybraných v jednotlivých podoblastech. Zvolíme body s nejvyšší hodnotou $C(x)$. Procházíme seznam bodů, jestliže bod nespadá do oddělovacího prostoru předchozího zvoleného bodu, vybereme tento nový bod. Jinak jej nezahrneme do výběru.
5. Ukončíme hledání, jestliže počet bodů překročí maximum, nebo všechny nalezené body, překročující mez, byly zanedbány.

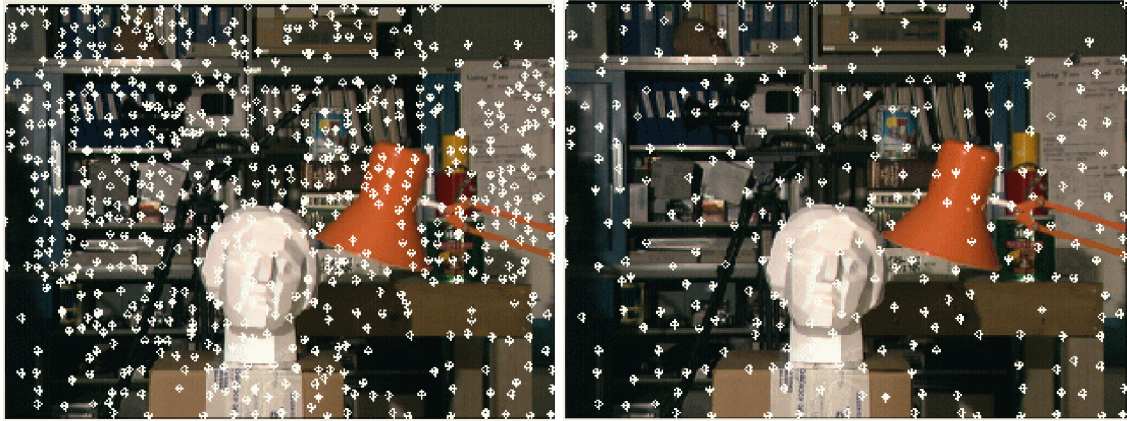
Tento algoritmus je implementován ve funkci *goodFeatureToTrack* z knihovny OpenCV[®] a je takto použit v programu. Na následujících obrázcích (Obr.22a-d) jsou zobrazeny detekované body pro různé zvolené parametry použité v algoritmu detekce. Testovací snímky (Tsukuba) jsou převzaty z [62] (www.middlebury.edu/stereo).



a) $q=0,01, D=10, n=323$

b) $q=0,001, D=10, n=541$

Obr.22 a),b) Detekované body na snímcích pro různé parametry, q představuje hodnotu kvality, D minimální oddělovací prostor v pixelech a n počet nalezených bodů



c) $q=0,001$, $D=7$, $n=909$

d) $q=0,001$, $D=15$, $n=277$

Obr.22c),d) Detekované body na snímcích pro různé parametry, q představuje hodnotu kvality, D minimální oddělovací prostor v pixelech a n počet nalezených bodů

c) nalezení korespondence mezi snímky

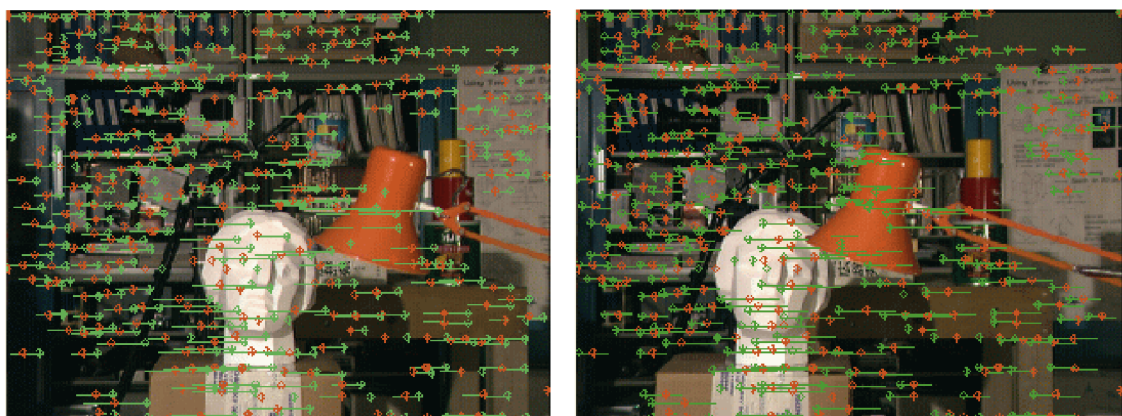
Jakmile jsou v předchozím kroku význačné body nalezeny, je snahou určit korespondenci mezi body přes jednotlivé snímky. Tyto páry (trojice) jsou poté využity pro nalezení epipolární geometrie a následně fundamentální matice.

Body mohou být sesouhlaseny na základě vyhledávací statistiky měřící podobnost zvolených bodů na snímcích. Toto je sice výpočetně náročné, nicméně pro tento účel dostačující, protože vyhodnocení se provádí pouze přes omezený počet bodů. Pro zvolené body se souřadnicemi x_1 na prvním snímku a x_2 na druhém snímku je určeno okolí $W(x_1)$, $W(x_2)$. Toto okolí je obvykle porovnáváno pomocí NCC.

Algoritmus sesouhlasení vypadá následovně:

1. Zvolíme body pomocí postupu popsaného v předešlém odstavci
2. Pro každý bod v prvním snímku najdeme bod na druhém snímku, který maximalizuje funkci NCC. Jestliže žádný bod na druhém snímku nemá skóre překračující zvolenou mez τ , daný bod není zahrnut do souboru korespondencí.

Obrázky Obr.23a-b ukazují body úspěšně nalezené ve dvojici snímků. Zelené body představují ztotožněný bod. Přímký posun daného bodu přes další snímky. Červené body představují ztotožněné body na dalších snímcích.



a)

b)

Obr.23a),b) Sesouhlasené body na snímcích

d) Kalibrace kamery

Jak bylo uvedeno v předchozích kapitolách, je důležité získat parametry vnitřní a vnější orientace pro správné určení prostorové souřadnice. Jestliže program komunikuje přímo s digitálním fotoaparátem nebo kamerou, je možné získat vnitřní parametry přímo například s využitím kalibračního obrazce, pokud je ten taktéž k dispozici. Poté je nutné určit pouze matici rotace a vektor posunutí. Tato transformace definuje pozici kamer vůči sobě. Jestliže není přímý přístup ke snímacímu zařízení a snímky jsou otevřeny z PC bez znalosti kalibrace je možné provést rekonstrukci prostorové souřadnice pomocí rozvrstveného přístupu popsaneho v kapitole 3.4.2

e) Projektivní rekonstrukce

Jakmile jsou korespondence mezi snímky nalezeny, lze určit 3D strukturu scény až k projektivní rekonstrukci. Jestliže nejsou dány či známy další informace o kameře je projektivní rekonstrukce maximum co je možné nalézt.

Z nalezených bodů ve dvojici snímků je možné určit fundamentální matici F použitím osmi bodového algoritmu popsaneho v kapitole 3.4. Určení projektivních matic a projektivní struktury tedy vychází z fundamentální matice.

Algoritmus projektivní rekonstrukce ze dvou snímků je následující:

1. Odhadneme kalibrační matici K tak, že zvolíme střed promítání ve středu snímku, pixely uvažujeme čtvercové. Odhadneme velikost ohniskové vzdálenosti f .
2. Ztotožněné body normalizujeme pomocí matice K a vypočteme fundamentální matici F
3. Určíme odhad matice rotace a vektor posunutí z matice F pomocí SVD.
4. Vypočítáme 3D souřadnice X_i bodů x_i pomocí triangulace a hloubku λ_i jakožto třetí prvek souřadnice X_i .

Pokud je k dispozici více snímků, lze pro zpřesnění odhadu projektivní matice a struktury použít například iterativní algoritmus minimalizující chybu zpětné projekce. Cílem je nalezení polohy kamer pro všechny snímky a 3D struktury pro jakýkoliv bod, který se objeví alespoň na dvojici snímků.

Algoritmus pro vícesnímkovou projektivní rekonstrukci je následující:

1. Nechť je $\alpha_0^j = 1/\lambda_1^j$ hodnota získaná z dvousnímkové rekonstrukce.
2. Pro různé k určíme $\alpha_k^j = \alpha_k^j / \alpha_k^1$ pro $j=1,2,\dots,n$, n je počet ztotožněných bodů.
3. Sestavíme matici P_i použitím α^j a ztotožněných bodů na snímcích tak, že

$$P_i = \begin{bmatrix} \mathbf{x}_1^{1T} \otimes \hat{\mathbf{x}}_i^1 & \alpha^1 \hat{\mathbf{x}}_i^1 \\ \mathbf{x}_1^{2T} \otimes \hat{\mathbf{x}}_i^2 & \alpha^2 \hat{\mathbf{x}}_i^2 \\ \vdots & \vdots \\ \mathbf{x}_1^{nT} \otimes \hat{\mathbf{x}}_i^n & \alpha^n \hat{\mathbf{x}}_i^n \end{bmatrix},$$

kde \otimes je Kroneckerův product,

\mathbf{x}_i^j je souřadnice j-tého bodu na i-tém snímku, $j=1,2,\dots,n$ a $i=2,3,\dots,m$,

$\hat{\mathbf{x}}_i^j$ je antisymetrické matice vytvořená ze souřadnice bodu \mathbf{x}_i^j .

4. Určíme singulární vektor \mathbf{v}_{12} , jehož prvních devět prvků přerovnáme do matice $\tilde{\mathbf{R}}_i$ a poslední tři prvky do vektoru $\tilde{\mathbf{T}}_i$. Pak $\mathbf{\Pi}_{ik+1} = [\tilde{\mathbf{R}}_i, \tilde{\mathbf{T}}_i]$.

5. Máme-li všechny transformační matice, vypočítáme opět hodnoty α_{k+1}^j pomocí

$$\alpha_{k+1}^j = -\frac{\sum_{i=2}^m (\hat{\mathbf{x}}_i^j \mathbf{T}_i)^T \hat{\mathbf{x}}_i^j \mathbf{R}_i \mathbf{x}_i^j}{\sum_{i=2}^m \|\hat{\mathbf{x}}_i^j \mathbf{T}_i\|^2}$$

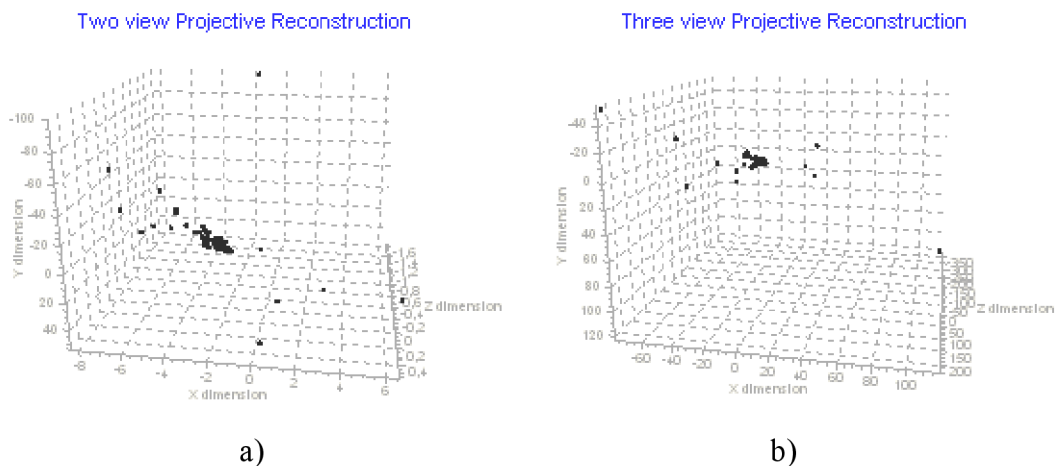
a 3D souřadnici jako $\mathbf{X}_{k+1}^j = \lambda_{ik+1}^j \mathbf{x}_1^j$.

6. Vypočítáme chybu reprojekce jako

$$\varepsilon_r = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|\mathbf{x}_i^j - \mathbf{\Pi}_{ik+1} \mathbf{X}_{k+1}^j\|^2.$$

jestliže $\varepsilon_r > \varepsilon$ pro $\varepsilon > 0$, pak $k = k + 1$ a opakujeme proces od bodu 2, jinak algoritmus ukončíme.

Obrázky 24a-b ukazují projektivní rekonstrukci stejné scény z předešlých odstavců pro dvousnímkovou a třísnímkovou rekonstrukci.



Obr. 24 a) dvousnímková projektivní rekonstrukce, b) třísnímková projektivní rekonstrukce

f) Euklidovská rekonstrukce

Projektivní struktura X_p získaná v předešlém kroku souvisí s euklidovskou strukturou X_e lineární transformací \mathbf{H} jak bylo ukázáno v kapitole 3.4.2.1

$$\mathbf{\Pi}_{ip} \cong \mathbf{\Pi}_{ie} \mathbf{H}^{-1}, \quad X_p \cong \mathbf{H} X_e, \quad i = 1, 2, \dots, m,$$

kde \mathbf{H} má tvar

$$\mathbf{H} = \begin{bmatrix} \mathbf{K}_1 & 0 \\ -\mathbf{v}^T \mathbf{K}_1 & 1_i \end{bmatrix}. \quad (3.96)$$

Řešení hledáme na základě absolutního kvadratického omezení

$$\mathbf{\Pi}_{ip} \mathbf{Q} \mathbf{\Pi}_{ip}^T \cong \mathbf{S}_i^{-1}. \quad (3.97)$$

Budeme-li uvažovat, že střed promítání leží se středu snímku, a že pixely jsou čtvercové, pak předchozí vztah získává jednodušší tvar

$$\mathbf{\Pi}_{ip} \begin{bmatrix} a_1 & 0 & 0 & a_2 \\ 0 & a_1 & 0 & a_3 \\ 0 & 0 & 1 & a_4 \\ a_2 & a_3 & a_4 & a_5 \end{bmatrix} \mathbf{\Pi}_{ip}^T \cong \begin{bmatrix} f_i^2 & 0 & 0 \\ 0 & f_i^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.98)$$

a z toho vyplývající omezení pro matici \mathbf{Q}

$$\begin{aligned} \boldsymbol{\pi}_i^{1T} \mathbf{Q} \boldsymbol{\pi}_i^1 &= \boldsymbol{\pi}_i^{2T} \mathbf{Q} \boldsymbol{\pi}_i^2, \\ \boldsymbol{\pi}_i^{1T} \mathbf{Q} \boldsymbol{\pi}_i^2 &= 0, \\ \boldsymbol{\pi}_i^{1T} \mathbf{Q} \boldsymbol{\pi}_i^3 &= 0, \\ \boldsymbol{\pi}_i^{2T} \mathbf{Q} \boldsymbol{\pi}_i^3 &= 0, \end{aligned} \quad (3.99)$$

kde $\boldsymbol{\pi}_i^1, \boldsymbol{\pi}_i^2, \boldsymbol{\pi}_i^3$ jsou řádky projektivní matice $\mathbf{\Pi}_{ip}$.

Algoritmus euklidovské rekonstrukce poté vypadá takto:

1. Pro m projektivních matic $\mathbf{\Pi}_{ip}$, získaných pomocí předešlého algoritmu, sestavíme omezení pro matici \mathbf{Q} . Mějme vektor $\mathbf{Q}^8 \doteq [a_1, a_2, a_3, a_4, a_5]^T$ sestavený z matice \mathbf{Q} .
2. Sestavíme matici \mathbf{D} seřazením m bloků o velikost 4×5 , kde každý řádek bloku odpovídá jednomu omezení ze vztahu (3.99). Jeden takový blok vypadá následovně

$$\begin{bmatrix} \pi_{11}^2 + \pi_{12}^2 - \pi_{21}^2 - \pi_{22}^2 & 2\pi_{14}\pi_{11} - 2\pi_{21}\pi_{24} & 2\pi_{14}\pi_{12} - 2\pi_{22}\pi_{24} & 2\pi_{14}\pi_{13} - 2\pi_{23}\pi_{24} & \pi_{14}^2 + \pi_{24}^2 \\ \pi_{11}\pi_{21} + \pi_{12}\pi_{22} & \pi_{14}\pi_{21} + \pi_{11}\pi_{24} & \pi_{14}\pi_{22} + \pi_{12}\pi_{24} & \pi_{14}\pi_{23} + \pi_{13}\pi_{24} & \pi_{14}\pi_{24} \\ \pi_{11}\pi_{31} + \pi_{12}\pi_{32} & \pi_{14}\pi_{31} + \pi_{11}\pi_{34} & \pi_{14}\pi_{32} + \pi_{12}\pi_{34} & \pi_{14}\pi_{33} + \pi_{13}\pi_{34} & \pi_{14}\pi_{34} \\ \pi_{21}\pi_{31} + \pi_{22}\pi_{32} & \pi_{24}\pi_{31} + \pi_{21}\pi_{34} & \pi_{24}\pi_{32} + \pi_{22}\pi_{34} & \pi_{24}\pi_{33} + \pi_{23}\pi_{34} & \pi_{24}\pi_{34} \end{bmatrix},$$

kde π_{ij} jsou prvky projektivní matice $\mathbf{\Pi}_p$.

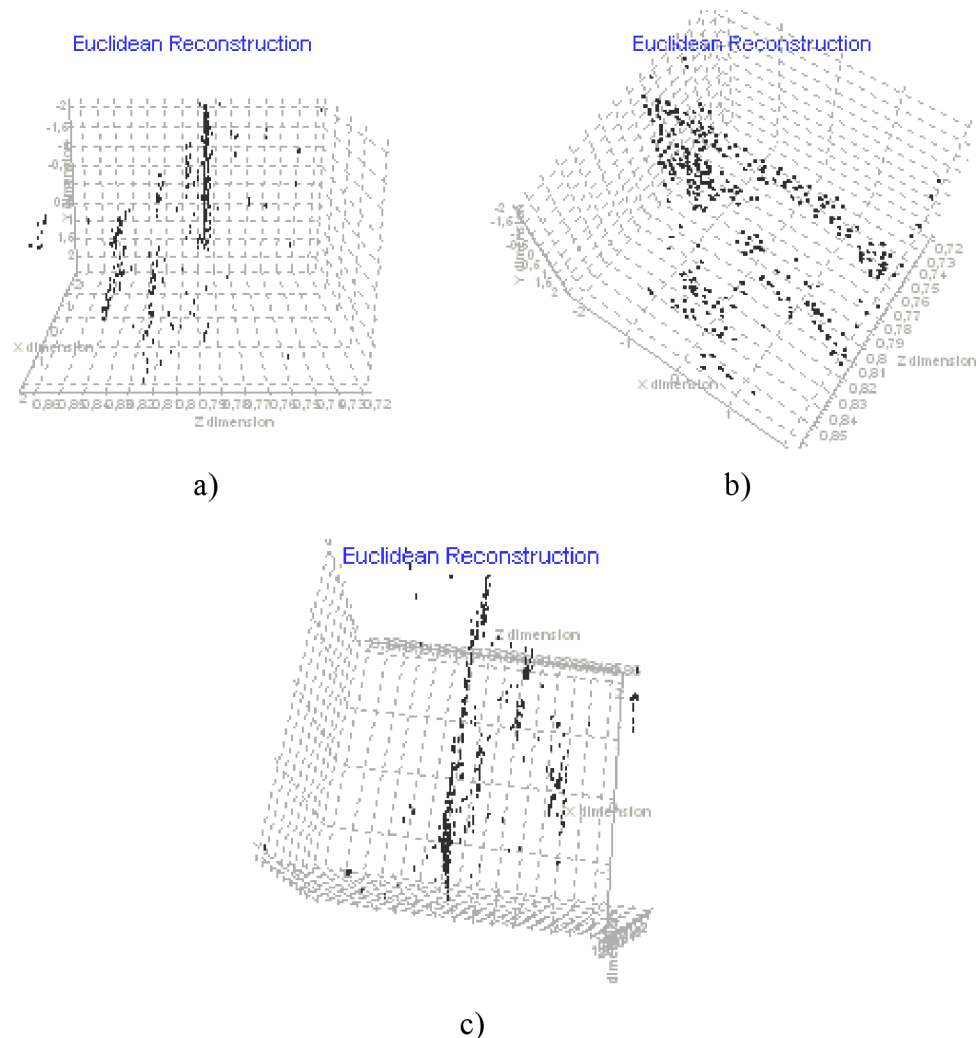
Rovněž sestavíme vektor \mathbf{b} seřazením m bloků o velikosti 4 řádků, a to sice

$$\begin{bmatrix} -\pi_{13}^2 + \pi_{23}^2 & -\pi_{13}\pi_{23} & -\pi_{13}\pi_{33} & -\pi_{23}\pi_{33} \end{bmatrix}.$$

3. Vypočítáme Q^8 jako $Q^8 \doteq \mathbf{D}\mathbf{b}$ a přestavíme prvky zpět do matice \mathbf{Q} .
4. Aplikujeme SVD na \mathbf{Q} a výsledné \mathbf{Q} je pak $\mathbf{Q} = \mathbf{U}_Q \mathit{diag}\{\sigma_1, \sigma_e, \sigma_s, 0\} \mathbf{V}_Q$.
5. Ohniskové vzdálenosti f_i jsou pak získány substitucí do vztahu (3.98).
6. Vypočítej euklidovskou strukturu pomocí transformační matice \mathbf{H} , přičemž \mathbf{K}_1 a \mathbf{v} určíme z prvků matice \mathbf{Q} jako

$$\mathbf{K}_1 = \begin{bmatrix} \sqrt{a_1} & 0 & 0 \\ 0 & \sqrt{a_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ a } \mathbf{v} = -[a_2/a_1, a_3/a_1, a_4]^T.$$

Obrázky 25a-c zobrazují euklidovskou rekonstrukci stejné scény z předešlých odstavců pro třísnímkovou rekonstrukci pro různé úhly pohledu.



Obr 25 a)-c). Euklidovská rekonstrukce z různých pohledů

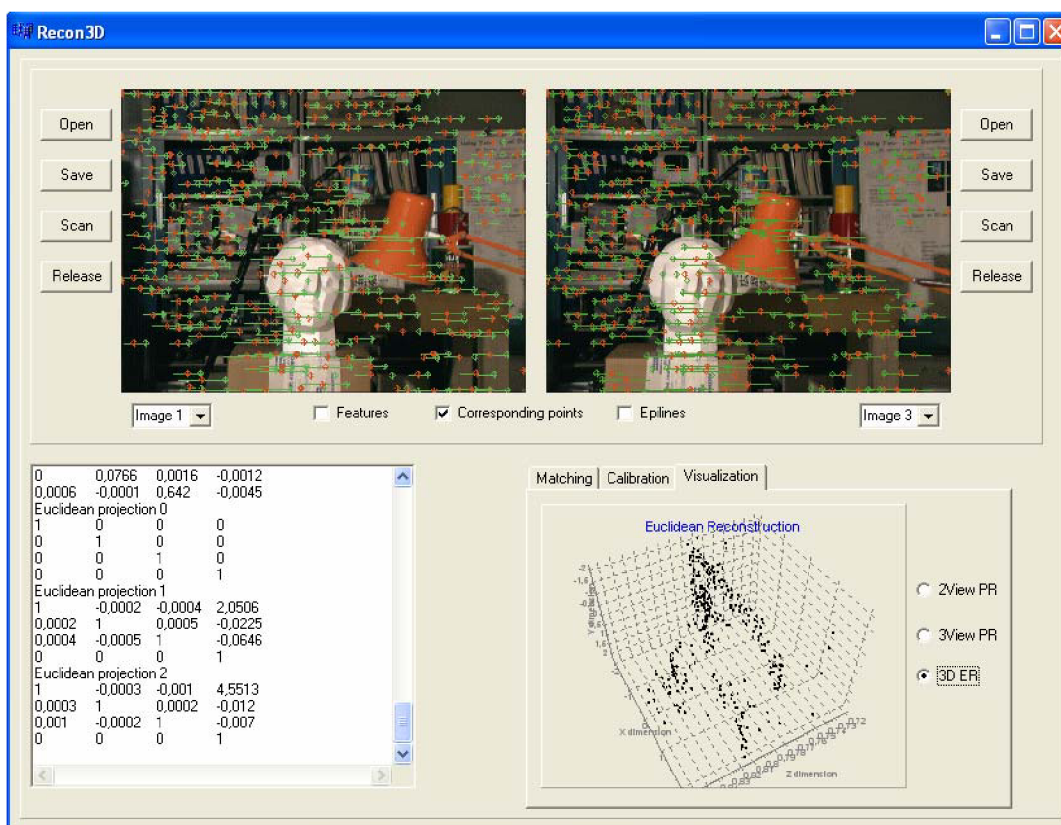
d) Vizualizace

Jakmile je kalibrace dána a vyřešen problém korespondence, výsledek, t.j. rekonstruovaná 3D souřadnice zvoleného či nalezeného bodu na snímku je zobrazená v 3D pohledu. Vzájemná poloha kamer je rovněž naznačena v 3D zobrazení jako souřadnicový systém kamery.

3.7.2 Komunikační okna programu Recon3D

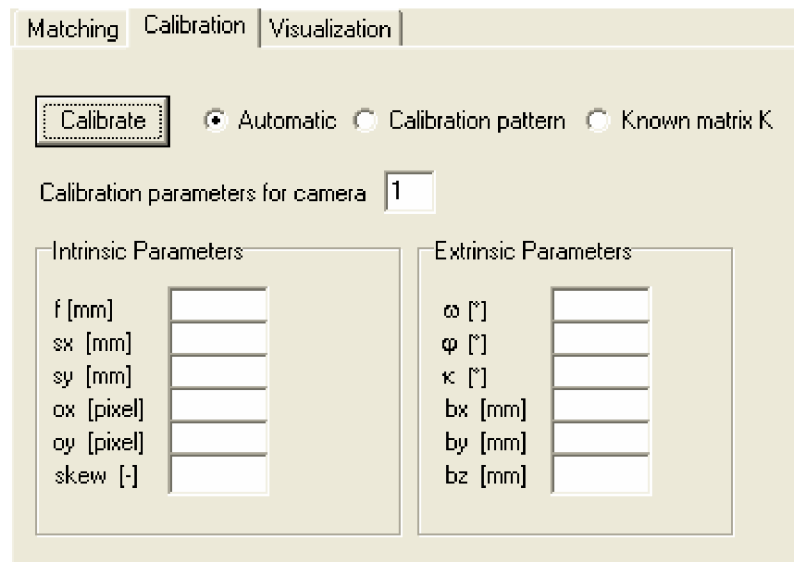
Základní okno vytvořeného programu Recon3D je zobrazeno na Obr. 26. Lze jej rozdělit na tři hlavní části. První část patří zobrazení dvojice snímků. Ke každému snímku jsou přiřazeny čtyři tlačítka. Pomocí nich je možné otevřít či uložit snímek z PC respektive do PC nebo se připojit ke kameře a získat snímek přímo z fotoaparátu či kamery. Druhá část, v pravém dolní rohu, obsahuje přepínatelné lišty. Lišta *Calibration* nabízí možnost volby mezi automatickou kalibrací, kalibrací pomocí kalibračního obrazce nebo manuální kalibrací, kdy je známá matice K . Jakmile je kalibrace provedena jsou kalibrační parametry zobrazeny do příslušných kolonek. Je možné rovněž zadat kalibrační parametry přímo, pokud jdou známy. V liště *Matching* je možné vybrat metodu nebo nastavit parametry použité v procesu vyhledávání totožných bodů na snímcích. Poslední lištou je lišta *Vizualization*, ve které je k dispozici 3D pohled rekonstruované scény. Ve třetí části okna, levém dolním rohu je zobrazováno hlášení generované během běhu programu, jakožto i informace o získaných parametrech.

Obrázek 26 ukazuje situaci při euklidovské rekonstrukci. Ztotožněné body zahrnuté do vlastní rekonstrukce jsou zobrazeny na snímcích zeleně. V liště *Visualization* je zobrazena výsledná 3D struktura získaná ze sesouhlasených bodů. Informace o získaných projektivních maticích je zobrazena v informačním okně.



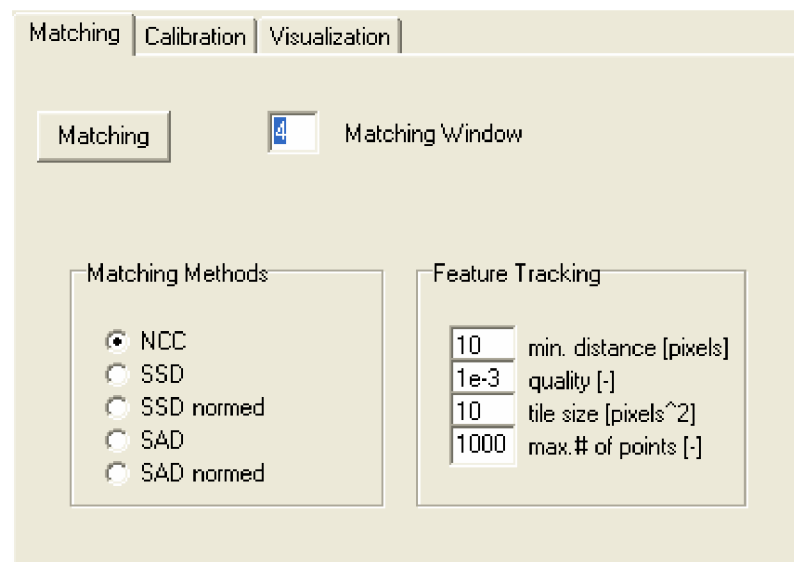
Obr.26 Okno testovacího programu Recon3D

Pohled na lištu *Calibration* je na Obr.27. Zde je možnost zvolit automatickou kalibraci, která využívá rozvrstveného přístupu při rekonstrukci. Je nutné mít trojici snímků. Další možností je kalibrace pomocí kalibračního obrazce. Pro tuto volbu je nutné mít v každém snímku viditelný kalibrační obrazec. Konečně třetí možnost představuje kalibraci, kdy jsou známy parametry matice K . Zobrazené hodnoty odpovídají parametrům kamery pro daný snímek.



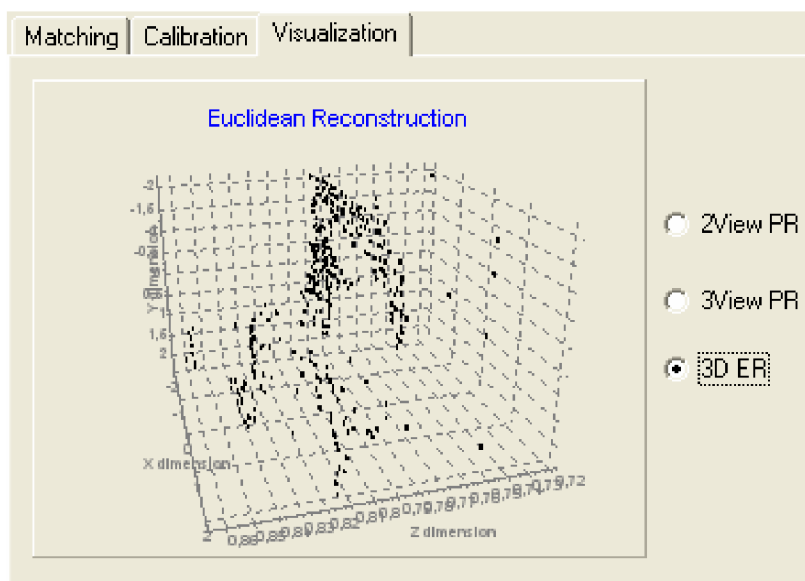
Obr.27 Vzhled lišty *Calibration*

Pohled na lištu *Matching* je na Obr.28. Zde se volí velikost okna použitého v procesu hledání totožných bodů pro zvolenou statistickou funkci. Parametry min. distance, quality a separation window jsou použity při vyhledávání význačných bodů.



Obr.28 Vzhled lišty *Matching*

Lišta *Visualization* (Obr.29) obsahuje 3Dgraf, ve kterém je možné zobrazit výsledné rekonstrukce pro dvousnímkovou a třísnímkovou projektivní rekonstrukci (2View PR a 3ViewPR0, jakožto i euklidovskou rekonstrukci nalezených bodů 3D ER).



Obr.29 Vzhled lišty *Visualization*


3.7.3 Základní ovládání programu Recon3D

Ovládání programu by mělo být intuitivní a jednotlivé úkony odovídají postupu při rekostrukci zobrazeného na Obr.21. Jednotlivé kroky jsou tedy následující: otevření snímků, nalezení bodů na snímcích a jejich ztotožnění, kalibrace kamery a výpočet prostorové struktury. Daný postup ovšem nemusí být dodržen přesně, neboť každá dílčí část vykoná nezbytné operace z předchozích kroků automaticky.

Otevření snímků. Jak bylo uvedeno dříve je možné sekvenci snímků otevřít z počítače, kde byly předtím uloženy, nebo využít připojeného fotoaparátu nebo digitální kamery. Je nutné otevřít alespoň dva snímky, které jsou poté zobrazeny v okně programu. Jelikož použitý algoritmus pro automatickou kalibraci a rekonstrukci využívá tří snímků je uživatel v daný okamžik vyzván k otevření třetího snímku. Je možné měnit právě zobrazovaný snímek pomocí výberového menu pod levým či pravým snímkem

Image 1 ▾.

Nalezení bodů na snímcích. Pro spuštění procesu nalezení bodů je možné na liště *Matching*. Zde si volíme metodu a parametry použité při hledání bodů a stiskneme tlačítko *Matching*. Jakmile jsou body nalezeny je možné je vykreslit do zobrazených snímků zaškrtnutím volby *Features*, která slouží jako přepínač pro zobrazení nalezených význačných bodů. Dále je nalezena korespondence mezi těmito body a pro právě zobrazované snímky je možné tyto ztotožněné body rovněž vykreslit do snímků, a to sice zaškrtnutím volby *Corresponding points*. Ztotožněné body a přímky, představující posun daného bodu přes další snímky, jsou zobrazeny zeleně. Červené body představují ztotožněné body na dalších snímcích. Zobrazení epipolárních přímek pro dané body je možné po zatržení volby *Epilines*. Epipolární přímky jsou vypočítané na základě ztotožněných bodů pomocí fundamentální matice F .

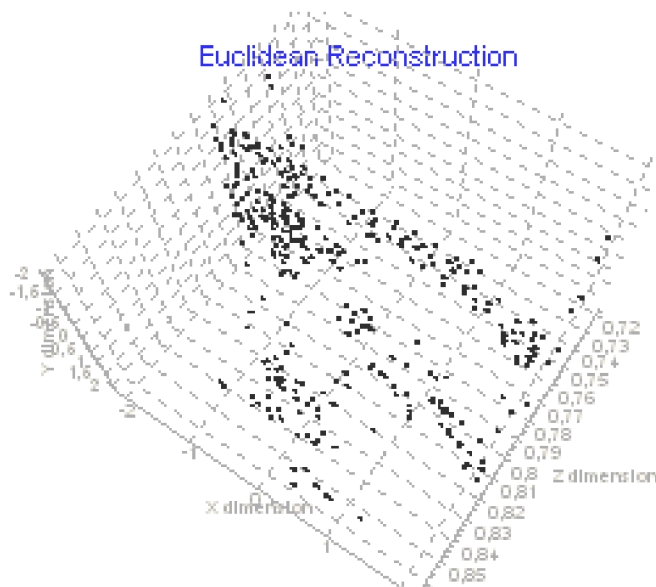
Kalibrace kamery. Při volbě automatické kalibrace je kalibrace kamery spojená s rekonstrukcí scény a je možné získat 3D strukturu scény až na násobný koeficient. Využívá se rozvrstveného přístupu a iterativního algoritmu minimalizující chybu zpětné projekce. Kalibrace je provedena po stisknutí tlačítka  na liště *Calibration*. Další alternativy jsou kalibrace pomocí plošného kalibračního vzoru a určení vzájemné polohy kamer pokud jsou známy prvky vnitřní orientace.

Vizualizace. Nalezená 3D struktura je zobrazována v 3D grafu na liště *Visualization*. Je možné přepínat mezi projektivní a euklidovskou rekonstrukcí rekonstrukcí. Graf umožňuje jeho natáčení pomocí levého tlačítka myši při současném pohybu myši, nebo zoomování volbou obdélníkové oblasti.

Na následujících obrázcích je zobrazena sekvence snímků a získaná 3D struktura pro automaticky nalezené a sesouhlasené body na snímcích



Obr.29 a) Sekvence vstupních snímků

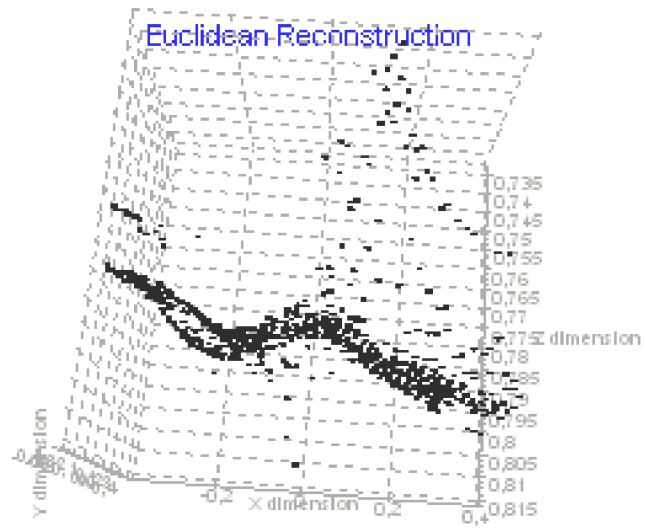


b)

Obr.29 b) Získaná 3D rekonstrukce



Obr.30 a) Sekvence vstupních snímků (převzato z [42])



Obr.29 b) Získaná 3D rekonstrukce

4 Závěr

Tématem této práce byla analýza a vývoj optoelektronických a fotogrammetrických systémů pro měření 2D a 3D rozměrů objektů snímaných kamerou s CCD nebo CMOS snímačem.

V první části práce, zabývající se měřením velikosti ploch a rozměrů rovinných objektů, byl předložen systémový návrh autonomních měřičů ploch pracujících na základě analogového i číslicového zpracování obrazového signálu kamery, snímající měřený objekt. Pro tyto dva systémy byla provedena analýza a z ní vyplývající odhad dosažitelné přesnosti měření na základě vyhodnocení systémových i dalších přídatných chyb dílčích funkčních bloků. Z této analýzy je patrné, že výsledná přesnost měření závisí především na rozlišení optoelektronického snímače (počtu obrazových elementů) použitého v kameře či fotoaparátu a také na využití celé plochy snímače při měření, to znamená na poměru měřené plochy k celému rastru snímače. Rovněž je zřejmé, že digitalizace analogového videosignálu nepřináší výrazné zlepšení výsledné přesnosti měření. Výhoda spočívá především v číslicovém zpracování obrazu, v možnosti velkého počtu způsobů vyhodnocování velikosti měřené plochy v případě, že data jsou přenesena do PC. V závěru této části práce byl vytvořen uživatelsky příjemný PC program AD-EVAL, sloužící k softwarovému vyhodnocení velikosti plochy, případně vzdálenosti mezi dvěma zvolenými body v plošné projekci snímané 2D či 3D scény. Program využívá k pořízení scény kamery či fotoaparátu připojených přes USB rozhraní, přičemž tyto vstupní data jsou uložena ve formátu BMP pro pozdější zpracování. Program nabízí snadnou volbu měřené oblasti, volbu módu měření, pohled na histogram snímku a především zobrazení výsledku měření.

Druhá část práce byla věnována stereofotogrammetrii. Byla zaměřená na popis algoritmů pro rekonstrukci prostorových souřadnic snímaných objektů. V úvodu této části byl nastíněn matematický popis vztahů mezi body v prostoru spolu s popisem geometrického modelu kamery. V následné kapitole jsou uvedeny postupy pro získání kalibračních parametrů kamery, to znamená prvků vnitřní a vnější orientace. Další stěžejní operací pro rekonstrukci prostorové scény je vlastní ztotožnění bodů na jednotlivých snímcích, proto zde byla část práce věnována právě této oblasti. Byly popsány dvě základní metody, globální a lokální, přičemž větší důraz je kladen na metody lokální, využívající především korelačních funkcí pro porovnávání bodů na snímcích. V další kapitole byl analyzován vliv nepřesného určení parametrů spojených s rekonstrukcí. Byly odhadnuty chyby v určení Z -tové souřadnice na chybně určeném úhlu natočení či posunutí kamery. Rovněž byl popsán vliv diskretizace obrazu. Ze získaných výsledků je patrné, že nejkritičtější pohybem je natočení kamer kolem osy y , dále posunutí v ose x a dále otočení kolem optické osy z . Také vlastní diskretizace má významný vliv na výslednou přesnost 3D rekonstrukce. V závěru této části byl popsán vytvořený PC program Recon3D sloužící k testování postupů při 3D rekonstrukci. Program opět využívá připojeného fotoaparátu či kamery k pořízení vstupní sekvence snímků. Ty mohou být uloženy do formátu BMP s osmi bitovou hloubkou pro pozdější zpracování. Je možné volit mezi automatickou kalibrací, kalibrací pomocí kalibračního obrazce a kalibrací při známých parametrech matice K . Program automaticky vyhledává

a sesouhlasuje body na snímcích , jejichž prostorové souřadnice pak mohou být zobrazeny v 3D grafu.

Hlavním přínosem této práce je především ucelený popis optoelektronických a fotogrammetrických systémů pro měření 2D a 3D rozměrů objektů snímaných kamerou spolu s analýzou dosažitelné přesnosti měření a vývojem dílčích systémů. Takovýto popis jsem doposud nenašel v literatuře, kterou jsem měl k dispozici

5 Literatura

- [1] BAKER, H.H., BINFORD, T.O. Depth from edge- and intensity-based stereo. In *Proceedings 7th Joint conference on Artificial Intelligence*, Vancouver, Canada, pages 631-636, August 1981.
- [2] BELHUMEUR, P. N. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision (IJCV)*, 19(3):237–262, 1996.
- [3] BELHUMEUR, P.N. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision (IJCV)*, 19(3):237–262, 1996.
- [4] BOBICK, A. F., INTILLE, S. S. Large occlusion stereo. *International Journal on Computer Vision*, 33(3):181–200, 1999.
- [5] BOUFAMA, B. AND MOHR, R. Epipole and fundamental matrix estimation using the virtual parallax property. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1030-1036, Boston, MA, 1995.
- [6] BOYKOV, Y., VEKSLER, O., ZABIH, R.. Disparity component matching for visual correspondence. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 470–475, 1997.
- [7] CANNY, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [8] COLLINS, R. AND WEISS, R. Vanishing point calculation as a statistical inference on the unit sphere. In *Proceedings of IEEE International Conference on Computer Vision*, pages 400-403, 1990.
- [9] COX I. J., HINGORANI S., MAGGS B. M., AND RAO S. B. Stereo without disparity gradient smoothing: a Bayesian sensor fusion solution. In D. Hogg and R. Boyle, editors, *Proceedings of British Machine Vision Conference*, pages 337–346, Leeds, UK, September 1992.
- [10] COX, I.J. A maximum likelihood n-camera stereo algorithm. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 733– 739, Seattle, Washington, 1994.
- [11] COX, I.J., HINGORANI, S.L., RAO, S.B., AND MAGGS, B.M. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.
- [12] FAUGERAS, O.D., LUONG, Q.T., AND MAYBANK, S. Camera self-calibration: theory and experiments. In *Proc. European Conference on Computer Vision*, LNCS588, pages 321–334. Springer-Verlag, 1992.
- [13] FAUREGAS, O. AND LUSTMANN, F. Motion and structure from motion in a piecewise planar environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):485-508, 1988
- [14] FAUREGAS, O. Stratification of three-dimensional vision: projective, affine, and metric representation. *Journal of the Optical Society of America*, 12(3):465-484, 1995.
- [15] FAUREGAS, O. *Three-Dimensional Computer Vision*. The MIT Press, 1993.
- [16] FAUREGAS, O. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proceedings of European Conference on Computer Vision*, pages 563-578. Springer-Verlag, 1992.
- [17] GEIGER, D., LADENDORF, B. AND YUILLE, A.. Occlusions and binocular stereo. *International Journal on Computer Vision*, 14:211–226, 1995.
- [18] GEIGER, D., LADENDORF, B., YUILLE, A. Occlusions and binocular stereo. *International Journal on Computer Vision*, 14:211–226, 1995.
- [19] GONG, M. AND YANG, Z-H. Multi-resolution stereo matching using genetic algorithm. In *Proceedings of Workshop on Stereo and Multi-Baseline Vision*, pages 21–29, Kauai, Hawaii, 2001.

- [20] GRIMSON, W.E.L. A Computer Implementation of a Theory of Human Stereo Vision, *Philosophical Transaction of the Royal Society of London*, B.,292(1058):217-253, 1981.
- [21] GRIMSON, W.E.L. Computational experiment with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17-34, 1985.
- [22] HARRIS, CH. AND STEPHENS, M. A combined corner and edge detector. In M. M. Matthews, editor, *Proceedings of the 4th ALVEY vision conference*, pages 147–151, University of Manchester, England, 1988.
- [23] HARTLEY, R. AND STURM, P. Triangulation. *Computer Vision and Image Understanding*, 68(2):146-157, 1997
- [24] HARTLEY, R. In defence of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580-593, 1997
- [25] HARTLEY, R., GUPTA, R. AND CHANG, T. Stereo from uncalibrated cameras. In *Proceedings of international Conference on Computer Vision and Pattern Recognition*, pages 761-764, Urbana Champaign, IL, USA, IEEE Comput. Soc. Press, 1992.
- [26] HEYDEN, A. AND SPARR, G. Reconstruction from calibrated cameras – a new proof of Kruppa Demazure theorem. *Journal of Mathematical Imaging and Vision*, pages 1-20, 1999.
- [27] HIRSCHMÜLLER, H. Improvements in real-time correlation-based stereo vision. In *Proceedings of Workshop on Stereo and Multi-Baseline Vision*, pages 141–148, Kauai, Hawaii, 2001.
- [28] HUANG, T. AND FAUREGAS, O. Some properties of E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310-1312, 1989
- [29] CHAI, J. AND MA, S. Robust epipolar geometry estimation using generic algorithm. *Pattern Recognition Letters*, 19(9):829-838, 1998.
- [30] CHEN, Q AND MEDIONI, G. A volumetric stereo matching method: Application to image-based modeling. In *Proceedings of International Conference On Computer Vision and Pattern Recognition*, pages 29–34, 1999.
- [31] ISHIKAWA, H. AND GEIGER, D. Occlusions, discontinuities, and epipolar lines in stereo. In *Proceedings of the 5th European Conference on Computer Vision*, Freiburg, Germany, June 1998.
- [32] KANADE, T., OKUTOMI, M. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.
- [33] KANATANI, K. Detecting the motion of planar surface by line & surface integrals. In *Computer Vision, Graphics, and Image Processing*, volume 29, pages 13-22, 1985.
- [34] KANATANI, K. *Geometric Computation for Machine Vision*. Oxford Science Publications, 1993.
- [35] KOLMOGOROV, V AND ZABIH, R. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada, July 2001.
- [36] LIBOWITZ, D, ZISSERMAN, A. Combining scene and autocalibration constrains. In *Proceedings of international Conference on Computer Vision*, pages I: 293-330, 1999.
- [37] LONGUET-HIGGINS, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-135, 1981.
- [38] LONGUET-HIGGINS, H. C. Reconstruction of a plane surface from two perspective projections. In *Proceedings of Royal Society of London*, volume 227 of B, pages 399-410, 1986.
- [39] LUONG, Q.-T. AND FAUREGAS, O. The fundamental matrix: theory, algorithms, and stability analysis. *Int. Journal of Computer vision*, pages 589-599, 1994.
- [40] LUTTON, E., MAITRE, H., AND LOPEZ-KRAHE, J. Contribution to the determination of vanishing points using Hough transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):430-438, 1994.

- [41] MA, Y., KOSECKA, J. AND SASTRY, S. optimization criteria and geometric algorithms for motion and structure estimation. *Int. Journal of Computer vision*, 44(3):219-249, 2001.
- [42] MA, Y., SOATTO, S., KOSECKA, J., AND SASTRY, S. An Invitation to 3-D Vision: Form Images to Geometric Models. Springer-Verlag, New York. 2004. 526 p. ISBN 0-387-00893-4.
- [43] MA, Y., SOATTO, S., KOSECKA, J., AND SASTRY, S. Euklidean reconstruction and reprojection up to subgroups. In *Proceedings of IEEE International Conference on Computer Vision*, pages 773-780, 1999.
- [44] MARR, D. AND HILDRETH, E. Theory of edge detection. *Proceedings Royal Society London*, B 207:187-217, 1980.
- [45] MARR, D. AND POGGIO, T. A computational theory of human stereo vision. *Proc. R. Soc.*, B-204:301-328, 1979.
- [46] MARR, D. AND POGGIO, T. Cooperative computation of stereo disparity. *Science*, 194:283-287, 1976
- [47] MAYBANK, S. AND FAUREGAS, O. A theory of selfcalibration of moving camera. *Int. Journal of Computer vision*, 8(2):123-151, 1992.
- [48] MAYBANK, S., Theory of Reconstruction from Image Motion, Springer Series in Information Science, Springe-Verlag, 1993.
- [49] OHTA, Y AND KANADE, T. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139-154, 1985.
- [50] OKUTOMI, M. AND KANADE, T. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353-363, 1993.
- [51] OKUTOMI, M., KATAYAMA, Y., AND OKA S. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. In *Proceedings of Workshop on Stereo and Multi-Baseline Vision*, pages 158-165, Kauai, Hawaii, 2001.
- [52] OKUTOMI, M., KATAYAMA, Y., OKA, S. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. In *Proceedings of Workshop on Stereo and Multi-Baseline Vision*, pages 158-165, Kauai, Hawaii, 2001.
- [53] PHILIP, J. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *Photogrammetric Records*, 15(88):589-599, 1996.
- [54] POLLARD, S.B., MAYHEW, J.E.W, AND FRISBY, J.P. PMF: a stereo correspondence algorithm using a disparity gradient constraint. *Perception*, 14:449-470, 1985
- [55] PONCE, J., MARIMONT, D., AND CASS, T. Analytical methods for uncalibrated stereo and motion reconstruction. In *Proceedings of European Conference on Computer Vision*, pages 463-470. 1994.
- [56] QUAN, L. Affine stereo calibration for relative affine shape reconstruction. In *Proceeding of British Machine Vision Conference*, pages 659-668, 1993.
- [57] QUAN, L., MOHR, R. Determining perspective structures using hierarchical Hough transform. *Pattern recognition Letter*, 9(4):279-286, 1989.
- [58] QUANN, L. AND MOHR, R. Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letter*, 9(4):279-286, 1989.
- [59] ROY, S. AND COX, I.J. A maximum-flow formulation of the n-camera stereo correspondence problem. In *The 6th International Conference on Computer Vision*, 1998.
- [60] ŘÍČNÝ, V. Contactless Opto-Electronic Area Meters and their Attainable Measuring Accuracy. *Radioengineering*, roč. 2001, č. 2, s. 20, ISSN 1210-2512
- [61] SCHARSTEIN, D. SZELISKI, R. A taxonomy and evaluation of dense twoframe stereo correspondence algorithms. Technical Report MSR-TR-2001-81, Microsoft Corporation, Redmond, WA 98052, USA, 2001.

- [62] SCHARSTEIN, D., SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002.
- [63] STANČÍK, P. *Televizní bezdotykový měřič ploch*, Diplomová práce UREL FEKT VUT Brno, 2000
- [64] STANČÍK, P. Digitální TV planimetr. *Elektrorevue - Internetový časopis*, (<http://www.elektrorevue.cz>), ISSN 1213-1539, 2002, roč. 2002, č. 75
- [65] STANČÍK, P. Algorithm for 3D Point Coordinates Reconstruction. In *RADIOELEKTRONIKA 2004 - Conference Proceedings*. 2004. s. 323648. ISBN: 8022720178.
- [66] STANČÍK, P. System for contactless surface measuring. In *Proceedings of 8th Conference STUDENT EEICT 2002, volume 3.*, FEKT VUT Brno, 2002, p.347 - 350. ISBN 80-214-2116-9.
- [67] STANČÍK, P. TV contactless area meter. In *Proceedings of 12th International Czech-Slovak Scientific Conference RADIOELEKTRONIKA 2002*. STU Bratislava, 2002, p. 425 - 428. ISBN 80-227-1700-2.
- [68] STURM, P. Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *Proceedings of international Conference on Computer Vision and Pattern Recognition*, pages 1000-1005, 1997.
- [69] STURM, P. Critical motion sequences for self-calibration of cameras and stereo systems with variable focal length. In *Proceeding of British Machine Vision Conference*, pages 63-72, 1999.
- [70] SUBBARAO, M. AND WAXMAN, A.M. On the uniqueness of image flow solution for planar surfaces in motion. In *Proceedings of the third IEEE workshop on computer vision: representation and control*, pages 129-140, 1985.
- [71] SZELISKI, R. AND SCHARSTEIN, D. Symmetric sub-pixel stereo matching. In *Proceedings 7th European Conference on Computer Vision*, volume 2, pages 525-540, Copenhagen, Denmark, 2002.
- [72] SZELISKI, R. AND SCHARSTEIN, D. Symmetric sub-pixel stereo matching. In *Proceedings 7th European Conference on Computer Vision*, volume 2, pages 525-540, Copenhagen, Denmark, May 2002.
- [73] ŠARA, R. Finding the largest unambiguous component of stereo matching. In *Proceedings 7th European Conference on Computer Vision*, volume 3, pages 900-914, Copenhagen, Denmark, May 2002.
- [74] ŠONKA, M., HLAVÁČ, V., AND BOYLE, R.D. *Image Processing, Analysis and Machine Vision*. PWS, Boston, USA, second edition, 1998.
- [75] TAO, H., SAWHNEY, H. S., KUMAR, R. A global matching framework for stereo computation. In *Proceedings 8th International Conference on Computer Vision*, volume 1, pages 532-539, Vancouver, Canada, 2001.
- [76] THOMPSON, E. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3(14):152-159.
- [77] TORR, P.H.S. AND MURRAY, D.W. The development and comparison of robust methods for estimating the fundamental matrix. *Int. Journal of Computer vision*, 24(3):271-300, 1997.
- [78] TORR, P.H.S., FITZGIBBON, A. AND ZISSERMAN, A. The problem of degeneracy in structure and motion recovery from uncalibrated images sequences. *Int. Journal of Computer vision*, 32(1):27-44, 1999.
- [79] TRIGGS, B. Autocalibration from planar scenes. In *Proceedings of European Conference on Computer Vision*, pages I: 89-105, 1998.
- [80] TSAI, R. Anefficient and accurate camera calibration technique for 3D machine vision. In *Proc. Computer Vision and Pattern Recognition*, 1986.

- [81] ZELLER, C. AND FAUREGAS, O. Camera self-calibration, from video sequences: the Kruppa equation revisited. *Research report 2793, INRIA, France*, 1996.
- [82] ZHANG, Z. A flexible new technique for camera calibration. *Microsoft Technical Report MSR-TR-98-71*, 1998.
- [83] ZHANG, Z. Determining the epipolar geometry and its uncertainty: a review. *Int. Journal of Computer vision*, 27(2):161-195, 1998.
- [84] ZHANG, Z. Understanding the relationship between the optimization criteria in two-view motion analysis. In *Proceedings of international Conference on Computer Vision*, pages 772-777, Bombay, India, 1998.
- [85] ZHANG, Z., DERICE, R, FAUREGAS, O. AND LUONG, Q.-T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87-119, 1995.
- [86] ZHAO W., NANDHAKUMAR, N. Effects of Camera Alignment Errors on Stereoscopic Depth Estimates. *Pattern Recognition* 29:2115-2126, 1996.
- [87] ZISSERMAN, A., LIEBOWITZ, D. AND ARMSTRONG, M. Resolving ambiguities in auto-calibration. *Philosophical Transactions of the Royal Society of London*, 356(1740):1193-1211.

6 Seznam použitých symbolů a zkratek

Optoelektronické měřiče

Symboly

f_0	horní mezní kmitočet kmitočtové charakteristiky kanálu
f_g	frekvence kalibračního oscilátoru
n_h	počet obrazových elementů CCD snímače v horizontálním směru
n_v	počet řádků CCD snímače
t_{al}	aktivní dobu řádku pro TV normu CCIR B,D
U_{pp}	mezivrcholové napětí za špičkovým usměrňovačem.
ΔH_q	absolutní chyba vodorovného rozměru
ΔV_q	absolutní chyba svislého rozměru
δ_{fq}	stabilita kalibračního oscilátoru
δ_{hq}	systemovou relativní chybu měření, způsobená rozlišením kamery ve vodorovném směru
δ_{nAGC}	relativní chybu v počtu čítaných vzorků způsobenou kolísáním zesílení předzesilovače δ_{AD} relativní chybu způsobená A/D převodem
δ_{tkomp}	relativní chybu měřeného časového intervalu t_k , způsobená kolísáním komparační úrovně
δ_{tkomp}	relativní chybu měřeného časového intervalu t_k , měnící se vlivem posunu ss. složky obrazového signálu
δ_{tndc}	relativní chybu v počtu čítaných vzorků způsobenou kolísáním ss. složky obrazového signálu
δ_{vq}	systemová relativní chyba měření vzniklá vzorkováním svislého rozměru

Zkratky

API	rozhraní pro programování aplikací (Application Programming Interface)
BMP	bitmapa, grafický formát (Microsoft Windows Bitmap)
DLL	dynamická knihovna (dynamic linking library)
LIB	statická knihovna (static linking library)
SDK	softwarový vývojový kit

Stereofotogrammetrie

Symboly

$\hat{\mathbf{u}}$	antisymetrická matice
\mathbf{x}_i^j	Souřadnice j-tého bodu na i-tém snímku
$\mathbf{M}_1 \otimes \mathbf{M}_2$	Kroneckerův produkt dvou matic nebo vektorů
b_x	báze kamer.
d	disparita, rozdíl snímkových souřadnic odpovídajících si bodů na dvou snímcích
$\det(\mathbf{M})$	determinant matice \mathbf{M}
e	epipola; projekce středu projekce jedné kamery na snímek druhé kamery
\mathbf{E}	Esenciální matice
f	ohnisková vzdálenost
\mathbf{F}	Fundamentální matice
g	transformace souřadnic mezi dvěma systémy $g = (\mathbf{R}, \mathbf{T})$, \bar{g} její homogenní reprezentace
\mathbf{K}	kalibrační matice kamery
l	epipolární přímka
\mathbf{M}^δ	přeskupená veze matice \mathbf{M} , všechny sloupce přeskupeny do jednoho vektoru
\mathbf{M}^T	Transponovaná matice \mathbf{M}
o_x	x-ová souřadnice středu snímku v pixelech
o_y	y-ová souřadnice středu snímku v pixelech
\mathbf{R}	ortonormální matice rotace 3x3
s_θ	šikmost pixelu, často blízko nule.
s_x	velikost pixelu v horizontálních směru
s_y	velikost pixelu ve vertikálním směru
\mathbf{T}	vektor posunutí
$\text{trace}(\mathbf{M})$	suma všech prvků matice \mathbf{M} na diagonále
\mathbf{x}	souřadnice $[x,z]$ bodu na snímku, $\mathbf{x} \doteq [x, y, 1]^T$ v homogenním vyjádření
\mathbf{X}	souřadnice $\mathbf{X}=[X,Y,Z]^T$ bodu v prostoru, v homogenním vyjádření $\mathbf{X} = [X, Y, Z, 1]^T$
\mathbf{x}_i	Souřadnice bodu na i-tém snímku
$\mathbf{\Pi}$	obecná matice projekce
$\mathbf{\Pi}_0$	standardní matice projekce
β	chybový úhel rotace kolem osy y
γ	chybový úhel rotace kolem osy z
φ	úhel rotace kolem osy y

κ	úhel rotace kolem osy z
ω	úhel rotace kolem osy x

Zkratky

CC	kros-korelace
DSI	obraz prostoru disparit
NCC	normalizovaná kros-korelace
SAD	suma absolutních diferencí
SSD	suma kvadrátů diferencí
SSDN	normalizovaná SSD
SVD	singular value decomposition
ZSAD	zero-mean SAD
ZSSD	zero-mean SSD

7 Seznam příloh

- Příloha A Výpis stěžejních algoritmů implementovaných v programu Recon3D
Příloha B CD-ROM se zdrojovými kódy programů AD-EVAL a Recon3D

Curriculum Vitae

Jméno: Petr STANČÍK

Narozen: 29.4. 1978 ve Vsetíně

Kontakt: stancik@feec.vutbr.cz

Vzdělání

1996 – 2001 **Vysoké učení technické v Brně / Fakulta elektrotechniky a informatiky.** Inženýrské studium oboru Elektronika a stělovací technika. Státní zkouška složena v červnu 2001. Diplomová práce na téma “Televizní bezdotykový měřič ploch” obhájena v červnu 2001.

2001 – 2004 **Vysoké učení technické v Brně / Fakulta elektrotechniky a komunikačních technologií, ústav radioelektroniky.** Interní doktorské studium oboru Elektronika a sdělovací technika. Státní doktorská zkouška složena v červnu 2003.

Zaměstnání

2004 – dosud **Freescale Polovodiče ČR, s.r.o.** Systémový aplikační inženýr, Technické Informační Centrum.

Jazyky

angličtina

Výpis stěžejních algoritmů implementovaných v programu Recon3D

```
// -----
// Find features on given image
// -----
int __fastcall TForm1::FindFeatures(IplImage *img, CvPoint2D32f* points, int
cnt)
{
    IplImage* eigL=cvCreateImage(cvSize(imageL->width,imageL->height),32,1);
    IplImage* tempL=cvCreateImage(cvSize(imageL->width,imageL->height),32,1);

    double quality = Quality->Text.ToDouble();
    double min_distance = MinDistance->Text.ToDouble();
    int cnt=MaxNumPoints->Text.ToInt();

    // detect features on image img
    cvGoodFeaturesToTrack( img, eigL, tempL, points, &cnt,
        quality, min_distance, 0, 3, 0, 0.04 );

    // refine position of found features, store them in points buffer
    cvFindCornerSubPix( img, points, cnt,
        cvSize(10,10), cvSize(-1,-1),
        cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,20,0.03));

    cvReleaseImage( &eigL ); cvReleaseImage( &tempL );
    return(cnt);
}

// -----
// Perform matching based on correlation algorithms
// -----
float __fastcall TForm1::MatchingPoint1(int Xt,int Yt,IplImage *tmp,int Xi,int
Yi,IplImage *img)
{
    int i,j,IMAGE_WIDTH,IMAGE_HEIGHT,RESULT_WIDTH,RESULT_HEIGHT;
    char *ptr1;
    float res;
    int method;
    int win = MatchWindow->Text.ToInt();

    // select appropriate matching method
    if(NCC->Checked) method = CV_TM_CCOEFF_NORMED;
    if(SSD->Checked) method = CV_TM_SQDIFF;
    if(SSDNorm->Checked) method = CV_TM_SQDIFF_NORMED;

    // select portion of images equal to window size
    cvSetImageROI( tmp, cvRect(Xt-win,Yt-win,2*win+1,2*win+1) );
    if(Xi!=-1)
    {
        cvSetImageROI( img, cvRect(Xi-win,Yi-win,2*win+1,2*win+1) );
        RESULT_WIDTH = img->roi->width - tmp->roi->width + 1;
        RESULT_HEIGHT = img->roi->height - tmp->roi->height + 1;
    }
    else // result's dimensions must be computed this way
    {
        RESULT_WIDTH = img->width - tmp->roi->width + 1;
        RESULT_HEIGHT = img->height - tmp->roi->height + 1;
    }
    IplImage* result; // result
    result=cvCreateImage(cvSize(RESULT_WIDTH,RESULT_HEIGHT), IPL_DEPTH_32F, 1);

    // perform matching
    cvMatchTemplate( img, tmp, result, method );
    // save resulted matching cost
    if(Xi!=-1) res = *(float*)result->imageData;
    else
    {
        cvMinMaxLoc(result, &minval, &maxval, &minLoc, &maxLoc, 0);
    }
}
```

```

        if((SSD->Checked) || (SSDNorm->Checked))
            {res=minval;respos.x=minLoc.x+win;respos.y=minLoc.y+win;}
        else {res=maxval;respos.x=maxLoc.x+win;respos.y=maxLoc.y+win;}
    }
    //Memo1->Lines->Add("res: "+CurrToStr(res)+" ");
    cvResetImageROI(img);cvResetImageROI(tmp);
    cvReleaseImage(&result);

    return res;
}

//-----
// Find correspondences over all images
//-----
void __fastcall TForm1::MatchFeatures5(IplImage *imgL, IplImage *imgR, int m)
{
    int cntL,cntR;
    CvPoint2D32f* pointsOF[2];
    char* status1 = 0,*status2 = 0;
    IplImage *pyrL=0,*pyrR=0;
    float dx=0,dy=0;

    double quality = Quality->Text.ToDouble();
    double min_distance = MinDistance->Text.ToDouble();
    int max_count = MaxNumPoints->Text.ToInt();

    // open third image
    if(m>2)
    {
        if(OpenPictureDialog1->Execute())
        {
            Bitmap1->LoadFromFile(OpenPictureDialog1->FileName);
            imageR1=cvCreateImage(cvSize(Bitmap1->Width,Bitmap1->Height),
                                IPL_DEPTH_8U,1);
            imageR1 = cvLoadImage(OpenPictureDialog1->FileName.c_str(),0);
        }
    }

    // find features on all images if not found before
    if(!cntF[0]) cntF[0]=FindFeatures(imageL,pointsF[0],cntF[0]);
    if(!cntF[1]) cntF[1]=FindFeatures(imageR1,pointsF[1],cntF[1]);
    if(!cntF[2]) cntF[2]=FindFeatures(imageR,pointsF[2],cntF[2]);
    Memo1->Lines->Add("Found corners in right image: "+CurrToStr(cntF[0])+" ");

    pointsOF[0] = (CvPoint2D32f*)cvAlloc(max_count*sizeof(pointsOF[0][0]));
    pointsOF[1] = (CvPoint2D32f*)cvAlloc(max_count*sizeof(pointsOF[0][0]));
    status1 = (char*)cvAlloc(max_count);
    flags = 0; cntL=cntF[0];

    // find correspondences between two image based on optical flow
    cvCalcOpticalFlowPyrLK( imgL, imageR1, pyrL, pyrR,
        pointsF[0], pointsOF[1], cntL, cvSize(10,10), 3, status1, 0,
        cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,20,0.03), flags );

    // save found correspondences into buffers
    count1 = 0;
    for(int i=0;i<cntL;i+=1)
    {
        points[0][count1]=pointsF[0][i];
        points[1][count1]=pointsOF[1][i];
        dx+=fabs(points[0][count1].x-points[1][count1].x);
        dy+=fabs(points[0][count1].y-points[1][count1].y);
        count1++;
    }
    dx/=count1;
    dy/=count1; cntL=0;

    // select just correct correspondences
    for(int i=0;i<count1;i++)
    {
        if((fabs(points[0][i].x-points[1][i].x)<4*dx)&&
            (fabs(points[0][i].y-points[1][i].y)<dy))

```



```

        {
            points[0][cntL]=points[0][i];
            points[1][cntL]=points[1][i];
            cntL++;
        }
    }
    count1 = 0;

    status2 = (char*)cvAlloc(max_count);
    flags = 0; pyrL=NULL; pyrR=NULL;

    // find correspondences between another two image based on optical flow
    cvCalcOpticalFlowPyrLK( imageR1, imgR, pyrR, pyrL,
        points[1], pointsOF[1], cntL, cvSize(10,10), 3, status2, 0,
        cvTermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS,20,0.03), flags );

    // save found correspondences into buffers and select just correct
    // correspondences
    for(int i=0;i<cntL;i+=1)
    {
        points[1][count1]=points[1][i];
        points[2][count1]=pointsOF[1][i];
        dx+=fabs(points[1][count1].x-points[2][count1].x);
        dy+=fabs(points[1][count1].y-points[2][count1].y);
        count1++;
    }
    dx/=count1;
    dy/=count1; cntL=0;
    for(int i=0;i<count1;i++)
    {
        if((fabs(points[1][i].x-points[2][i].x)<4*dx)&&
            (fabs(points[1][i].y-points[2][i].y)<dy))
        {
            points[0][cntL]=points[0][i];
            points[1][cntL]=points[1][i];
            points[2][cntL]=points[2][i];
            cntL++;
        }
    }
    count1 = cntL;
    Memol->Lines->Add("Found correspondences: "+CurrToStr(count1)+" ");
}

//-----
// Compute fundamental matrix from a set of corresponding points
// based on 8-point algorithm
//-----
void __fastcall TForm1::DFundamental(CvMat *pa, CvMat *qa, CvMat *fund)
{
    int npoints = pa->cols;

    // normalization of image coordinates
    float mi_x1=0.0, mi_y1=0.0, sig_x1=0.0, sig_y1=0.0;
    float mi_x2=0.0, mi_y2=0.0, sig_x2=0.0, sig_y2=0.0;
    for (int i=0; i<npoints; i++)
    {
        mi_x1+=cvmGet(pa,0,i); mi_x2+=cvmGet(qa,0,i);
        mi_y1+=cvmGet(pa,1,i); mi_y2+=cvmGet(qa,1,i);
    }
    mi_x1/=npoints; mi_x2/=npoints;
    mi_y1/=npoints; mi_y2/=npoints;
    for (int i=0; i<npoints; i++)
    {
        sig_x1 += (cvmGet(pa,0,i)-mi_x1)*(cvmGet(pa,0,i)-mi_x1);
        sig_x2 += (cvmGet(qa,0,i)-mi_x2)*(cvmGet(qa,0,i)-mi_x2);
        sig_y1 += (cvmGet(pa,1,i)-mi_y1)*(cvmGet(pa,1,i)-mi_y1);
        sig_y2 += (cvmGet(qa,1,i)-mi_y2)*(cvmGet(qa,1,i)-mi_y2);
    }
    sig_x1 = cvSqrt(sig_x1/npoints);
    sig_x2 = cvSqrt(sig_x2/npoints);
    sig_y1 = cvSqrt(sig_y1/npoints);
}

```

```

sig_y2 = cvSqrt(sig_y2/npoints);
CvMat *H1=cvCreateMat(3,3,CV_32F);
CvMat *H2=cvCreateMat(3,3,CV_32F);

cvmat_assign33(H1,1./sig_x1,0,-mi_x1/sig_x1,0,1./sig_y1,
               -mi_y1/sig_y1,0,0,1);
cvmat_assign33(H2,1./sig_x2,0,-mi_x2/sig_x2,0,1./sig_y2,
               -mi_y2/sig_y2,0,0,1);

CvMat *p=cvCreateMat(3,npoints,CV_32F);
CvMat *q=cvCreateMat(3,npoints,CV_32F);
for (int i=0; i<npoints; i++)
{
    cvmSet(p,0,i,((1./sig_x1)*cvmGet(pa,0,i)-mi_x1/sig_x1));
    cvmSet(p,1,i,((1./sig_y1)*cvmGet(pa,1,i)-mi_y1/sig_y1));
    cvmSet(p,2,i,1);

    cvmSet(q,0,i,((1./sig_x2)*cvmGet(qa,0,i)-mi_x2/sig_x2));
    cvmSet(q,1,i,((1./sig_y2)*cvmGet(qa,1,i)-mi_y2/sig_y2));
    cvmSet(q,2,i,1);
}

CvMat *A = cvCreateMat(npoints,9,CV_32F);
// do a kronecker product from input points
for (int i=0; i<npoints; i++)
{
    cvmSet(A,i,0,cvmGet(p,0,i)*cvmGet(q,0,i));
    cvmSet(A,i,1,cvmGet(p,0,i)*cvmGet(q,1,i));
    cvmSet(A,i,2,cvmGet(p,0,i)*cvmGet(q,2,i));
    cvmSet(A,i,3,cvmGet(p,1,i)*cvmGet(q,0,i));
    cvmSet(A,i,4,cvmGet(p,1,i)*cvmGet(q,1,i));
    cvmSet(A,i,5,cvmGet(p,1,i)*cvmGet(q,2,i));
    cvmSet(A,i,6,cvmGet(p,2,i)*cvmGet(q,0,i));
    cvmSet(A,i,7,cvmGet(p,2,i)*cvmGet(q,1,i));
    cvmSet(A,i,8,cvmGet(p,2,i)*cvmGet(q,2,i));
}

CvMat *U = cvCreateMat(npoints,9,CV_32F);
CvMat *S = cvCreateMat(9,9,CV_32F);
CvMat *V = cvCreateMat(9,9,CV_32F);

// compute SVD on matrix A
cvSVD(A,S,U,V, CV_SVD_MODIFY_A);

// pick the eigenvector corresponding to the smallest eigenvalue
//CvMat *f = cvCreateMat(9,1,CV_32F);
CvMat *est =cvCreateMat(3,3,CV_32F);

cvmat_assign33(est,cvmGet(V,0,8),cvmGet(V,3,8),cvmGet(V,6,8),
               cvmGet(V,1,8),cvmGet(V,4,8),cvmGet(V,7,8),
               cvmGet(V,2,8),cvmGet(V,5,8),cvmGet(V,8,8));

CvMat *fa = cvCreateMat(3,3,CV_32F);
CvMat *pom = cvCreateMat(3,3,CV_32F);

cvGEMM(H2,est,1.,0,0.,pom,CV_GEMM_A_T);
cvGEMM(pom,H1,1.,0,0.,fa,0);

CvMat *UA = cvCreateMat(3,3,CV_32F);
CvMat *SA = cvCreateMat(3,3,CV_32F);
CvMat *VA = cvCreateMat(3,3,CV_32F);

cvSVD(fa,SA,UA,VA, CV_SVD_MODIFY_A+ CV_SVD_V_T);

cvmSet(SA,2,2,0);

CvMat *F = cvCreateMat(3,3,CV_32F);
cvGEMM(UA,SA,1.,0,0.,pom,0);
cvGEMM(pom,VA,1.,0,0.,F,0);

```

```

// release used matrixes
cvReleaseMat (&H1); cvReleaseMat (&H2);
cvReleaseMat (&p); cvReleaseMat (&q);
cvReleaseMat (&A); cvReleaseMat (&U); cvReleaseMat (&S); cvReleaseMat (&V);
cvReleaseMat (&f); cvReleaseMat (&fa); cvReleaseMat (&pom);
//cvReleaseMat (&pom1);
cvReleaseMat (&F); cvReleaseMat (&UA); cvReleaseMat (&SA); cvReleaseMat (&VA);
}

//-----
// compute 3D structure form multiple views
//-----
void __fastcall TForm1::CalibrateButtonClick(TObject *Sender)
{
    CvPoint2D32f temp1,temp2;
    int m=3;
    int win = MatchWindow->Text.ToInt();

    Series1->Clear();
    count1=0;
    if (AutoCalBtn->Checked)
    {
        // write necessary action here
    }
    else
    {
        // write necessary action here
    }

    // intialize calibration_matrix
    CvMat *Aguess = cvCreateMat (3,3,CV_32F);
    CvMat *invAguess = cvCreateMat (3,3,CV_32F);
    cvmSet (Aguess,0,0,imageL->width*1.5); cvmSet (Aguess,0,1,0);
    cvmSet (Aguess,0,2,imageL->width/2);
    cvmSet (Aguess,1,0,0); cvmSet (Aguess,1,1,imageL->height*1.5);
    cvmSet (Aguess,1,2,imageL->height/2);
    cvmSet (Aguess,2,0,0); cvmSet (Aguess,2,1,0); cvmSet (Aguess,2,2,1);

    cvInvert (Aguess,invAguess,CV_LU);

    PrintMat (Aguess,0,0,3,3);
    PrintMat (invAguess,0,0,3,3);

    CopyMat (iAguess,0,0,invAguess,0,0,3,3);

    CvMat *pointsMat1, *pointsMat1n,*epilinesCoef1,*pointMat, *pointMatn;
    CvMat *pointsMat2, *pointsMat2n,*epilinesCoef2, *eCoef2,*pointMat1;
    CvMat *fund = cvCreateMat (3,3,CV_32F);
    CvMat *fundT = cvCreateMat (3,3,CV_32F);

    // find correspondences
    MatchFeatures5 (imageL,imageR,m);

    pointMat = cvCreateMat (3*m,count1,CV_32F);
    // save it into matrix
    for( int i = 0; i < count1; i++)
    {
        cvmSet (pointMat,0,i,points[0][i].x); //x
        cvmSet (pointMat,1,i,points[0][i].y);
        cvmSet (pointMat,2,i,1.0);
        cvmSet (pointMat,3*(m-1)+0,i,points[2][i].x); //x
        cvmSet (pointMat,3*(m-1)+1,i,points[2][i].y); //y
        cvmSet (pointMat,3*(m-1)+2,i,1.0);
        if (m>2)
        {
            cvmSet (pointMat,3*(m-2)+0,i,points[1][i].x); //x
            cvmSet (pointMat,3*(m-2)+1,i,points[1][i].y); //y
            cvmSet (pointMat,3*(m-2)+2,i,1.0);
        }
    }
}

```

```

pointsMat1 = cvCreateMat (3, count1, CV_32F);
pointsMat2 = cvCreateMat (3, count1, CV_32F);
pointsMat1n = cvCreateMat (3, count1, CV_32F);
pointsMat2n = cvCreateMat (3, count1, CV_32F);
//pointMat = cvCreateMat (3*m, count1, CV_32F);
pointMatn = cvCreateMat (3*m, count1, CV_32F);
epilinesCoef1 = cvCreateMat (3, count1, CV_32F); //, &B[0][0]);
epilinesCoef2 = cvCreateMat (3, count1, CV_32F);
eCoef2 = cvCreateMat (count1, 3, CV_32F);

// normalize the measurement
for(int i=0; i<m; i++)
{
    CopyMat (pointsMat1, 0, 0, pointMat, 3*i, 0, 3, count1);
    cvGEMM(invAguess, pointsMat1, 1., 0, 0., pointsMat1n, 0);
    CopyMat (pointMatn, 3*i, 0, pointsMat1n, 0, 0, 3, count1);
}
CopyMat (pointsMat1n, 0, 0, pointMat, 0, 0, 3, count1);
CopyMat (pointsMat2n, 0, 0, pointMat, 3*(m-1), 0, 3, count1);

// calculate fundamental matrix F from normalized coordinates
// decompose it
DFundamental (pointsMat1n, pointsMat2n, fund);

PrintMat (fund, 0, 0, 3, 3);

// compute epipolar lines
cvTranspose (fund, fundT);
cvComputeCorrespondEpilines (pointsMat1n, 1, fund, epilinesCoef2);
cvComputeCorrespondEpilines (pointsMat2n, 2, fund, epilinesCoef1);
for( int i = 0; i < count1; i++ )
{
    PaintEpline (epilinesCoef2, i, imageR);
    PaintEpline (epilinesCoef1, i, imageL);
}

CvMat *U, *W, *V, *ep1, *ep2, *ep_skew, *M1;
U = cvCreateMat (3, 3, CV_32F);
W = cvCreateMat (3, 3, CV_32F);
V = cvCreateMat (3, 3, CV_32F);
ep1 = cvCreateMat (3, 1, CV_32F);
ep2 = cvCreateMat (3, 1, CV_32F);
ep_skew = cvCreateMat (3, 3, CV_32F);
M1 = cvCreateMat (3, 3, CV_32F);
// cvTranspose (&fund, fundT);
CvMat *That = cvCreateMat (3, m, CV_32F);
CvMat *Rhat = cvCreateMat (3*m, 3, CV_32F);
cvZero (That); cvZero (Rhat);
cvmSet (Rhat, 0, 0, 1); cvmSet (Rhat, 1, 1, 1); cvmSet (Rhat, 2, 2, 1);

cvSVD (fundT, W, U, V);

cvmat_assign31 (ep2, cvmGet (V, 0, 2), cvmGet (V, 1, 2), cvmGet (V, 2, 2));
skew (ep2, ep_skew);
cvGEMM (ep_skew, fund, 1., 0, 0., M1, CV_GEMM_A_T);
cvCopy (M1, protmatrixR, 0);
cvCopy (ep2, ptransvectorR, 0);
CopyMat (That, 0, m-1, ep2, 0, 0, 3, 1);
CopyMat (Rhat, 3*(m-1), 0, M1, 0, 0, 3, 3);

Memol->Lines->Add ("rotation matrixs ");
PrintMat (Rhat, 0, 0, 3*m, 3);
Memol->Lines->Add ("transition vectors ");
PrintMat (That, 0, 0, 3, m);
CvMat *XP=cvCreateMat (4, count1, CV_32F);
CvMat *PP1=cvCreateMat (3, 4, CV_32F);
CvMat *PP2=cvCreateMat (3, 4, CV_32F);
CvMat *lambda=cvCreateMat (1, count1, CV_32F);
CvMat *alpha=cvCreateMat (1, count1, CV_32F);
cvZero (PP1); cvmSet (PP1, 0, 0, 1); cvmSet (PP1, 1, 1, 1); cvmSet (PP1, 2, 2, 1);
CopyMat (PP2, 0, 0, protmatrixR, 0, 0, 3, 3);
CopyMat (PP2, 0, 3, ptransvectorR, 0, 0, 3, 1);
Memol->Lines->Add ("PP1"); PrintMat (PP1, 0, 0, 3, 4);

```

```

Memol->Lines->Add("PP2");PrintMat(PP2,0,0,3,4);

// compute projective structure and save it
Compute3DStructure(pointsMat1n,pointsMat2n,PP1,PP2,XP,lambda);
CopyMat(XP2,0,0,XP,0,0,4,count1);
for (int i=0;i<count1;i++)
{
    Series1->AddXYZ(cvmGet(XP,0,i),cvmGet(XP,1,i),cvmGet(XP,2,i),i,c1Red );
    cvmSet(alpha,0,i,cvmGet(lambda,0,0)/cvmGet(lambda,0,i));
}

// compute initial motion for all images
CvMat *fs=cvCreateMat(1,30,CV_32F); cvmSet(fs,0,0,100);
float errabs =10, errrel = 10, errlambda=100, alp, scale, val, f;
int iter = 1,sum;

CvMat *Q=cvCreateMat(3*count1,12,CV_32F);
CvMat *Um = cvCreateMat(3*count1,3*count1,CV_32F);
CvMat *Wm = cvCreateMat(3*count1,12,CV_32F);
CvMat *Vm = cvCreateMat(12,12,CV_32F);
CvMat *a = cvCreateMat(3,2,CV_32F);
CvMat *ap1 = cvCreateMat(3,1,CV_32F); CvMat *ap2 = cvCreateMat(3,3,CV_32F);
CvMat *skew_xn = cvCreateMat(3,3,CV_32F);
CvMat *xnp = cvCreateMat(3,1,CV_32F);
CvMat *Tp = cvCreateMat(3,1,CV_32F); CvMat *Rp=cvCreateMat(3,3,CV_32F);
CvMat *P = cvCreateMat(3*m,4,CV_32F); CvMat *Ppl=cvCreateMat(3,4,CV_32F);
CvMat *tt = cvCreateMat(3,count1,CV_32F);
CvMat *xr = cvCreateMat(3,count1,CV_32F);
CvMat *xd = cvCreateMat(2,count1,CV_32F);
CvMat *xim = cvCreateMat(3,count1,CV_32F);
CvMat *res = cvCreateMat(1,m*count1,CV_32F);
CvMat *errnorm = cvCreateMat(1,count1,CV_32F);
CvMat *lambda_prev = cvCreateMat(1,count1,CV_32F);
//-----
while ((errabs > 1e-4) & (iter < 30) & (errrel > 1e-5)) {
for(int k=1;k<m;k++)
{
    cvZero(Q);
    for(int i=0;i<count1;i++)
    {
        cvmSet(Q,3*i+0,3,-cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+0,4,-cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+0,5,-cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+0,6,cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+0,7,cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+0,8,cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+0,10,-cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+2,i));
        cvmSet(Q,3*i+0,11,cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+1,i));
        cvmSet(Q,3*i+1,0,cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+1,1,cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+1,2,cvmGet(pointMatn,k*3+2,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+1,6,-cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+1,7,-cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+1,8,-cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+1,9,cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+2,i));
        cvmSet(Q,3*i+1,11,-cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+0,i));
        cvmSet(Q,3*i+2,0,-cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+2,1,-cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+2,2,-cvmGet(pointMatn,k*3+1,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+2,3,cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,0,i));
        cvmSet(Q,3*i+2,4,cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,1,i));
        cvmSet(Q,3*i+2,5,cvmGet(pointMatn,k*3+0,i)*cvmGet(pointMatn,2,i));
        cvmSet(Q,3*i+2,9,-cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+1,i));
        cvmSet(Q,3*i+2,10,cvmGet(alpha,0,i)*cvmGet(pointMatn,k*3+0,i));
    }

    cvSVD(Q,Wm,Um,Vm);
    cvmSet(That,0,k,cvmGet(Vm,9,11));
    cvmSet(That,1,k,cvmGet(Vm,10,11));cvmSet(That,2,k,cvmGet(Vm,11,11));
    cvmSet(Rhat,3*k+0,0,cvmGet(Vm,0,11));
    cvmSet(Rhat,3*k+0,1,cvmGet(Vm,1,11));
    cvmSet(Rhat,3*k+0,2,cvmGet(Vm,2,11));
}
}

```

```

cvmSet (Rhat, 3*k+1, 0, cvmGet (Vm, 3, 11));
cvmSet (Rhat, 3*k+1, 1, cvmGet (Vm, 4, 11));
cvmSet (Rhat, 3*k+1, 2, cvmGet (Vm, 5, 11));
cvmSet (Rhat, 3*k+2, 0, cvmGet (Vm, 6, 11));
cvmSet (Rhat, 3*k+2, 1, cvmGet (Vm, 7, 11));
cvmSet (Rhat, 3*k+2, 2, cvmGet (Vm, 8, 11));
}
CopyMat (lambda_prev, 0, 0, lambda, 0, 0, 1, count1);

// recompute alpha's based on all views
for(int i=0;i<count1;i++)
{
    for(int k=1;k<m;k++)
    {
        CopyMat (xnp, 0, 0, pointMatn, 3*k, i, 3, 1); CopyMat (Tp, 0, 0, That, 0, k, 3, 1);
        CopyMat (Rp, 0, 0, Rhat, 3*k, 0, 3, 3);
        skew (xnp, skew_xn); cvGEMM (skew_xn, Tp, 1., 0, 0., ap1, 0);
        CopyMat (a, 0, 0, ap1, 0, 0, 3, 1);
        cvGEMM (skew_xn, Rp, 1., 0, 0., ap2, 0); cvGEMM (ap2, xnp, 1., 0, 0., ap1, 0);
        CopyMat (a, 0, 1, ap1, 0, 0, 3, 1);
        CopyMat (M, 3*(k-1), 0, a, 0, 0, 3, 2);
    }
    alp = 0;
    for(int l=0;l<3*(m-1);l++){
        alp += -cvmGet (M, l, 0) *cvmGet (M, l, 1);
        CvMat *Mp = cvCreateMat (3*(m-1), 1, CV_32F);
        CopyMat (Mp, 0, 0, M, 0, 0, 3*(m-1), 1);
        alp = alp/ cvNorm (Mp, 0, CV_L2, 0) /cvNorm (Mp, 0, CV_L2, 0);
        cvmSet (alpha, 0, i, alp);
    }
    scale = cvmGet (alpha, 0, 0);
    cvConvertScale (alpha, alpha, 1/scale, 0);
    for(int i=0;i<count1;i++)
    {
        cvmSet (lambda, 0, i, 1/cvmGet (alpha, 0, i));
        cvmSet (XP, 0, i, cvmGet (lambda, 0, i) *cvmGet (pointMatn, 0, i));
        cvmSet (XP, 1, i, cvmGet (lambda, 0, i) *cvmGet (pointMatn, 1, i));
        cvmSet (XP, 2, i, cvmGet (lambda, 0, i) *cvmGet (pointMatn, 2, i));
        cvmSet (XP, 3, i, 1);
    }

    int i=0;
    for(int l=0;l<m;l++)
    {
        CopyMat (Pp1, 0, 0, Rhat, 3*l, 0, 3, 3);
        CopyMat (Tp, 0, 0, That, 0, l, 3, 1); cvConvertScale (Tp, Tp, scale, 0);
        CopyMat (Pp1, 0, 3, Tp, 0, 0, 3, 1);
        cvGEMM (Pp1, XP, 1., 0, 0., tt, 0);
        sum=0;
        for(int k=0;k<count1;k++) // counts sum(sign(..))
        {
            val = cvmGet (tt, 2, i);
            if(val<0) sum--;
            if(val>0) sum++;
        }
        if(sum < -count1/2)
        {
            cvConvertScale (Pp1, Pp1, -1, 0);
            CopyMat (Tp, 0, 0, That, 0, l, 3, 1); CopyMat (Rp, 0, 0, Rhat, 3*l, 0, 3, 3);
            cvConvertScale (Tp, Tp, -1, 0); cvConvertScale (Rp, Rp, -1, 0);
            CopyMat (That, 0, l, Tp, 0, 0, 3, 1); CopyMat (Rhat, 3*l, 0, Rp, 0, 0, 3, 3);
            cvGEMM (Pp1, XP, 1., 0, 0., tt, 0);
        }
        CopyMat (P, 3*l, 0, Pp1, 0, 0, 3, 4);
        for(int k=0;k<count1;k++) // projective trans.
        {
            cvmSet (tt, 0, k, cvmGet (tt, 0, k) /cvmGet (tt, 2, k));
            cvmSet (tt, 1, k, cvmGet (tt, 1, k) /cvmGet (tt, 2, k));
            cvmSet (tt, 2, k, 1);
        }
        cvGEMM (Aguess, tt, 1., 0, 0., xr, 0);
        //CopyMat (xim, 0, 0, pointMat, 3*l, 0, 3, count1);
        for(int k=0;k<count1;k++) // projective trans.

```

```

        {
            cvmSet(xd,0,k,cvmGet(pointMat,3*1+0,k)-cvmGet(xr,0,k));
            cvmSet(xd,1,k,cvmGet(pointMat,3*1+1,k)-cvmGet(xr,1,k));
            cvmSet(errnorm,0,k,cvSqrt(cvmGet(xd,0,k)*cvmGet(xd,0,k)+
            cvmGet(xd,1,k)*cvmGet(xd,1,k)));
        }
        CopyMat(res,0,1*count1,errnorm,0,0,1,count1);
        i = i+1;
    }
    f=0;
    for(int i=0;i<(count1*m);i++)
        f += cvmGet(res,0,i);
        f = f/count1/m;
        iter = iter + 1;
        cvmSet(fs,0,iter-1,f);
        if (iter>1)
        {
            errrel = fabs(cvmGet(fs,0,iter-2)-f);
            if (cvmGet(fs,0,iter-2)<f)
            {
                errrel = 1e-10;
            }
        }
        errabs = fabs(f);
        errlambda = cvNorm(lambda_prev,lambda,CV_L2,0);
        CopyMat(lambda_prev,0,0,lambda,0,0,0,count1);
    } // end of while

//-----
Memol->Lines->Add("P1 ");
PrintMat(P,0,0,3,4);
Memol->Lines->Add("P2 ");
PrintMat(P,3,0,3,4);

CopyMat(XP3,0,0,XP,0,0,4,count1);

// automatic calibration, compute unknown focal length
// set up absolute quadric constrains
CvMat *Omeg = cvCreateMat(4,4,CV_32F);

linear_quadric(P,Omeg);
if(cvmGet(Omeg,0,0)< 0)
cvConvertScale(Omeg,Omeg,-1,0);
Memol->Lines->Add("omega ");
PrintMat(Omeg,0,0,4,4);

CvMat *t= cvCreateMat(3,3,CV_32F);
CvMat *p = cvCreateMat(3,4,CV_32F);
cvGEMM(PP1,Omeg,1.,0,0.,p,0);
cvGEMM(p,PP1,1.,0,0.,t,CV_GEMM_B_T);
Memol->Lines->Add("t ");
PrintMat(t,0,0,3,3);
float fest = cvSqrt(fabs(cvmGet(t,0,0)/cvmGet(t,2,2)));
float fest2 = cvSqrt(fabs(cvmGet(t,1,1)/cvmGet(t,2,2)));

CvMat *v= cvCreateMat(3,1,CV_32F);
CvMat *K1= cvCreateMat(3,3,CV_32F);

    cvmat_assign31(v,-cvmGet(Omeg,0,3)/(fest*fest),
                -cvmGet(Omeg,1,3)/(fest2*fest2),
                -cvmGet(Omeg,2,3));
    cvmat_assign33(K1,fest,0,0,0,fest2,0,0,0,1);

CvMat *vp= cvCreateMat(1,3,CV_32F);
CvMat *Hp= cvCreateMat(4,4,CV_32F);
cvGEMM(v,K1,-1.,0,0.,vp,CV_GEMM_A_T);
cvSetZero(Hp);
CopyMat(Hp,0,0,K1,0,0,3,3);
CopyMat(Hp,3,0,vp,0,0,1,3);
cvmSet(Hp,3,3,1);
// cvmSet(Hp,0,0,fest/imageL->width);
// cvmSet(Hp,1,1,fest/imageL->height);

```

```

Memol->Lines->Add("Hp ");
PrintMat (Hp, 0, 0, 4, 4);

CvMat *Xe=cvCreateMat(4, count1, CV_32F);
CvMat *invHp= cvCreateMat(4, 4, CV_32F);

// estimate the projective to Euclidean upgrade
cvInvert (Hp, invHp, CV_LU);
cvGEMM(invHp, XP, 1., 0, 0., Xe, 0);
Memol->Lines->Add("inverse Hp ");
PrintMat (invHp, 0, 0, 4, 4);

Series1->Clear();
int sc=1; sum=0;
for(int i=0; i<count1; i++)
{
    cvmSet (Xe, 0, i, cvmGet (Xe, 0, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 1, i, cvmGet (Xe, 1, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 2, i, cvmGet (Xe, 2, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 3, i, 1);
    if (cvmGet (Xe, 2, i) < 0) sum--;
    if (cvmGet (Xe, 2, i) > 0) sum++;
}
if (sum < 0 & sum == -count1)
{
    cvmSet (Hp, 3, 3, -1);
    cvInvert (Hp, invHp, CV_LU);
    cvGEMM(invHp, XP, 1., 0, 0., Xe, 0);
    Memol->Lines->Add("Warning: change the sign of Hp ");
}
CopyMat (iHp, 0, 0, invHp, 0, 0, 4, 4);
for(int i=0; i<count1; i++)
{
    cvmSet (Xe, 0, i, cvmGet (Xe, 0, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 1, i, cvmGet (Xe, 1, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 2, i, cvmGet (Xe, 2, i) /cvmGet (Xe, 3, i));
    cvmSet (Xe, 3, i, 1);
}

Memol->Lines->Add("Perspective projection ");
PrintMat (P, 0, 0, 3*m, 4);
CvMat *PP=cvCreateMat(3*m, 4, CV_32F);
cvGEMM(P, Hp, 1., 0, 0., PP, 0);
Memol->Lines->Add("Euclidean projection ");
PrintMat (PP, 0, 0, 3*m, 4);

CvMat *Rep=cvCreateMat(3, 3, CV_32F);
CvMat *Arem=cvCreateMat(3, 3, CV_32F);
CvMat *invArem=cvCreateMat(3, 3, CV_32F);
CvMat *Tep=cvCreateMat(3, 1, CV_32F);
CvMat *Re=cvCreateMat(3*m, 3, CV_32F);
CvMat *Arem=cvCreateMat(3*m, 3, CV_32F);
CvMat *X=cvCreateMat(4*m, count1, CV_32F);
CvMat *X1=cvCreateMat(4, count1, CV_32F);
CvMat *p1=cvCreateMat(4, 4, CV_32F);
CvMat *Te=cvCreateMat(3, m, CV_32F);
cvZero(p1); cvmSet(p1, 3, 3, 1);

// rigid body motion and 3D Euclidean recovery
for(int i=0; i<m; i++)
{
    CopyMat (Rep, 0, 0, PP, 3*i, 0, 3, 3);
    rq(Rep, Arem, Rep);
    cvInvert (Arem, invArem);
    CvMat *ppt=cvCreateMat(3, 1, CV_32F);
    CopyMat (ppt, 0, 0, PP, 3*i, 3, 3, 1);
    cvGEMM(invArem, ppt, 1., 0, 0., Tep, 0);
    CopyMat (p1, 0, 0, Rep, 0, 0, 3, 3); CopyMat (p1, 0, 3, Tep, 0, 0, 3, 1);
    cvGEMM(p1, Xe, 1., 0, 0., X1, 0);
    CopyMat (X, 4*i, 0, X1, 0, 0, 4, count1);
    CopyMat (Re, 3*i, 0, Rep, 0, 0, 3, 3);
    CopyMat (Te, 0, i, Tep, 0, 0, 3, 1);
}

```



```

    Mem01->Lines->Add("Euclidean projection "+CurrToStr(i));
    PrintMat (p1,0,0,4,4);
}

// print euclidean structure
for(int i=0;i<count1;i++)
{
    if(fabs(cvmGet(X,0,i))<100 & fabs(cvmGet(X,1,i))<100 &
        fabs(cvmGet(X,2,i))<100)
        Series1->AddXYZ(sc*cvmGet(X,0,i),sc*cvmGet(X,1,i),
            sc*cvmGet(X,2,i),i,clRed );
}
Chart1->Title->Text->SetText("Euclidean Reconstruction");
CopyMat (XE,0,0,X,0,0,4,count1);
CopyMat (PeRight,0,3,Te,0,m-1,3,1);

CopyMat (PeLeft,0,0,PP,0,0,3,4);
CopyMat (PeRight,0,0,PP,3*(m-1),0,3,4);
// CopyMat (pointsMat1,0,0,pointMat,0,0,3,count1);
// CopyMat (pointsMat2,0,0,pointMat,3*(m-1),0,3,count1);
CopyMat (Fund,0,0,fund,0,0,3,3);

// release used matrixes
cvReleaseMat (&Aguess); cvReleaseMat (&invAguess); cvReleaseMat (&pointsMat1);
    CvReleaseMat (&pointsMat1n);cvReleaseMat (&epilinesCoef1);
cvReleaseMat (&pointMat);cvReleaseMat (&pointMatn);
cvReleaseMat (&pointsMat2);cvReleaseMat (&pointsMat2n);
cvReleaseMat (&epilinesCoef2);cvReleaseMat (&eCoef2);cvReleaseMat (&fund);
cvReleaseMat (&fundT);cvReleaseMat (&U);cvReleaseMat (&W);cvReleaseMat (&V);
cvReleaseMat (&ep1);cvReleaseMat (&ep2); CvReleaseMat (&ep_skew);
cvReleaseMat (&M1);cvReleaseMat (&That);cvReleaseMat (&Rhat);
cvReleaseMat (&XP);cvReleaseMat (&PP1);cvReleaseMat (&PP2);
cvReleaseMat (&lambda);cvReleaseMat (&alpha); cvReleaseMat (&fs);
cvReleaseMat (&Q);cvReleaseMat (&Um);cvReleaseMat (&Wm);cvReleaseMat (&Vm);
cvReleaseMat (&M); CvReleaseMat (&a);cvReleaseMat (&skew_xn);cvReleaseMat (&Tp);
cvReleaseMat (&P);cvReleaseMat (&tt);cvReleaseMat (&xr);cvReleaseMat (&xd);
cvReleaseMat (&xim);cvReleaseMat (&res); cvReleaseMat (&errnorm);
cvReleaseMat (&lambda_prev);cvReleaseMat (&Omeg);cvReleaseMat (&t);
cvReleaseMat (&p);cvReleaseMat (&v);cvReleaseMat (&K1);cvReleaseMat (&vp);
cvReleaseMat (&Hp);cvReleaseMat (&Xe);cvReleaseMat (&invHp);cvReleaseMat (&PP);
cvReleaseMat (&Rep);cvReleaseMat (&Arem);cvReleaseMat (&invArem);
cvReleaseMat (&Tep);cvReleaseMat (&Re);cvReleaseMat (&Arem);
cvReleaseMat (&X);cvReleaseMat (&X1);cvReleaseMat (&p1);
}

```