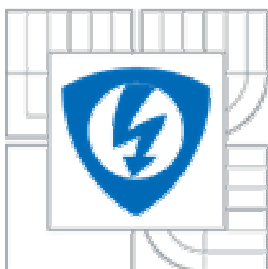




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘENÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF AUTOMATION AND MEASUREMENT

TVORBA 3D MODELŮ 3D RECONSTRUCTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN MUSÁLEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MILOSLAV RICHTER, PHD.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Martin Musálek

ID: 125554

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Tvorba 3D modelů

POKYNY PRO VYPRACOVÁNÍ:

Úkolem je napsat aplikaci, která vytvoří prostorový model reálného objektu na základě jeho snímků pořízených kamerou za pomoci nasvícení objektu definovaným vzorem pomocí projektoru.

- 1) Nastudujte metody vytvoření (360 stupňů) 3D modelu objektu za pomoci nasvětlení vzorem.
- 2) Navrhněte měřicí stanoviště a zvolte vhodný vzor pro nasvícení.
- 3) Realizujte metody pro kalibraci systému.
- 4) Realizujte metody pro 3D rekonstrukci objektu. Pro uložení zvolte vhodný datový formát.
- 5) Vhodným způsobem vizualizujte výsledek rekonstrukce a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 1998, ISBN 80-251-0454-0

Hlaváč V., Šonka M.: Počítačové vidění, Grada, Praha 1992, ISBN 80-85424-67-3

Faugeras O.: Three-Dimensional Computer Vision, The MIT Press 1993

Kraus K.: Photogrammetrie 1 und 2, Ummeler / Bonn, 1996

Termín zadání: 10.2.2014

Termín odevzdání: 19.5.2014

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce řeší 3D rekonstrukci objektu pomocí metody nasvícení vzorem. Projektor nasvěcuje měřený objekt definovaným vzorem a dvojice kamer z něj snímá body. Podstavec s objektem se otáčí, a během více měření je objekt sejmuto z více úhlů. Body jsou identifikovány z naměřených snímků, transformovány na 3D pomocí stereovidění, spojeny do 3D modelu a zobrazeny.

KLÍČOVÁ SLOVA

3D rekonstrukce, stereovidění, počítačové vidění

ABSTRACT

Thesis solves 3D reconstruction of an object by method of lighting by pattern. A projector lights the measured object by defined pattern and two cameras are measuring 2D points from it. The pedestal of object rotates and during the measure are acquired data from different angles. Points are identified from measured images, transformed to 3D using stereovision, connected to 3D model and displayed.

KEYWORDS

3D reconstruction, stereovision, computer vision

MUSÁLEK, M. *Tvorba 3D modelů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měření, 2014. 72 s., 0 s. příloh. Diplomová práce. Vedoucí práce: ing. Miloslav Richter, PhD.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Tvorba 3D modelů jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce ing. Miloslavu Richterovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ	5
1 ÚVOD	7
1.1 Popis pracoviště.....	7
1.2 Cíle práce.....	8
2 UŽÍVANÉ METODY	9
2.1 Triangulace.....	9
2.1.1 Aktivní triangulace.....	9
2.1.2 Pasivní triangulace.....	9
2.1.3 Metoda zaostřování.....	10
2.1.4 Tvar ze stínování.....	11
2.1.5 Využití teodolitu.....	11
2.1.6 Doba letu modulovaného paprsku.....	11
2.1.7 Optická interferometrie.....	12
2.2 Metody s nasvícovaným vzorem.....	12
3 VÝBĚR METODY	15
3.1 Mřížka.....	15
3.2 Diferenční metody.....	16
3.3 Šachovnice.....	17
3.4 Vzájemná identifikace bodů.....	18
3.4.1 Postupné měření.....	18
3.4.2 Značka.....	18
3.4.3 Postupná změna rozlišení vzoru.....	19
3.4.4 Grayův kód.....	19
3.5 Bílý šum.....	20
3.6 Vzor s pevným kódem.....	20
3.7 Proužky.....	21
3.8 Další možné metody.....	21

3.8.1	Trojúhelníky a šestiúhelníky	21
3.8.2	Kruhy.....	21
3.8.3	Použití barevných kamer	22
3.9	Nasvícování pozadí.....	22
3.10	Aktivní metody.....	22
3.11	Zvolená metoda	23
4	MATEMATICKÝ ZÁKLAD METODY	24
4.1	Prostorové transformace.....	24
4.1.1	Homogenní souřadnice.....	24
4.1.2	Posunutí.....	24
4.1.3	Rotace.....	24
4.1.4	Měřítko	25
4.1.5	Projekce 3D - 2D.....	25
4.1.6	Skládání transformací.....	25
4.2	Obrazové transformace	25
4.2.1	Prahování.....	25
4.2.2	Konvoluce	26
5	KALIBRACE SYSTÉMU	28
5.1	Změření zorného pole kamer.....	28
5.2	Kalibrační tabule	29
5.3	Rozpoznání bodů.....	30
5.4	Identifikace bodů kalibrační tabule	33
5.5	Výpočet kalibrace.....	35
5.6	Zpřesnění kalibrace	38
5.6.1	Průměrování	38
5.6.2	Vážené průměrování	38
5.6.3	Metoda nejmenších čtverců.....	39
6	NASNÍMÁNÍ DAT	40
6.1	Příprava pracoviště.....	40
6.1.1	Značky na podstavci.....	40

6.2	Program pro měření.....	40
6.2.1	Ukládání obrazu z kamer.....	41
6.2.2	Nastavení světlosti vzoru	41
6.2.3	Automatické měření	42
7	KNIHOVNY PRO 3D GEOMETRII	44
7.1	Vzdálenost dvou bodů.....	44
7.2	Velikost vektoru	44
7.3	Jednotkový vektor	44
7.4	Vektorový součin	45
7.5	Nejbližší bod přímky ke přímce	45
7.6	Průsečík dvou přímek.....	46
7.7	Průsečík přímky a úsečky.....	46
7.8	Obecná rovnice roviny	47
7.9	Průsečnice dvou rovin	47
7.10	Bod na rovině nejbližší k danému bodu	47
7.11	Test bodu uvnitř trojúhelníku	48
8	REKONSTRUKCE OBJEKTU	50
8.1	Zvolený datový formát	50
8.2	Rozpoznání bodů.....	50
8.3	Vzájemná identifikace bodů.....	54
8.4	Transformace bodů.....	56
8.4.1	Transformace v rámci jednoho úhlu.....	56
8.4.2	Zjištění osy natočení.....	56
8.4.3	Otočení bodu podle osy.....	56
8.4.4	Problém posunutí osy objektu	57
8.5	Zobrazení 3D modelu.....	58
8.6	Algoritmus spojování bodů	58
8.7	Metody spojení 3D modelu	60
8.7.1	Výpočet středu objektu.....	60
8.7.2	Metoda polárních souřadnic	60

8.7.3	Metoda projekce na roviny.....	61
8.7.4	Spojení podle 3D vzdálenosti.....	62
8.7.5	Přitažení dvou dílčích modelů.....	63
8.7.6	Řez dvou dílčích modelů.....	63
8.8	Další úkoly pro tvorbu modelů.....	64
9	ZÁVĚR	65
	LITERATURA	66
	SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK	67

SEZNAM OBRÁZKŮ

Obr. 1 Použitý projektor a kamery	7
Obr. 2 Podstavec a testovací objekt	7
Obr. 3 Princip stereovidění (převzato ze zdroje [2]).....	10
Obr. 4 Snímky metody depth from focus (převzato ze zdroje [6])	10
Obr. 5 Výsledek metody depth from focus (převzato ze zdroje [6]).....	11
Obr. 6 Princip měření doby letu	12
Obr. 7 Uspořádání Michelsonova interferometru (převzato ze zdroje [5]).....	12
Obr. 8 Uspořádání kamer při měření s nasvíceným vzorem (převzato ze zdroje [3])	13
Obr. 9 Ztráta informace při měření nasvíceným vzorem	13
Obr. 10 Mřížka	15
Obr. 11 Ztráta informace odleskem.....	16
Obr. 12 Invertovaná mřížka.....	16
Obr. 13 Objekt nasvícený invertovanou mřížkou	17
Obr. 14 Šachovnice	17
Obr. 15 Varianty značek na šachovnici.....	18
Obr. 16 Řádková varianta mřížky	18
Obr. 17 První tři kroky měření s postupnou změnou rozlišení	19
Obr. 18 Sekvence Grayova kódu (převzato ze zdroje [1]).....	20
Obr. 19 Bílý šum (převzato ze zdroje [8]).....	20
Obr. 20 Vzor s pevným kódem	21
Obr. 21 Kruhy	22
Obr. 22 První tři kroky měření s postupnou změnou rozlišení pro RGB kamery.....	22
Obr. 23 Měření zorného pole kamery	28
Obr. 24 Výpočet zorného pole kamery	29
Obr. 25 Kalibrační tabule	29
Obr. 26 Chybné vyhodnocení bodů konvoluční masky	31
Obr. 27 Chyba vyhodnocení natočených bodů konvoluční maskou	32
Obr. 28 Srovnání rozpoznávaných bodů kalibrační tabule	33

Obr. 29 Výsledek rozpoznání bodů	33
Obr. 30 Postup při identifikaci bodů	35
Obr. 31 Snímky kalibrační tabule z obou kamer.....	37
Obr. 32 Vizualizace výpočtu kalibrace	37
Obr. 33 Chyba kalibrace.....	38
Obr. 34 Rozhraní měřicího programu	41
Obr. 35 Ztráta dat přesvědčením vzoru.....	42
Obr. 36 Chyba vyhodnocení bodu v trojúhelníku	49
Obr. 37 Princip vyhodnocení bodu v trojúhelníku	49
Obr. 38 Snímek nasvíceného a nenasvíceného objektu	51
Obr. 39 Snímky objektu nasvíceného šachovnicí a difference nasvícení.....	51
Obr. 40 Obraz po aplikaci prahování	52
Obr. 41 Obraz po aplikaci hranového detektoru	52
Obr. 42 Rozpoznané body objektu	53
Obr. 43 Kód pro identifikaci bodů	54
Obr. 44 První krok identifikace bodů	54
Obr. 45 Druhý krok identifikace bodů	55
Obr. 46 Terčíky ke snímání osy otáčení.....	56
Obr. 47 Terčíky k vyhodnocení posunutí osy objektu	57
Obr. 48 Posunutí osy objektu	57
Obr. 49 Program pro vizualizaci dat	58
Obr. 50 3D model jednoho úhlu měření.....	59
Obr. 51 Překrytí měření dvou různých úhlů.....	60
Obr. 52 Celkový 3D model objektu	61
Obr. 53 Princip metody projekce na roviny	62
Obr. 54 Výsledek metody přitažení modelů.....	63
Obr. 55 Důkaz nefukčnosti metody řezu modelů.....	64

1 ÚVOD

Cílem práce je vytvořit 3D model objektu, který je snímán dvěma kamerami a natáčen na podstavci. Objekt je nasvícován projektorem. Z projektoru musí být na objekt promítán takový vzor, ze kterého lze rozpoznat skutečné body na objektu.

1.1 Popis pracoviště

K dispozici byly dvě kamery ImagingSource DMK 21AF04 na stativu Manfrotto. Jde o šedotónové kamery s rozlišením 640·480, které se připojují přes rozhraní FireWire. Celý tento přípravek (viz. Obr. 1 vpravo) měl vodováhy pro vyrovnání roviny a nastavitelné úhly kamer ve všech osách. Projektor byl typu Epson TW-600 s rozlišením 800·600.



Obr. 1 Použitý projektor a kamery



Obr. 2 Podstavec a testovací objekt

1.2 Cíle práce

- 1) Nastudujte metody vytvoření (360 stupňů) 3D modelu objektu za pomoci nasvětlení vzorem.
- 2) Navrhněte měřicí stanoviště a zvolte vhodný vzor pro nasvícení.
- 3) Realizujte metody pro kalibraci systému.
- 4) Realizujte metody pro 3D rekonstrukci objektu. Pro uložení zvolte vhodný datový formát.
- 5) Vhodným způsobem vizualizujte výsledek rekonstrukce a zhodnoťte dosažené výsledky.

2 UŽÍVANÉ METODY

V této kapitole budou popsány běžně používané metody pro 3D měření objektů.

2.1 Triangulace

K výpočtu souřadnic 3D bodu se využívá vlastností trojúhelníků, a to převážně goniometrických funkcí a vět. V současnosti jde o nejpoužívanější metody optického měření. Triangulace se vyskytuje v následujících variantách.

2.1.1 Aktivní triangulace

Snímaná scéna je nasvícena. Vrcholy triangulačního trojúhelníka jsou zdroj světla, nasvícený bod, a detektor. Triangulační báze zdroj - snímač je neměnná. Mění se úhel, pod kterým je na snímači detekován vysvícený obrazec. Ten může mít podobu proužků, pseudonáhodného šumu nebo využívat techniku moiré. Jako příklad aktivní triangulace bude uvedeno měření s nasvíceným vzorem a jednou kamerou, které bude podrobně rozebráno níže.

2.1.2 Pasivní triangulace

U této metody není uvážena poloha zdroje osvětlení. Jsou zapotřebí dva snímky z různého úhlu pohledu. U statických scén stačí jedna kamera s proměnnou polohou. U rychle se měnících dějů jsou potřeba minimálně dvě kamery. Vzájemná poloha kamer musí být známa dopředu nebo zjištěna kalibrací. Speciálním případem pasivní triangulace je stereovidění. To bude popsáno jako příklad těchto metod. Dvě kamery musí ležet v jedné rovině a mít známou vzdálenost optických středů.

Paralaxa je úhlová změna pozice. Počítá se:

$$p = x' - x'' \quad (1)$$

, kde x' a x'' jsou průměty bodu na první a druhou kameru

Souřadnice se počítají:

$$X = x' (b_x / p) \quad Y = y' (b_x / p) \quad Z = f (b_x / p) \quad (2)$$

, kde y' je průmět bodu na první detektor, b_x je vzdálenost kamer, f je zobrazovací konstanta. Všechno je vyznačeno na Obr. 3.

Na Obr. 4 je příklad dvou různých zaostření scény. Výstupní mapa vzdáleností může vypadat viz. Obr. 5.



Obr. 5 Výsledek metody depth from focus (převzato ze zdroje [6])

2.1.4 Tvar ze stínování

Při těchto metodách je známá poloha kamery a osvětlení. Z obrazu měřeného objektu se určí vektory kolmé k jeho povrchu a z nich 3D informace. Také se dá použít více snímků při různých polohách osvětlení.

2.1.5 Využití teodolitu

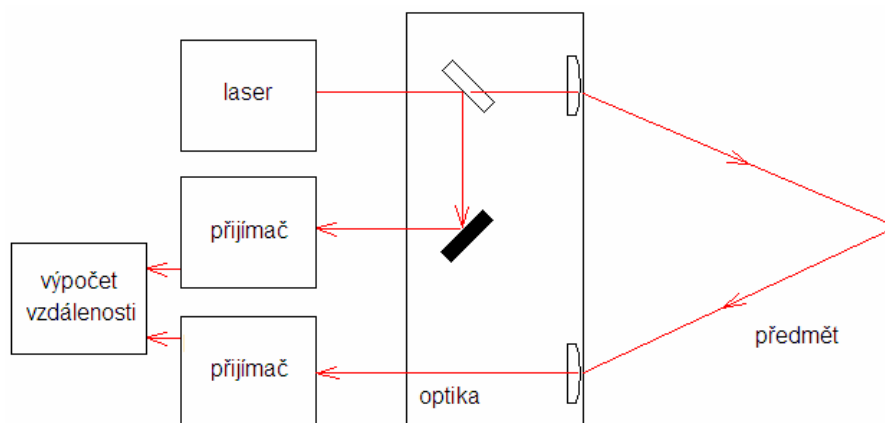
V současnosti je to nepřesnější triangulační systém pro měření rozlehlých objektů. Předmět je zaměřen aspoň dvěma teodolity. Ze získaných vertikálních a horizontálních úhlů a polohy přístrojů je vypočtena 3D informace. Měření je velmi časově náročné.

2.1.6 Doba letu modulovaného paprsku

Na měřený předmět se vysílá modulovaný paprsek světla. Je spuštěn časovač. Když je odražený paprsek detekován snímačem, časovač se zastaví. Změřená doba je úměrná vzdálenosti předmětu.

Touto metodou lze změřit jeden bod. Když chceme změřit více bodů, musí být použit čárový scanner. Jeho princip je bodové měření pro různé úhly, které jsou postupně nastavovány krokovým motorem. Pro každé bodové měření je znám úhel. Tak se dá změřit 2D profil. Obdobně se dá měřit i 3D.

Pro modulaci nosného světelného paprsku se užívá pulsů, spojitých funkcí nebo pseudonáhodného šumu. Přesnost této metody je relativně nízká - v řádu centimetrů. Používá se pro vzdálenosti v řádu desítek metrů.

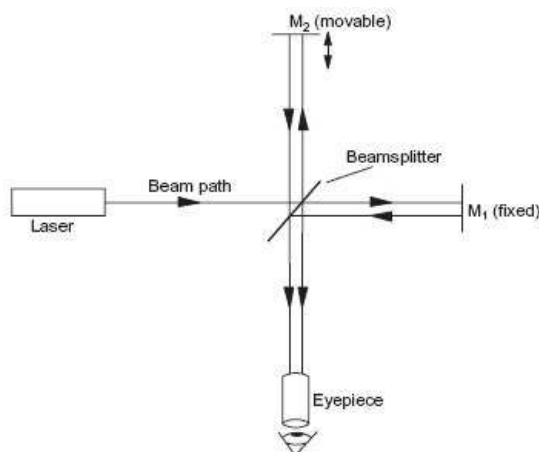


Obr. 6 Princip měření doby letu

2.1.7 Optická interferometrie

Metoda je založena na stejných principech jako měření doby letu, ale paprsek není modulován. Je rozdělen na referenční a měřicí svazek. Referenční směřuje do snímače. Měřicí směřuje do snímače až po odrazu od měřeného povrchu. To způsobuje fázový posun paprsků. Ve snímači dochází k jejich interferenci. Z intenzity výsledného stojatého vlnění lze určit fázový posun signálů a tím i časový posuv.

Nevýhoda této metody je, že umožňuje měřit jen nerovnosti povrchu do velikosti poloviny vlnové délky použitého světla, protože časový posuv není jednoznačný údaj. Výhoda je velká přesnost, která může být až v řádu nanometrů. Na Obr. 7 převzatého ze zdroje [5] je uspořádání Michelsonova interferometru. Zrcadlo je polopropustné. Vlevo je zdroj světla; vpravo je referenční vzdálenost; nahoře měřená vzdálenost; a dole detektor.

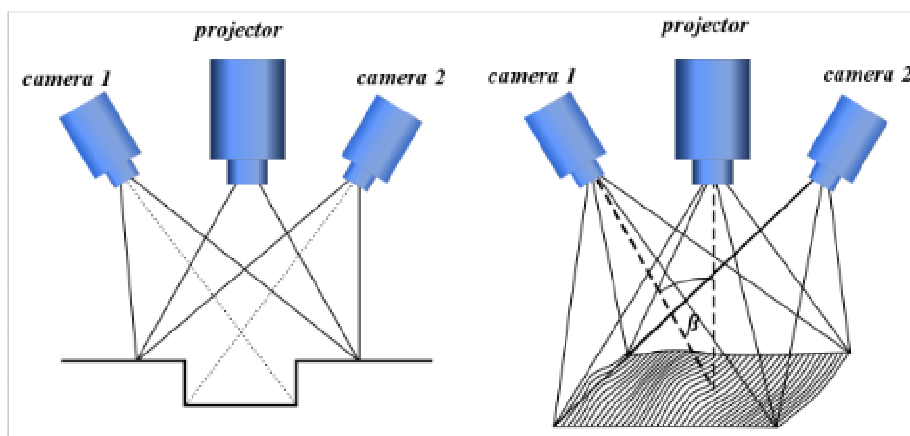


Obr. 7 Uspořádání Michelsonova interferometru (převzato ze zdroje [5])

2.2 Metody s nasvícovaným vzorem

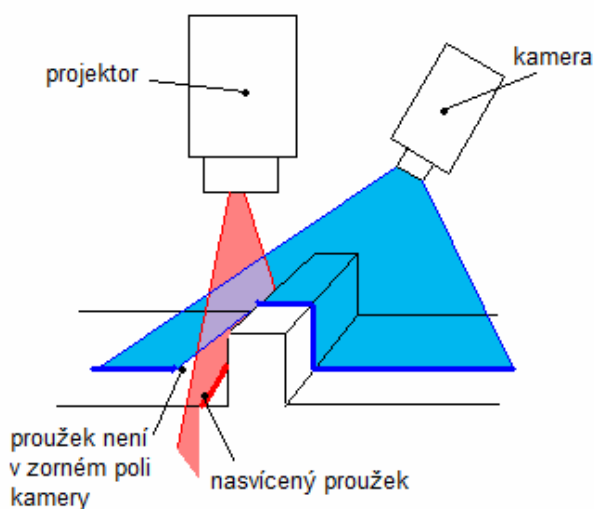
Princip těchto metod je promítání vzoru na povrch. Nejjednodušší variantou vzoru je

jeden proužek. Z různých perspektiv se na povrchu jeví zkresleně oproti promítanému vzoru, a toho se využívá při výpočtu 3D souřadnic. Metoda může být vylepšena promítáním více proužků najednou. Tato varianta se často používá v praxi. Na Obr. 8 je vidět obvyklé uspořádání kamer a projektoru při měření.



Obr. 8 Uspořádání kamer při měření s nasvíceným vzorem (převzato ze zdroje [3])

Měření lze provést i s jednou kamerou, ale při použití dvou kamer není při hodně členitém povrchu tak velká ztráta informace. Problém je ilustrován na Obr. 9.



Obr. 9 Ztráta informace při měření nasvíceným vzorem

Promítání proužku se provádí dvěma metodami - laserovou interferencí; nebo projekční metodou (zdroj [3]).

Metoda laserové interference používá dva lasery. Jejich interference vytvoří na objektu pravidelné pruhy, které budou mít od sebe stejnou vzdálenost. Velikost vzoru lze měnit změnou úhlu mezi světelnými svazky. Metoda umožňuje přesně a jednoduše generovat nasvícovaný vzor. Nevýhody jsou vysoká cena, problémy s geometrií svazků, a nežádoucí efekty typické pro laser, jako například interference s částmi svazku odražených od předmětů. Dále nelze individuálně nastavovat nasvícovaný vzor (např. Grayovým kódem).

Naproti tomu **projekční metoda** používá nekoherentní světlo z projektoru. Vzory jsou promítány projektorem - typicky LCD nebo LCOS. Také se používají projektory na principu DLP (zdroj [4]). Obraz je tvořen pomocí mikroskopických zrcadel umístěných na polovodičovém čipu DMD (Digital Micromirror Device). Každé zrcadlo reprezentuje jeden nebo více pixelů promítaného obrazu. Každá odrazná ploška se může přemísťovat mezi dvěma krajními polohami – jedna světlo propouští a druhá odráží. Intenzita barvy je regulována rychlým střídáním těchto poloh.

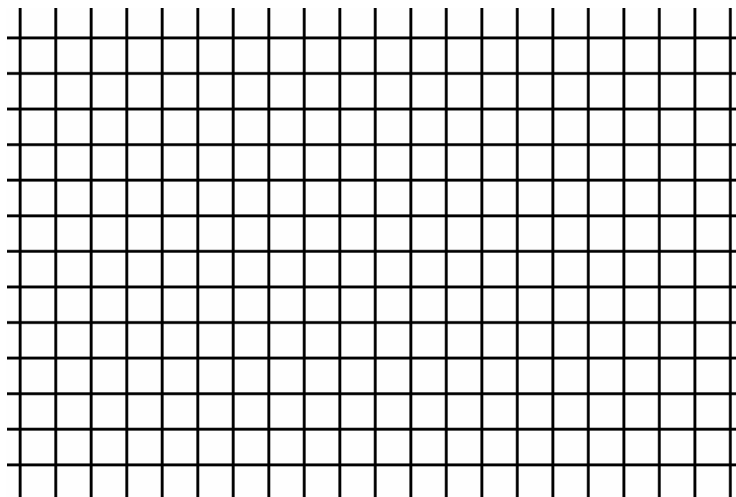
V některých aplikacích se používá IR světlo nebo přepínání dvou opačných vzorů.

3 VÝBĚR METODY

V kapitole 2 již bylo napsáno, že se používají pasivní a aktivní metody. V této práci byla použita pasivní metoda. Proto budou aktivní metody popsány jen stručně a kapitola bude zaměřena na metody pasivní. Budou zde uvažovány různé promítané vzory a vyjmenovány jejich výhody a nevýhody.

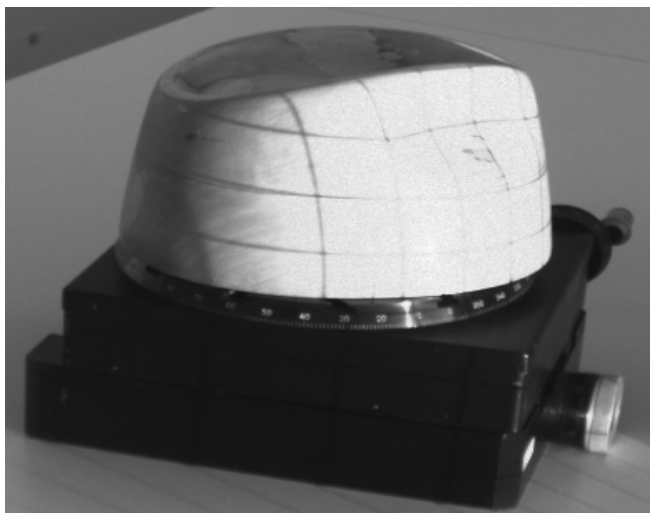
3.1 Mřížka

Na objekt je promítnut vzor mřížky. Body jsou detekovány na průsečících vodorovných a svislých čar na obrazech obou kamer na základě hranových detektorů.



Obr. 10 Mřížka

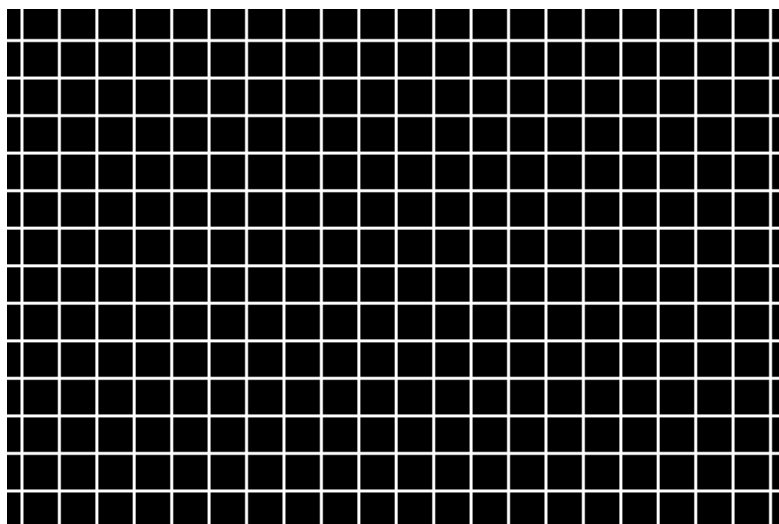
Často dochází ke ztrátě informace jak při měření, tak při detekci. Mřížka je totiž vzor, který je nejvíc náchylný na vnější vlivy. Největší ztráty způsobují stíny a odlesky viz. Obr. 11. Při detekci se pak body ztrácejí kvůli možné malé diferencí nasvícených a nenasvícených míst.



Obr. 11 Ztráta informace odleskem

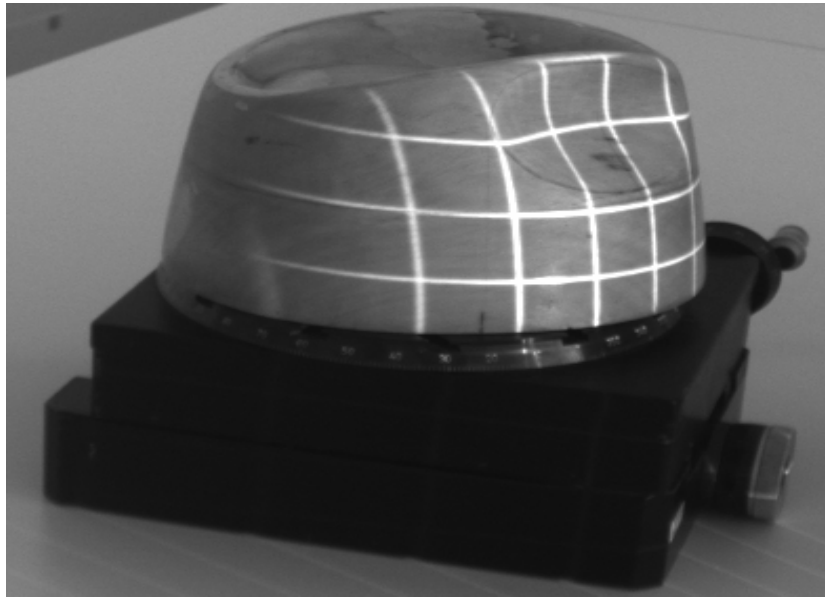
3.2 Diferenční metody

Problém výše uvedené metody je, že zjistí jen body na objektu, ale nezjistí jeho přesný tvar. Dále má nízkou odolnost vůči vnějším vlivům. Oba tyto problémy částečně řeší následující modifikace. Jsou pořizovány čtyři druhy snímků - jeden s mřížkou; druhý s mřížkou s invertovanými barvami; třetí nasvícený bíle a čtvrtý nenasvícený. Pro detekci tvaru stačí snímek nasvíceného a nenasvíceného objektu. Při použití libovolných tří z těchto snímků navíc může být z velké části eliminována **textura objektu**.



Obr. 12 Invertovaná mřížka

Zpracování obrazů je založeno na jejich odečítání od sebe a zjištění difference. Ta je dále prahována a je získán černobílý obraz přesného tvaru objektu (viz. kapitola 8.1).

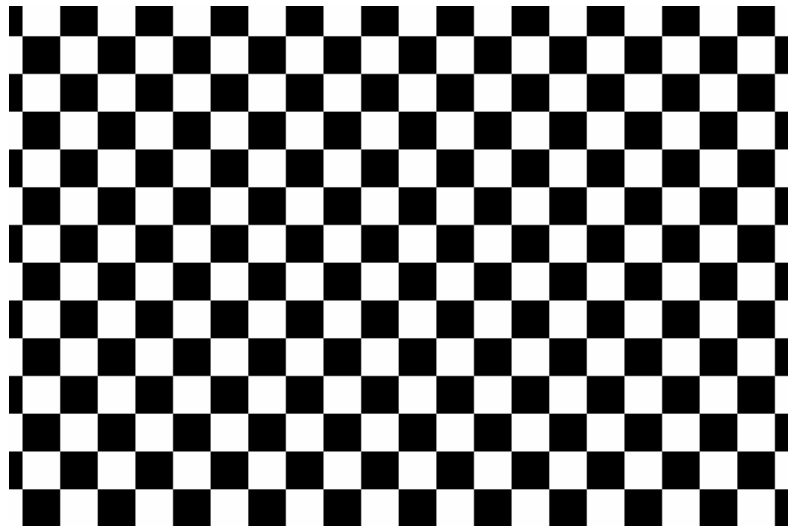


Obr. 13 Objekt nasvícený invertovanou mřížkou

Obrys objektu může být získán morfologickými operacemi s prahovaným snímkem difference (viz. Obr. 40).

3.3 Šachovnice

Lepších výsledků lze dosáhnout při použití šachovnice místo mřížky. Metoda difference se na ni dá uplatnit stejně jako na mřížku a kterýkoli jiný vzor.



Obr. 14 Šachovnice

Oproti mřížce je zde vyšší odolnost vůči stínům a odleskům. Další výhodou je možnost rozpoznání bodů na jiném principu. Je použita šachovnicová konvoluční maska, a ve výsledku konvoluce jsou nalezeny extrémy. Postup je stejný jako v kapitole 5.3.

3.4 Vzájemná identifikace bodů

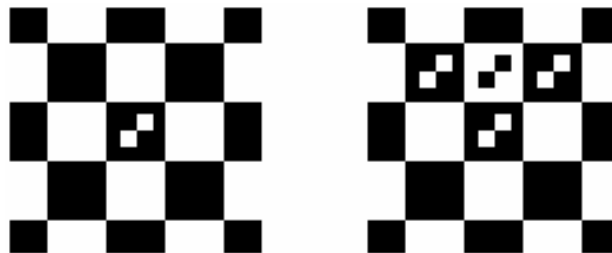
Výstupem výše uvedených měření a operací jsou nalezené body na jedné a druhé kameře. Pro jejich vzájemnou identifikaci se používají různé metody.

3.4.1 Postupné měření

Nejjednodušší způsob, při kterém problém vzájemné identifikace úplně odpadne, je postupně nasvécovat projektorem každý bod zvlášť a pořizovat pro něj snímky z jedné i druhé kamery. Způsob je nejjistější, ale pomalý. Nepoužívá se. Může být ale použit kompromis mezi tímto způsobem a některou z dalších uvedených metod.

3.4.2 Značka

Umístěním značky na jedno nebo více polí šachovnice se získá pozice v pixelech, vůči které mohou být vztaheny pozice identifikovaných bodů.



Obr. 15 Varianty značek na šachovnici

Vlevo na Obr. 15 je nejjednodušší způsob s jednou značkou. Výhoda způsobu vpravo je, že neudává jen pozici, ale také natočení vzoru promítnutého na objekt. Za předpokladu, že je objekt rovinný, se dají lehce identifikovat všechny následující body. Pro obecné objekty je možné body identifikovat tak, že na diferenční obraz bude aplikován hranový detektor, který vytvoří "mřížku". Zavedou se objekty typu "detekovaný bod" a typu "hrana". Každá hrana směřuje z jednoho detekovaného bodu do druhého. Pak je vycházeno ze známých bodů kolem značek přes hrany do dalších bodů, které jsou tímto identifikovány.

Také je možné použít kombinaci postupného měření a značky - měření po řádcích nebo po sloupcích. Složitá implementace výše uvedeného postupu pak odpadá za cenu vyšší časové náročnosti měření. Při identifikaci se vychází z apriorní znalosti, že body vlevo od značky mají nižší pixelovou souřadnici x a vpravo ji mají vyšší; resp. směrem nahoru a dolů do značky se mění souřadnice y . Jak je vidět z obrázku, pro jednu sadu snímků je naměřena jedna řada bodů.



Obr. 16 Řádková varianta mřížky

Další problém je ale samotná identifikace značky. Jednoduše je vyřešen tak, že kromě normálního a invertovaného vzoru je pořízen snímek, kdy je na objekt nasvícena pouze značka. Jiný možný způsob bez nutnosti pořízení dalšího snímku by byl použit šachovnicovou značku na mřížce nebo mřížkovou značku na šachovnici, a identifikovat zvlášť značku a zvlášť body v algoritmu provedeném až po měření.

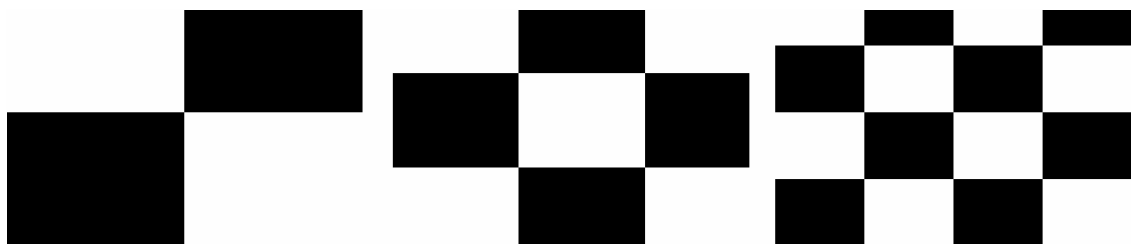
3.4.3 Postupná změna rozlišení vzoru

Tento způsob je nejlepší a v praxi se hodně používá (hlavně jeho modifikace s Grayovým kódem, viz. kapitola 3.4.4). Je provedeno několik postupných měření za sebou, a v každém z nich je postupně zvyšováno rozlišení vzoru. Například v prvním kroku je šachovnice rozdělena jen na 4 segmenty a detekován může být pouze 1 bod. Ve druhém kroku je to 9 segmentů a detekovány můžou být až 4 body, atd. . Obecně platí:

$$\begin{aligned} n_{seg} &= 2^{n+1} \\ n_{bodu} &= 2^n \end{aligned} \quad (3)$$

,kde n je číslo kroku; n_{seg} počet segmentů vzoru a n_{bodu} nejvyšší možný počet detekovaných bodů. Je důležité, aby se po každém kroku měření vynásobila velikost segmentů přesně $\cdot\sqrt{2}$. Jen tak je zajištěno, aby každý bod následujícího měření byl detekován v segmentu vzoru měření předchozího.

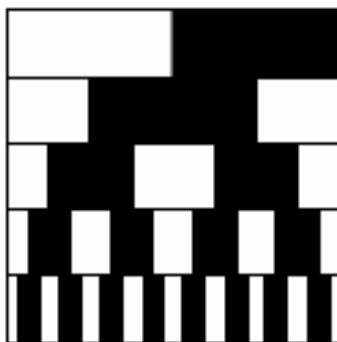
Metoda je použitelná pro libovolný vzor, ale šachovnice má navíc tu výhodu, že oblasti jsou prakticky dané už tím, kde jsou černá a bílá pole. Například u mřížky by bylo třeba obraz navíc složitě segmentovat.



Obr. 17 První tři kroky měření s postupnou změnou rozlišení

3.4.4 Grayův kód

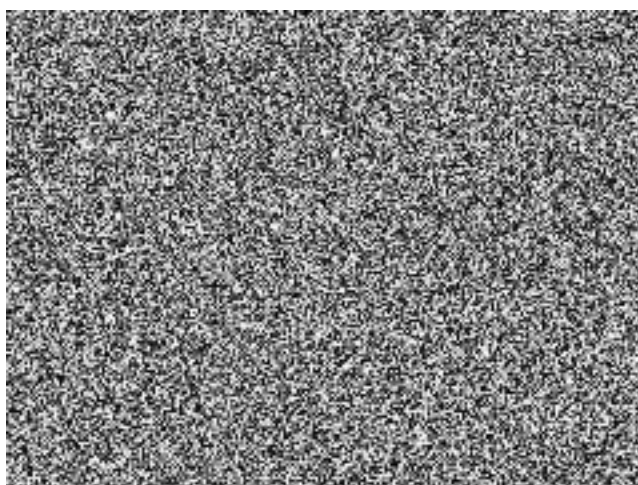
Princip je úplně stejný jako u postupné změny rozlišení vzoru. Velikost segmentů není ale násobena $\cdot\sqrt{2}$, ale sekvence černých a bílých polí je generována pomocí Grayova kódu. Takový způsob umožňuje změřit více bodů za kratší počet měření.



Obr. 18 Sekvence Grayova kódu (převzato ze zdroje [1])

3.5 Bílý šum

Bílý šum je náhodný signál s rovnoměrnou výkonovou spektrální hustotou. Je používán pro aktivní i pasivní metody. Podle rozmístění promítnutých bodů vůči sobě je detekován kód. Tím, že je kód náhodný, je detekce o hodně složitější než u pevných kódů. Výhoda metody oproti postupné změně vzoru je, že stačí změřit jeden obraz. Je tedy vhodná pro měření rychlých dějů; typicky objektů na páse. Děje ale nesmí být příliš rychlé, jinak by docházelo k rozmazání objektu na snímku. Při použití synchronizovaných kamer je možná i pasivní varianta této metody.

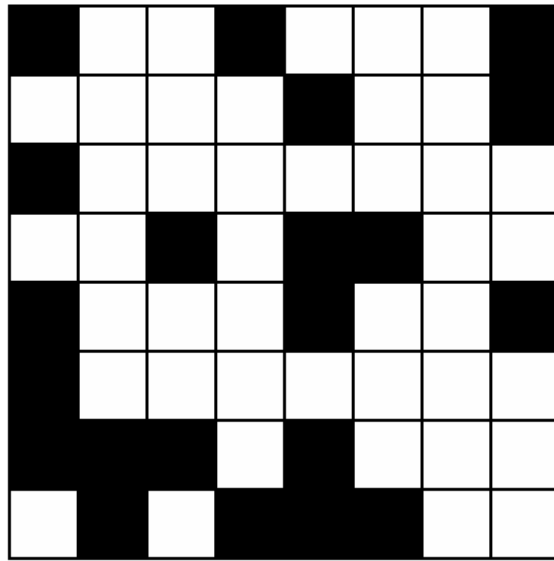


Obr. 19 Bílý šum (převzato ze zdroje [8])

3.6 Vzor s pevným kódem

Stejně jako u bílého šumu jde o metody vhodné k měření rychlých dějů, které mají složitější algoritmus detekce. Narozdíl od bílého šumu ale nejde o náhodný kód. Body jsou identifikovány podle kódu vytvořeného černými a bílými poli vzoru. Může jít například o mřížku, která má některá pole vyplněná a některá ne. Tato mřížka může být trojúhelníková, čtyřúhelníková nebo šestiúhelníková. Vyhodnocuje se určité okolí každého pole (u čtvercové

mřížky např. 3·3). Je důležité, aby okolí každého pole bylo jedinečné v rámci celého promítaného vzoru. Obr. 20 je pouze orientační.



Obr. 20 Vzor s pevným kódem

3.7 Proužky

Jako vzor se používají vodorovné a svislé pruhy, u kterých je použit nejčastěji Grayův kód. K detekci se používají hranové detektory a následně nalezení průniků snímků s vodorovným a svislým nasvícením. Metoda je stejně dobrá jako šachovnice, navíc zde odpadá chyba vyhodnocení konvoluční maskou.

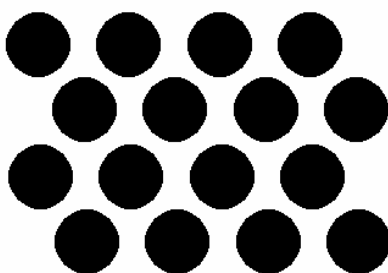
3.8 Další možné metody

3.8.1 Trojúhelníky a šestiúhelníky

Trojúhelníková nebo šestiúhelníková mřížka nebo šachovnice by mohla být použita stejně dobře jako čtyřúhelníková. Trojúhelník umožňuje změřit nejvíce bodů při daném rozlišení projektoru. Čtyřúhelník je nejintuitivnější a umožňuje nejjednodušší algoritmy. Šestiúhelník je nevhodnější při použití metod s pevným kódem.

3.8.2 Kruhy

Vyhodnocení bodů by bylo provedeno přes nalezení oblastí a výpočet středu každé z nich. Kromě toho, že by to bylo velmi nepřesné, by vzájemná identifikace bodů byla prakticky nemožná nebo neúměrně složitá vzhledem k výsledku.



Obr. 21 Kruhy

3.8.3 Použití barevných kamer

Zajímavé možnosti by byly při použití barevných kamer místo černobílých. U měření s Grayovým kódem by bylo například možné zrychlit měření za cenu pomalejšího zpracování dat. Složením následujících tří RGB složek by se nasvítil objekt, a tím by byly provedeny tři kroky "černobílé verze" měření najednou. Při zpracování dat by se naměřené snímky rozdělily zpětně na jednotlivé RGB složky.



Obr. 22 První tři kroky měření s postupnou změnou rozlišení pro RGB kamery

Při použití barevných kamer by také bylo možné použít barevné pozadí jiné barvy, než barva měřeného objektu, které by sloužilo k vyhodnocení tvaru objektu podobným, ale jednodušším způsobem než v kapitole 3.9, u kterého by se neztrácela možnost měřit na objektu body.

3.9 Nasvícování pozadí

V této metodě by nebyl nasvícován přímo snímáný objekt, ale pozadí za ním. Kamera by musela snímat objekt přesně vodorovně. Nevhodnocovaly by se body, ale pouze obrys objektu, který by se jako jediný používal pro 3D rekonstrukci.

3.10 Aktivní metody

Jde o metody, u kterých je třeba znát pozici kamer i projektoru. Krokem navíc je tedy kalibrace projektoru. Oproti pasivním metodám ale stačí pouze jedna kamera. Krokem, který tedy odpadá, je vzájemná identifikace bodů a u měření rychlých dějů také synchronizace kamer.

Kalibrace projektoru může být provedena například tak, že jím budou nasvícovány body na daných pixelových souřadnicích, a snímány již dříve kalibrovanou kamerou. Body jsou pak matematicky promítnuty na známou rovinu pomocí kalibračních parametrů (zdroj [1]).

3.11 Zvolená metoda

Po vyzkoušení nejlepších z dostupných uvedených metod byla zvolena pasivní metoda s promítáním šachovnice s postupnou změnou rozlišení. Informace je vyhodnocována ze:

- snímku nenasvíceného objektu
- snímku nasvíceného objektu
- snímků objektu nasvíceného šachovnicí
- snímků objektu nasvíceného invertovanou šachovnicí

4 MATEMATICKÝ ZÁKLAD METODY

Zde je popsán matematický aparát kalibrace, identifikace a rekonstrukce 3D bodů pro zvolenou metodu.

4.1 Prostorové transformace

Prostorové transformace jsou převzaty ze zdroje [1].

4.1.1 Homogenní souřadnice

Chceme-li v homogenních souřadnicích popsat polohu bodu v k rozměrech, potřebujeme znát uspořádanou n -tici čísel, kde $n=k+1$. 3D bod v kartézských souřadnicích zapíšeme $[X,Y,Z]$. V homogenních souřadnicích jej zapíšeme $[x,y,z,w]$, kde parametr w je váha bodu. Platí:

$$X = \frac{x}{w}; \quad Y = \frac{y}{w}; \quad Z = \frac{z}{w}; \quad w \neq 0. \quad (4)$$

V homogenních souřadnicích lze reprezentovat posloupnost lineárních transformací jedinou maticí. Transformační matice mají pro 3D rozměr 4×4 .

4.1.2 Posunutí

Vektor posunutí má ve 3D tvar $\mathbf{t}(X_t, Y_t, Z_t)$. Transformační matice je:

$$T = \begin{bmatrix} 1 & 0 & 0 & X_t \\ 0 & 1 & 0 & Y_t \\ 0 & 0 & 1 & Z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

4.1.3 Rotace

Rotace v prostoru je popsána třemi úhly α, β, γ , které reprezentují rotace bodu kolem jednotlivých rovin XY, XZ , a YZ v počátku souřadnicového systému. Nejobecnější případ transformační matice bude ve výsledku:

$$R = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) & \sin(\beta) & 0 \\ \cos(\alpha)\sin(\gamma) + \sin(\alpha)\sin(\beta)\cos(\gamma) & \cos(\alpha)\cos(\gamma) - \sin(\alpha)\sin(\beta)\sin(\gamma) & -\sin(\alpha)\cos(\beta) & 0 \\ \sin(\alpha)\sin(\gamma) - \cos(\alpha)\sin(\beta)\cos(\gamma) & \sin(\alpha)\cos(\gamma) - \cos(\alpha)\sin(\beta)\sin(\gamma) & \cos(\alpha)\cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

4.1.4 Měřítko

Měřítko se mění po osách X,Y,Z. Násobky měřítka menší než 1 znamenají zmenšení, větší než 1 zvětšení. Když označíme násobky měřítka s_x, s_y, s_z , pak transformační matice bude:

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

4.1.5 Projekce 3D - 2D

Výstupem této operace budou 3D body, které budou všechny ležet v jedné rovině. Jde o zobrazení libovolných bodů na průmětnou rovinu. Pokud je průmětná rovina kolmá na osu z a je od počátku souřadnic vzdálená o z_0 , bude transformační matice:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & z_0^{-1} & 1 \end{bmatrix} \quad (8)$$

4.1.6 Skládání transformací

Posloupnost lineárních transformací lze zapsat jednou maticí. Dílčí operace jsou násobeny. Násobení matic není komutativní - záleží na pořadí, ve kterém se operace provádějí.

4.2 Obrazové transformace

Obrazové transformace jsou převzaty z literatury [9] a [10].

4.2.1 Prahování

Jde o bodovou jasovou transformaci obrazu, která vypočítá výstupní obraz o dvou nebo třech možných hodnotách pixelů ({černá, bílá} nebo {černá, šedá, bílá}; prakticky {0, 1} nebo {0, 127, 255}). Dvouhodnotová varianta je nejběžnější a umožňuje převod šedotónového obrazu na černobílý. V této práci je ale použita i tříhodnotová varianta, viz. kapitola 8.2. U dvouhodnotové varianty platí pro šedotónový výstupní obraz:

$$\begin{array}{lll}
h(k,l) \geq \text{práh} & \dots & f(k,l) = 255 \\
h(k,l) < \text{práh} & \dots & f(k,l) = 0
\end{array} \tag{9}$$

Pro černobílý obraz dvouhodnotové varianty platí:

$$\begin{array}{lll}
h(k,l) \geq \text{práh} & \dots & f(k,l) = 1 \\
h(k,l) < \text{práh} & \dots & f(k,l) = 0
\end{array} \tag{10}$$

Pro tříhodnotovou variantu platí (za předpokladu, že $\text{práh}_1 > \text{práh}_2$):

$$\begin{array}{lll}
h(k,l) > \text{práh}_1 & \dots & f(k,l) = 255 \\
h(k,l) < \text{práh}_2 & \dots & f(k,l) = 0 \\
\text{práh}_1 \geq h(k,l) \geq \text{práh}_2 & \dots & f(k,l) = 127
\end{array} \tag{11}$$

,kde h je vždy vstupní obraz a f výstupní; a k, l jsou souřadnice všech pixelů obrazů. Prahů může být obecně ještě více, ale nejběžnější je jeden práh (výstup nabývá dvou hodnot) a dva prahy (výstup nabývá tři hodnot).

4.2.2 Konvoluce

Konvoluce je obecně základem většiny lokálních jasových transformací obrazu. Používá se pro filtraci šumu a detekci hran.

Matematický zápis 2D konvoluce je:

$$g(k,l) = \sum_i \sum_j f(k+i, l+j) \cdot \text{Konv}(i, j) \tag{12}$$

,kde f je původní obraz, g výsledný obraz, a Konv konvoluční maska.

Konvoluční maska (konvoluční jádro) musí mít vždy rozměr $n \cdot n$, kde n je liché číslo ≥ 3 .

4.2.2.1 Filtrace šumu

Konvoluční maska 3·3 pro filtraci šumu prostým průměrem je:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{13}$$

Další možnosti jsou vážený průměr, modus nebo medián. Některé z těchto možností nelze řešit konvolucí. Obraz je touto operací "vyhlazen" a dochází jí nejen ke ztrátě šumu, ale i ke ztrátě informace.

4.2.2.2 Detekce hran

K detekci hran se používá kromě jiných detektorů například Sobelův hranový detektor. Jeho varianta 3·3 může být vertikální, horizontální a diagonální. Každá z těchto variant je citlivá na jiný směr hran.

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (14)$$

Často je používána kombinace vertikální a horizontální masky. Platí:

$$hrany(k,l) = \sqrt{(vert(k,l))^2 + (hor(k,l))^2} \quad (15)$$

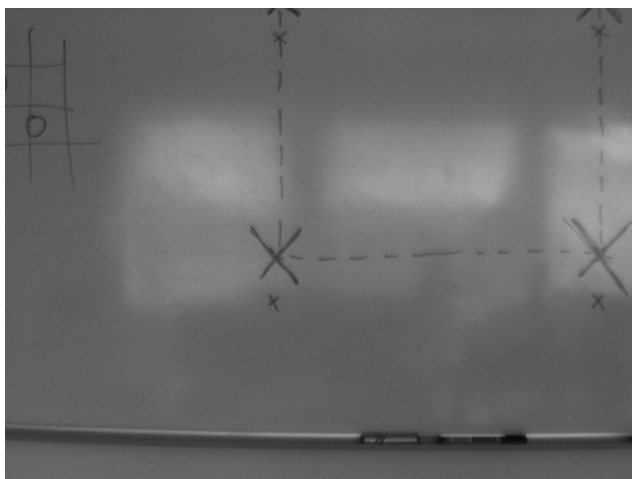
,kde *hrany* je výsledný obraz, *vert* je obraz po konvoluci vertikální Sobelovou maskou a *hor* je obraz po konvoluci horizontální Sobelovou maskou.

5 KALIBRACE SYSTÉMU

Před samotným měřením je třeba zjistit vzájemnou pozici kamer a jejich natočení. Je k tomu použita kalibrační tabule se známými body. Postup kalibrace je inverzní k postupu 3D rekonstrukce. Kapitola popisuje všechny kroky potřebné ke správné kalibraci systému.

5.1 Změření zorného pole kamery

Pro kalibraci i 3D rekonstrukci z obrazů kamer je třeba znát poměrné roztažení jejich zorného pole na X i na Y. Na tabuli byl nakreslen obdélník známých rozměrů, který byl natočen rovnoběžně s X a Y souřadnicemi kamery. Kamera byla postavena kolmo na rovinu tabule ve vzdálenosti l tak, aby byly na snímku všechny body obdélníku.

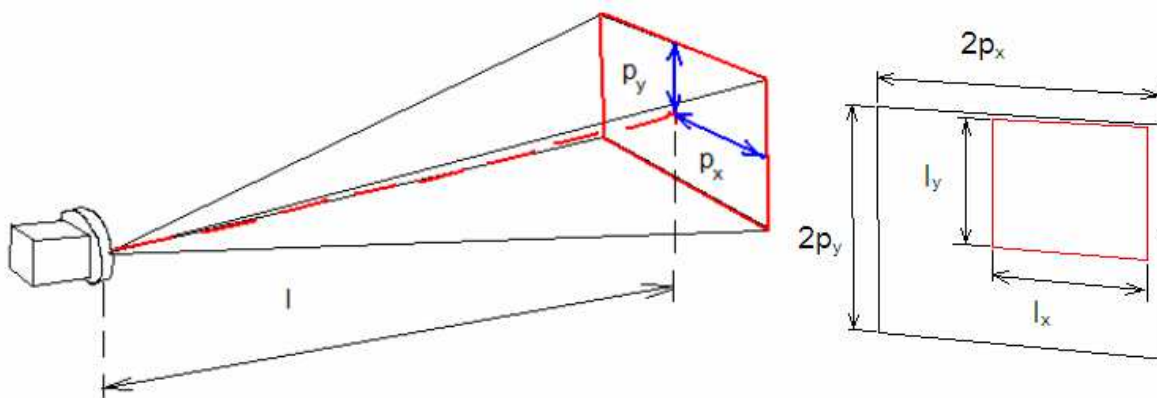


Obr. 23 Měření zorného pole kamery

Po uložení snímku z něj byly odečteny délky stran v pixelech a zjištěno, kolik dělá jeden pixel na každé ose při vzdálenosti l . Roztažení p_x a p_y pro tuto vzdálenost je vypočítáno:

$$\begin{aligned} p_{x,l} &= l_x \text{ [px]} / l_x \text{ [cm]} \\ p_{y,l} &= l_y \text{ [px]} / l_y \text{ [cm]} \end{aligned} \quad (16)$$

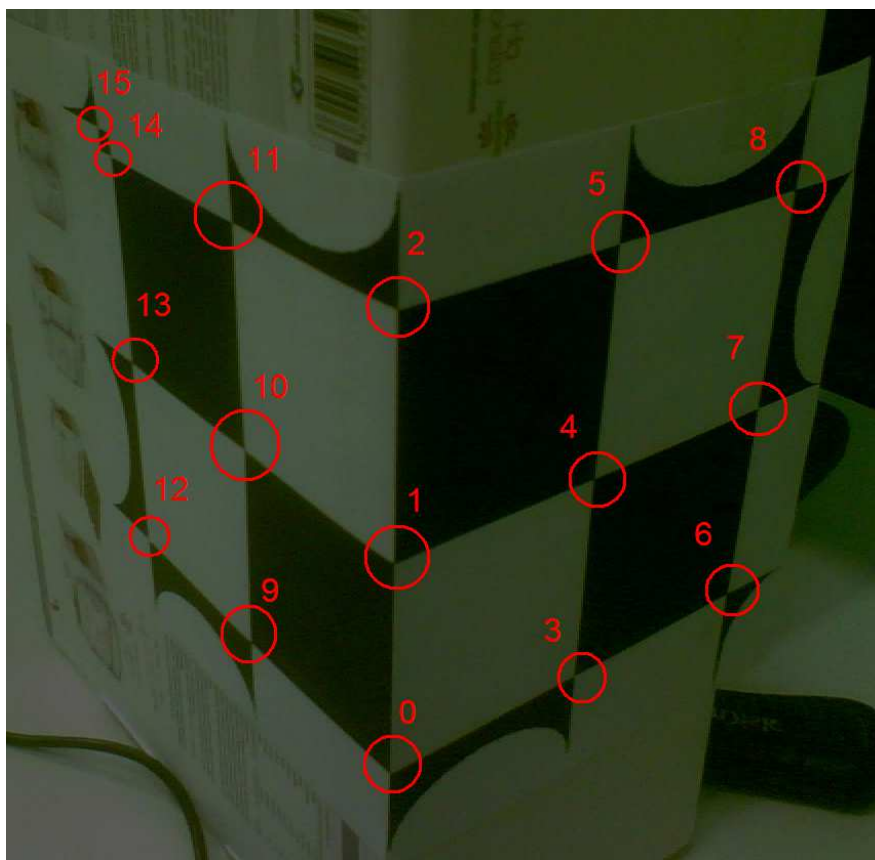
Obr. 24 ilustruje hlavní proměnné tohoto měření.



Obr. 24 Výpočet zorného pole kamery

5.2 Kalibrační tabule

K měření se používá kalibrační tabule. Jsou pořízeny snímky z obou kamer tak, aby všechny body kalibrační tabule byly na každém snímku. Body jsou detekovány na rozhraní každých čtyř sousedních políček šachovnice. Každému bodu je přiděleno unikátní číslo. Bod 15 je významný při jednoznačné identifikaci všech bodů.



Obr. 25 Kalibrační tabule

Na kalibrační tabuli je důležité, aby se její body měnily ve všech souřadnicích. Rovinná

tabule by nestačila ke kalibraci, protože by nemohly být nalezeny všechny neznámé kalibračních rovnic. Proto je tabule přeložena v úhlu 90°.

Tabulka souřadnic bodů na kalibrační tabuli																
BOD	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X	0	0	0	p	p	p	2p	2p	2p	0	0	0	0	0	0	0
Y	0	p	2p	0	p	2p	0	p	2p	0	p	2p	0	p	2p	2p+p ₂
Z	0	0	0	0	0	0	0	0	0	p	p	p	2p	2p	2p	2p+p ₂

Tab. 1 Souřadnice bodů na kalibrační tabuli

V Tab. 1 je napsáno, jakým **zafixovaným** prostorovým souřadnicím odpovídají body kalibrační tabule. Proměnná p je délka hrany jednoho políčka, p_2 je délka hrany políčka mezi bodem 14 a bodem 15. Algoritmus kalibrace spočítá pozici kamer, jejich směrové vektory, a vektory x-ových a y-ových pixelových os kamery ve 3D prostoru. Tyto informace pak umožňují rekonstruovat 3D body z naměřených dat.

5.3 Rozpoznání bodů

Z obou snímků musí být nejdřív získány pixelové souřadnice jednotlivých bodů tabule. Používá se k tomu šachovnicová konvoluční maska, která vypadá při rozměru 7·7 takto:

$$\check{S}ach_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix} \quad (17)$$

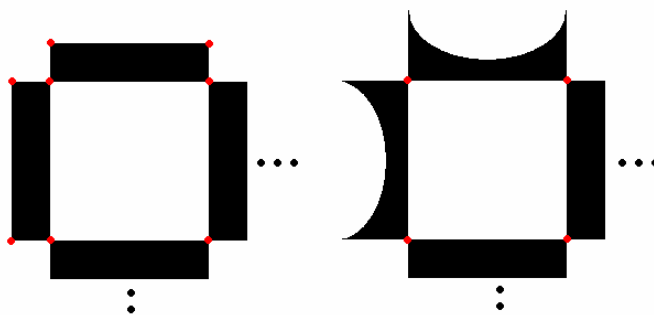
Tato maska ale může mít libovolný rozměr $n \cdot n$, kde n je liché číslo ≥ 5 . Příliš malá maska nerozezná body od náhodného šumu, hran objektů, atd. . Příliš velká maska také snižuje selektivitu, protože jsou jí brány i pixely mimo detekovanou šachovnici. Při aplikaci se používalo $n = 15$. Pro libovolné n by maska byla popsána:

$$\begin{array}{llll} \text{pro } i > n/2 & a & j > n/2 & \dots & Konv_{i,j} = 1 \\ \text{pro } i \leq n/2 & a & j \leq n/2 & \dots & Konv_{i,j} = 1 \\ \text{pro } i \leq n/2 & a & j > n/2 & \dots & Konv_{i,j} = -1 \\ \text{pro } i > n/2 & a & j \leq n/2 & \dots & Konv_{i,j} = -1 \end{array} \quad (18)$$

Už přímo během konvoluce se na výsledek aplikuje absolutní hodnota. Konvoluce totiž nabývá kladných i záporných hodnot. Obraz výsledné konvoluce je dále vyprahován. Hodnota prahu byla nejdřív nastavována ručně. K programovému nastavení by bylo třeba zjistit, jaké

parametry ovlivňují správnou hodnotu prahu. Bylo zjištěno, že správný práh je určitým poměrem maximální hodnoty konvoluce v rozsahu $0.15 \max - 0.85 \max$. V ideálním případě by bylo vycházeno z apriorní znalosti, že bodů na kalibrační tabuli je 16, a práh by byl postupně přibližován tak dlouho, dokud by nalezených bodů nebylo 16. Tato metoda je ale nepoužitelná, protože na obrazu se zpravidla nachází místa, která mají konvoluci stejně vysokou jako samotné body. Proto je práh nastavován záměrně na hodnotu vyšší, a body jsou dále vybírány podle dalších kritérií.

Kalibrační tabule byla přizpůsobena tak, aby na ní nebyly vyhodnocovány jiné body než na rozhraní čtyř polí. Obecně byly tyto neexistující body nacházeny na rozích okrajových polí, což je patrné ze zvolené konvoluční masky a ilustrováno na Obr. 26. Červeně jsou vyznačeny nalezené body. V levé části obrázku vidíme původní okrajové pole s chybně vyhodnocenými body. V pravé části je jeho modifikace, která slouží k eliminaci této chyby.



Obr. 26 Chybné vyhodnocení bodů konvoluční masky

K další chybě docházelo vlivem osvětlení. Jedna stěna kalibrační tabule je obvykle osvětlena více než druhá, což způsobuje, že bílá pole mají na kameře vyšší pixelovou hodnotu, a tím jsou konvoluce bodů na jedné stěně výrazně vyšší než na druhé. Tato chyba většinou vedla ke ztrátě všech bodů na jedné stěně. Problém byl vyřešen modifikací algoritmu konvoluce. Před konvolucí každého pixelu je vyhledána v jeho n -okolí maximální hodnota jasu pixelu, a výsledek konvoluce je dělen touto hodnotou. Podmínka je, že rozdíl maximální a minimální hodnoty nesmí být menší než určitý práh - tím se předchází občasně chybné identifikaci bodů na ostatních objektech. Černá pole šachovnice měla i při velkém osvětlení v laboratoři jasovou hodnotu maximálně 27, a proto byl tento práh s rezervou zvolen jako 50.

Největší chyba metody s šachovnicovou konvoluční maskou je, že není citlivá na velká natočení podobná jako na Obr. 27.



Obr. 27 Chyba vyhodnocení natočených bodů konvoluční maskou

Na zeleně označených bodech je konvoluce s danou maskou funkční. Na červeně označených bodech má ale hodnoty tak nízké, že práh konkrétně na tomto obraze musel být nastaven tak nízko, aby bodů bylo rozpoznáno až 122, aby i tyto body byly rozpoznány. U modře označených bodů je tento problém také, jen se neuplatní tak intenzivně. Jako řešení tohoto problému byla navržena druhá konvoluční maska:

$$\check{S}ach_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix} \quad (19)$$

Bylo uváženo i to, že natočení může nyní být i obráceně. Proto je zavedena ještě konvoluční maska $\check{S}ach_3$, která je zrcadlově převrácenou variantou masky $\check{S}ach_2$ podle osy y . Matematicky:

$$\check{S}ach_3(i, j) = \check{S}ach_2(i, n - j) \quad (20)$$

,kde i a j jsou x -ová a y -ová pozice na masce a n je její rozměr. Z výsledků pro masky $\check{S}ach_1$, $\check{S}ach_2$ a $\check{S}ach_3$ bylo vybíráno maximum pro každý pixel. Před tím bylo experimentováno se skládáním výsledků podobným jako pro vertikální a horizontální masky konvoluce podle vzorce:

$$Konv = \sqrt{\check{S}ach_1^2 + \check{S}ach_2^2 + \check{S}ach_3^2} \quad (21)$$

,ale lepších výsledků bylo dosaženo prostým výběrem maxima.

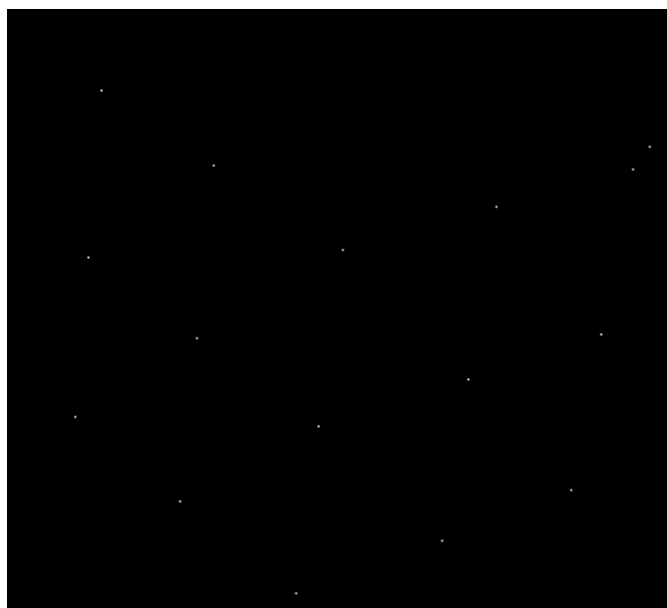
Poslední krok této části je nalezení lokálních maxim. Kolem každého bodu je prohledáno okolí 3 pixelů a pokud žádný nemá větší hodnotu $Konv$, je tento bod označen. Práh byl až po provedení těchto operací nastaven tak, aby bylo nalezeno 16 bodů s nejvyšší konvolucí. Tato opatření se ukázala jako dostatečná pro funkčnost kalibrace. Výsledek může vypadat viz. Obr. 28.



Obr. 28 Srovnání rozpoznávaných bodů kalibrační tabule

5.4 Identifikace bodů kalibrační tabule

Podmínkou správné identifikace je, aby na obraze zůstaly jen skutečné body kalibrační tabule.



Obr. 29 Výsledek rozpoznání bodů

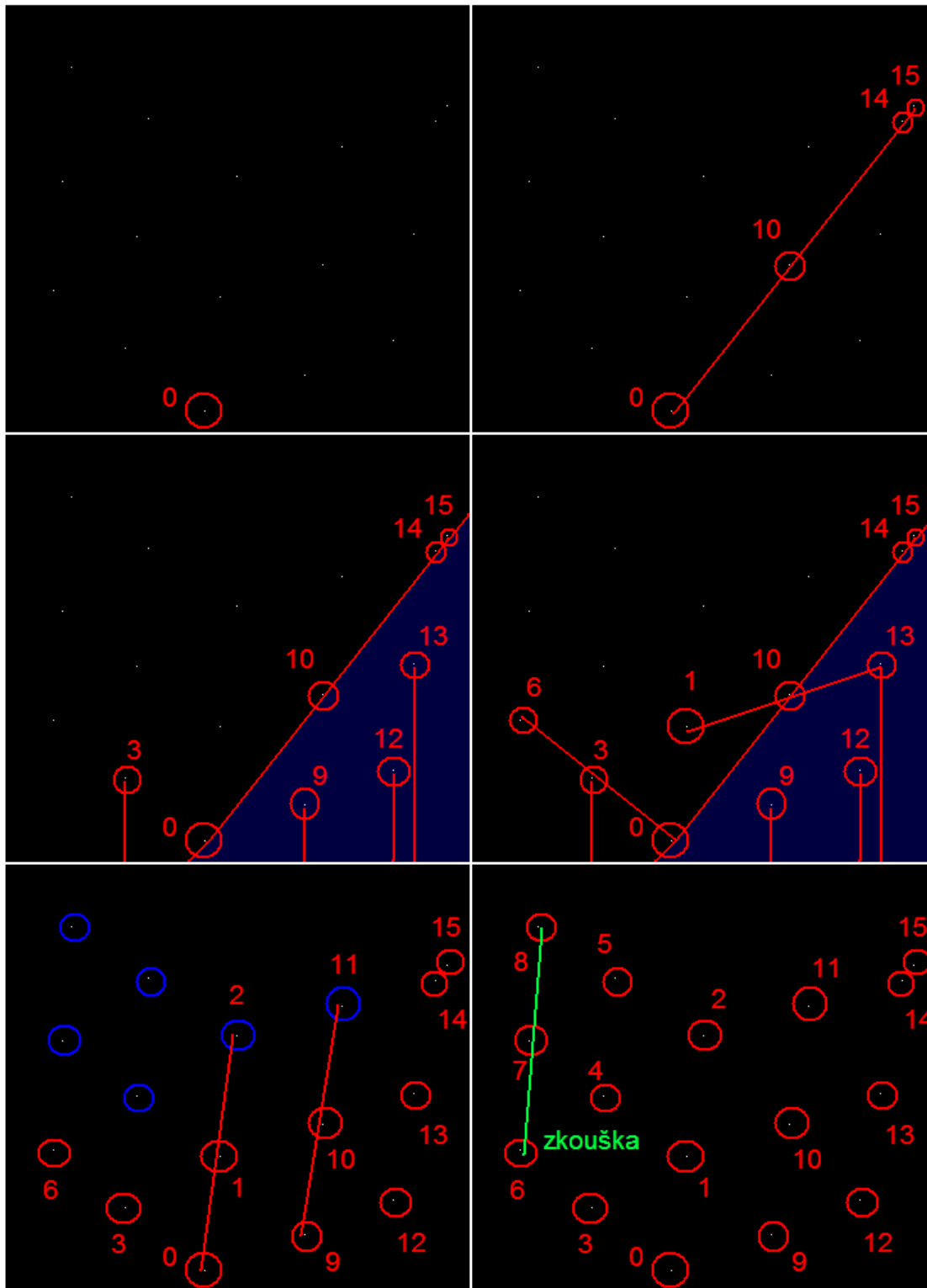
Je třeba přiřadit těmto bodům odpovídající čísla z Obr. 25. Významné jsou při identifikaci body 0 - nejspodnější v místě ohybu; a bod 15 - jediná nesymetrická značka na rohu. Vychází se z předpokladu, že kamery míří na kalibrační tabuli shora, a jsou natočeny přibližně rovnoběžně s osou Z. Díky tomu může být bod umístěný na nejspodnější souřadnici identifikován jako **bod 0** - viz. Obr. 30 vlevo nahoře.

Z tohoto bodu je pak hledáno spojení s takovým bodem, u kterého vzniklá úsečka protíná nebo je blízko dvou dalším bodům. Když je tento bod nalezen, může být identifikován jako **bod 15**, a protnuté body jako **bod 14 a bod 10** na základě toho, který je bližší k bodu 15. To je ilustrováno na Obr. 30 vpravo nahoře.

Dále je třeba pomocí funkcí knihovny pro geometrii (viz. kapitola 7) rozdělit prostor přímkou definovanou bodem 0 a bodem 15. U všech zbývajících bodů se pak zjišťuje, na které straně této přímky leží. Je jasné, že na jedné straně bude 9 bodů a na druhé 3. Tyto 3 body budou identifikovány jako první. Pokud kamery skutečně snímají kalibrační tabuli shora, bude vždy **bod 9** těchto bodů nejspodnější, **bod 12** bude druhý nejspodnější a **bod 13** třetí nejspodnější. Ze druhé strany přímky je možné identifikovat zatím pouze **bod 3** jako nejspodnější - viz. Obr. 30 vlevo uprostřed.

V následující části jsou již pouze dopočítávány body ze vždy ze znalostí, které body musí být na jedné přímce. Identifikace se provádí v následujícím pořadí.

Bod 0 a bod 3 = **bod 6**. Bod 13 a bod 10 = **bod 1**. Bod 9 a bod 10 = **bod 11**. Bod 0 a bod 1 = **bod 2**. Bod 2 a bod 6 = **bod 4**. Bod 1 a bod 4 = **bod 7**. Bod 3 a bod 4 = **bod 5**. Bod 2 a bod 5 = **bod 8**. Zkouška je, že bod 6 a bod 7 musí být na přímce s bodem 8. V případě, že zkouška vyjde u obou snímků, jsou kalibrační data získána a je možné přejít k dalšímu kroku.



Obr. 30 Postup při identifikaci bodů

5.5 Výpočet kalibrace

Kalibrace z identifikovaných bodů probíhá pro každou kameru zvlášť, a z toho pro každou osu XYZ zvlášť. Ke kalibraci je potřeba znát aspoň tři body kalibrační tabule, které

nesmí být v jedné její rovině, a jsou pokud možno co nejdál od sebe. Pro všechny body na obou kamerách platí:

$$\overline{KAM_{LOOK}} + p_x(\overline{PIX_x} - 360)\overline{VEKX} + p_y(\overline{PIX_y} - 240)\overline{VEKY} = \overline{BOD} \quad (22)$$

,kde **neznámá** KAM_{LOOK} je prostorový bod, do kterého je kamera natočená

; **konstanty** p_x a p_y jsou roztažení zorného pole kamery

; PIX_x a PIX_y jsou **vstupní** pixelové souřadnice daného bodu

; **neznámé** $VEKX$ a $VEKY$ jsou vektory odpovídající pixelovým souřadnicím v prostoru

; **vstupní** BOD je odpovídající zafixovaný prostorový bod kalibrační tabule

Sestavením třech rovnic byla získána matice o třech výše uvedených neznámých, které byly vypočítány. Po získání X , Y , a Z složek byl vypočítán směrový vektor kamery VEK_{SMER} . Při jeho výpočtu bylo využito znalosti, že vektory $VEKX$ a $VEKY$ jsou na sebe kolmé, a byl získán jako jejich vektorový součin.

K tomu, aby bylo možné provádět 3D rekonstrukci, tyto informace stačí. Skutečná pozice kamery je irelevantní. Má význam jenom v určení přesnosti kalibrace, pokud je známa skutečná vzdálenost kamer. Pro výpočet 3D bodů jsou důležité vektory $VEKX$ a $VEKY$ a bod KAM_{LOOK} .

Určení pozice kamery se provádí následovně. Je zvolen druhý bod na přímce dané bodem KAM_{LOOK} a vektorem VEK_{SMER} . Tento bod musí být na správné straně kalibrační tabule - na straně, kde je kamera skutečně. Tuto stranu reprezentuje znaménko ve výrazu

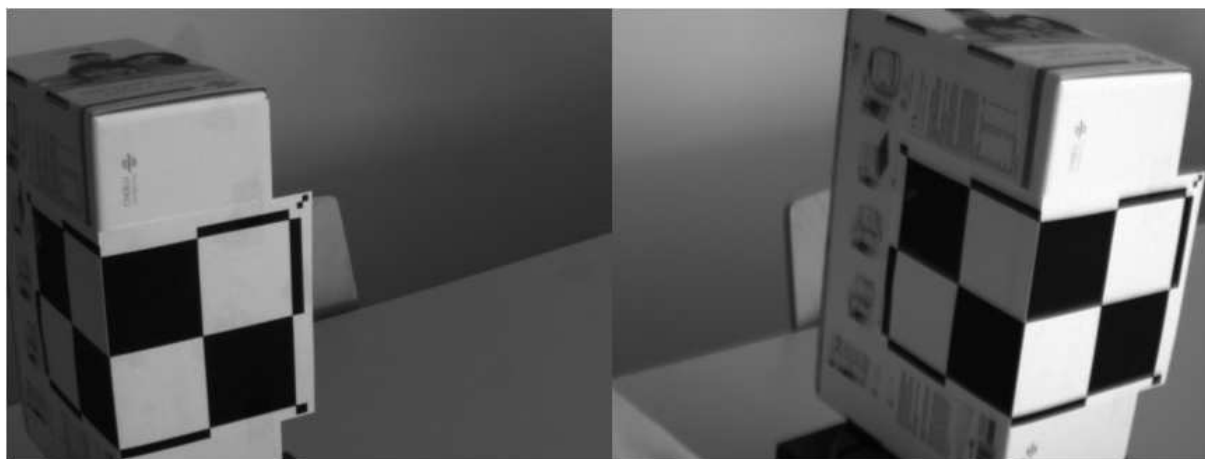
$$KAM_{POM} = KAM_{LOOK} \pm l \cdot VEK_{SMER} \quad (23)$$

,kde l je vzdálenost, pro kterou bylo měřeno zorné pole kamer (v tomto případě 225 cm)

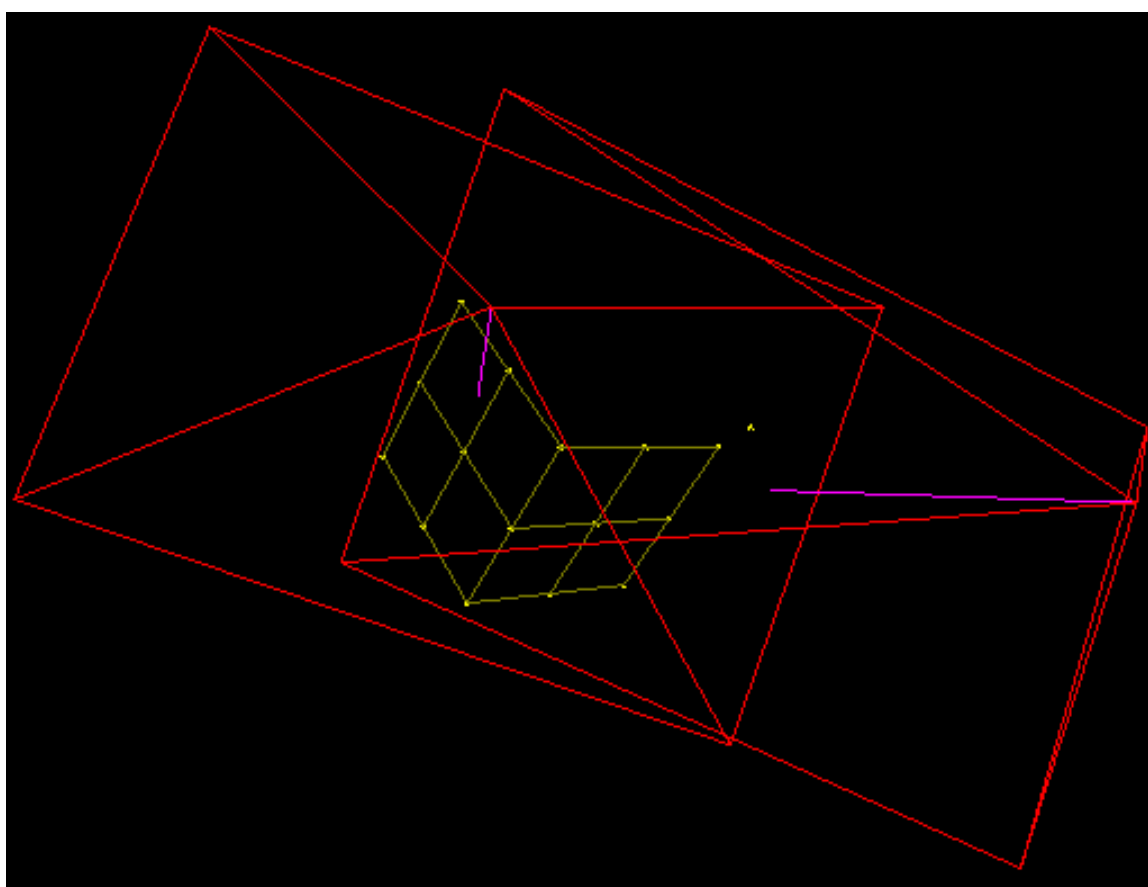
; KAM_{POM} je pomocný bod pro určení pozice kamery.

Znaménko je zvoleno na základě znalosti přehnutí kalibrační tabule. Ta musí mít body $\{0,1,2\}$ v místě ohybu dál než body na krajích $\{6,7,8\}$ a $\{12,13,14,15\}$. Pokud by bylo znaménko zvoleno špatně, byla by pozice kamery vypočítána se záporným parametrem a vyšla by na opačné straně přímky.

Nyní jsou ze tří vyhodnocovaných měřených bodů vybrány všechny dvojice a následující krok je proveden pro všechny zvlášť. Je spočítán nejbližší bod přímek mezi oběma body z dvojice a pomocným bodem KAM_{POM} . Výsledky pro všechny tři dvojice jsou zprůměrovány a uloženy jako pozice kamery KAM_{POS} . Na obrázcích je vidět srovnání vstupních dat (Obr. 31) a výstupu kalibrace pro obě kamery (Obr. 32).



Obr. 31 Snímky kalibrační tabule z obou kamer



Obr. 32 Vizualizace výpočtu kalibrace

Na Obr. 32 je žlutě kalibrační tabule, fialově jsou směry natočení kamer, a červeně jejich zorná pole. Při srovnání se vstupními daty na Obr. 31 je vidět, že středy obrázků přibližně odpovídají vypočítaným směrům kamer.

5.6 Zpřesnění kalibrace

V předchozí kapitole je popsán postup kalibrace pomocí tří bodů tabule. Použit samotné tři body je ale velmi nepřesné a výsledek je s jejich použitím pouze přibližný. Následující kapitoly popisují způsoby, jak zpřesnit kalibraci za použití všech ostatních bodů.

5.6.1 Průměrování

Nejjednodušší způsob zpřesnění kalibrace je zopakovat ji pro několik různých trojic bodů a výsledky mezi sebou zprůměrovat. Také je možné průměrovat zvlášť změřené pixelové souřadnice kalibrační tabule a zvlášť jejich odpovídající 3D reprezentace a následně provést výpočet pro tyto průměry. Musí se ale zvolit takové body, aby tyto průměry neležely na jedné přímce.

Navíc z Obr. 33 je vidět, že některé trojice bodů mají dobré výsledky pro jednu kameru, a pro druhou mají výsledky s takovou chybou, že je kalibrace nepoužitelná. U jiných výsledků to zase může být naopak. Proto je průměrování nedostatečným řešením tohoto problému.



Obr. 33 Chyba kalibrace

5.6.2 Vážené průměrování

Je třeba zvolit takovou metodu, která výsledky váží a dokáže tak eliminovat výsledky s velkou chybou. Když se podíváme na vstupní snímky na Obr. 31, můžeme zjistit poměry vzdáleností bodů kalibrační tabule od středu snímku. Proto musí po projekci vypočítaných

bodů na rovinu kamery danou vektory $VEKX$ a $VEKY$ být poměry vzdáleností takto získaných bodů od bodu KAM_{LOOK} co nejpodobnější. Toto kritérium by pak mohlo sloužit pro vážení průměrovaných výsledků.

5.6.3 Metoda nejmenších čtverců

Předchozí metoda by se dala vylepšit jako hledání minima kvadrátů odchylek poměrů vzdáleností 2D bodů od středu. Tato metoda by byla nejvhodnější, protože by s jejím použitím bylo možno získat nejpřesnější výsledky. Minimum by muselo vyhledáváno iterativními metodami, které jsou časově náročné, a pro daný počet rovnic složité.

6 NASNÍMÁNÍ DAT

6.1 Příprava pracoviště

Pro správné a co nejpřesnější měření je třeba, aby byl objekt **při jakémkoli natočení** v záběru obou kamer všemi svými částmi, a zároveň byly kamery objektu co nejbližší. Musí být zajištěny takové světlené podmínky, aby objekt nebyl přesvícen. Musí proto být stíněn. V okolí podstavce a objektu, ani za ním, nesmí být jiné objekty, na které by projektor promítal vzor. Software pro detekci bodů k tomu není přizpůsoben.

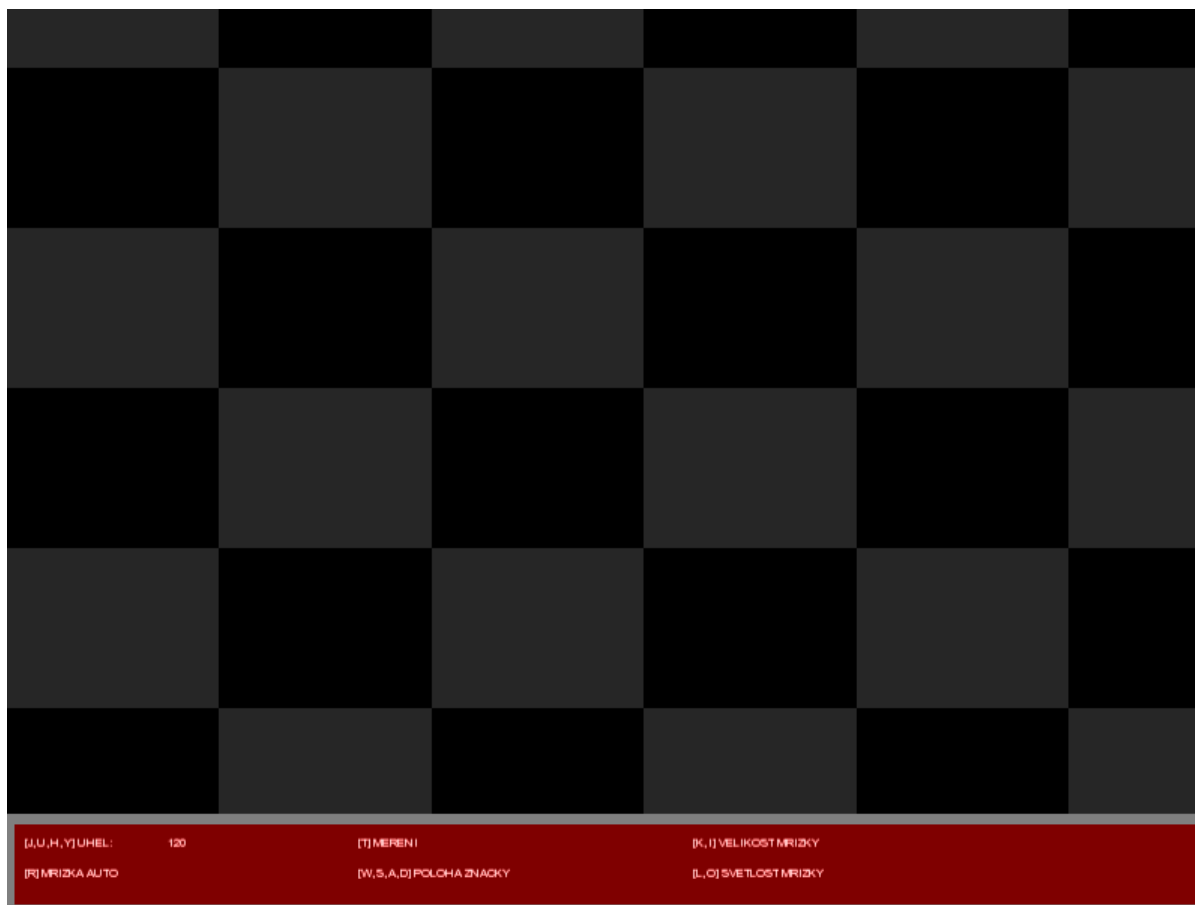
6.1.1 Značky na podstavci

Před měřením musí být na podstavec nalepeny značky (viz. Obr. 46). Tyto značky slouží k určení osy otáčení objektu. Tato znalost je využita při skládání 3D modelů jednotlivých úhlů. Značky jsou detekovány ze snímku nenasvíceného objektu. Pro jejich detekci je potřeba, aby kamera mířila na objekt shora pod dostatečným úhlem. Je nutné, aby při měření byly čitelné tři značky pro určení roviny. Protože podstavec má známé rozměry a značky mají od sebe známé vzdálenosti, můžou být tyto značky použity i pro vyhodnocení přesnosti měření.

6.2 Program pro měření

Pro částečnou automatizaci měření byl vytvořen software v jazyce C++, který používá knihovny OpenCV, OpenGL a SDL. Je ovládán pouze z klávesnice, a to:

[Esc] Ukončit	[T] Automatizované měření
[J,U] Jemné nastavení úhlu	[H,Y] Hrubé nastavení úhlu
[L,O] Man. nastavení světlosti vzoru	[R] Aut. nastavení světlosti vzoru



Obr. 34 Rozhraní měřícího programu

6.2.1 Ukládání obrazu z kamer

Pro ukládání snímků se používá knihovna OpenCV. Formát názvu souboru je vždy "UUURMK".bmp

;kde UUU je úhel ve stupních nastavený v programu (na Obr. 34 vypsán vlevo dole)

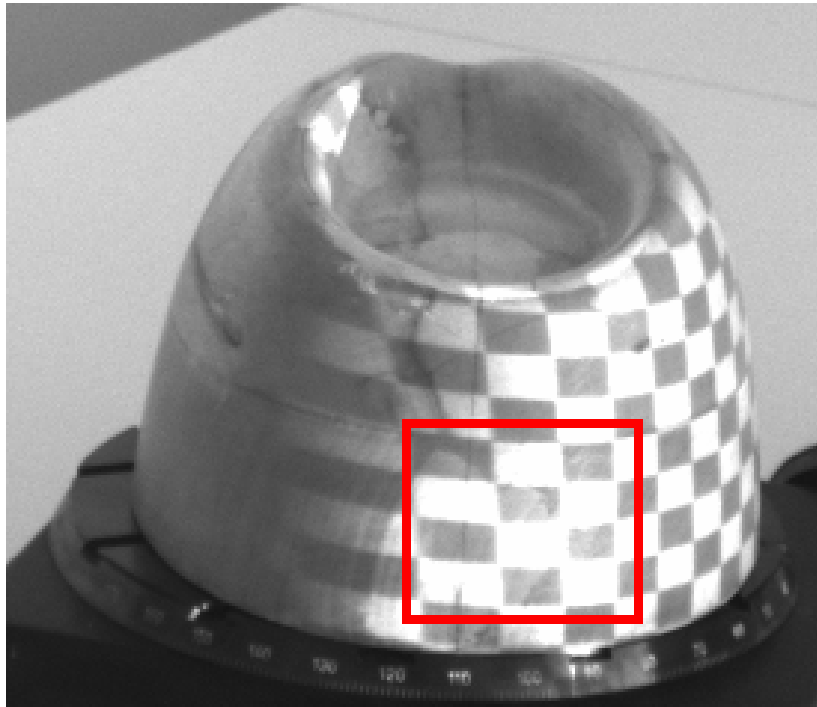
;R je dané rozlišení vzoru od 0 - nejhrubší po 9 - nejjemnější

;M je mód měření, 1 pro šachovnici, 2 pro inverzní šachovnici, 3 pro zcela nasvícený objekt, a 4 pro zcela nenasvícený objekt

;K je číslo kamery (1 nebo 2)

6.2.2 Nastavení světlosti vzoru

Při spuštění volby automatického nastavení je nastavena světlost vzoru tak, aby objekt nebyl přesvícen a nedocházelo ke ztrátě dat (viz. Obr. 35). Na obrázku je vyznačeno přesvícené místo, kde bude konvoluce v bodech velmi nízká. Jak je vidět, černá pole mají větší jas, a na rozhraní splývají s bílými. K eliminaci tohoto jevu často stačí správně nastavit světlost promítaného vzoru. Jediný případ, kdy toto nepomůže, jsou odlesky. U těch ale může částečně pomoci metoda difference, popsána v kapitole 8.2.



Obr. 35 Ztráta dat přesvícením vzoru

Nastavení světlosti je založeno na počítání bílých pixelů s nejvyšší hodnotou jasu - 255. Pokud jsou některá místa přesvětleny, budou mít hodnotu právě 255. Proto je nejdřív pořízen snímek nenasvíceného objektu, a jsou v něm spočítány pixely s hodnotou 255. Další kroky probíhají ve smyčce - hodnota světlosti vzoru je postupně přibližována. V prvním kroku je světlost vzoru nastavena na hodnotu 127. Je pořízen snímek a jsou spočítány pixely s hodnotou 255. Pokud jich je více, než na nenasvíceném objektu, je znaménko mínus - světlost se odečítá. Pokud jich je stejně nebo méně, je znaménko plus. Platí:

$$\text{světlost}_{\text{nová}} = \text{světlost}_{\text{původní}} \pm 2^{7-n} \quad (24)$$

,kde n je číslo kroku.

Kromě automatického nastavení světlosti lze světlost nastavit také manuálně.

6.2.3 Automatické měření

Tato volba naměří všechny potřebné snímky z obou kamer a uloží je způsobem uvedeným v kapitole 8.1. Postup automatického měření je následující:

1. Zakázání zobrazení menu
2. Pořízení snímku nenasvíceného objektu z obou kamer
3. Pořízení snímku nasvíceného objektu z obou kamer
4. Smyčka probíhá 10krát
 - a) Pořízení snímku objektu nasvíceného šachovnicí z obou kamer
 - b) Pořízení snímku objektu nasvíceného inverzní šachovnicí z obou kamer

c) Nastavení nového kroku šachovnice

5. Povolení zobrazení menu

Pořízení jednoho snímku z jakékoli kamery probíhá podle postupu:

1. Nastavení názvu souboru podle formátu uvedeného výše
2. Změna parametrů vzoru
3. Překreslení zobrazení
4. Ustálení parametrů kamery
5. Pořízení a uložení snímku

Před pořízením jakéhokoli snímku je nejdříve nastaven název výstupního souboru. Po každé změně vzoru musí být zobrazení překresleno, aby se projevíly změny. Musí být volána zobrazovací funkce. Dále musí být ustálen jas a kontrast kamery, který je nastavován automaticky. Obraz je několikrát aktualizován, než se tyto hodnoty ustálí. Počet aktualizací obrazu byl experimentálně zjištěn na 20. Až pak může být snímek pořízen. Všechna měření v této aplikaci jsou provedena pro obě kamery.

Před začátkem měření je zakázáno zobrazení menu na spodní části rozhraní programu. Menu by způsobovalo přesvícení a nežádoucí chyby měření. Potom je pořízen nasvícený i nenasvícený snímek objektu, které stačí změřit jen jednou, což vychází z předpokladu, že se nebudou příliš měnit světlené podmínky v místě měření. Dále jsou ve smyčce postupně změřeny pro všechna rozlišení vzoru snímky šachovnice a inverzní šachovnice. Na konci každého kroku je krok šachovnice vynásoben $\cdot\sqrt{2}$, což zajistí snadnou vzájemnou identifikaci bodů. Kroků je 10. Pro měření s vyšším rozlišením dochází k velkým chybám v samotném rozpoznání bodů i v jejich vzájemné identifikaci, a proto nemají tato další měření význam. Po skončení smyčky je opět obnoveno menu.

Menu je důležité jen kvůli zobrazení úhlu. Úhlem jsou popsány výstupní soubory z tohoto měření a je používán při rekonstrukci výsledného objektu. Je na podstavci nastavován ručně.

7 KNIHOVNY PRO 3D GEOMETRII

Při kalibraci, výpočtu a zobrazování 3D modelu byly napsány knihovny pro 3D geometrii. Následuje seznam funkcí a maker, které jsou používány. Pokud jde o makra, bude uváděn doporučený typ všech vstupních proměnných; a také, jestli jde podle významu funkce o "vstupy" nebo "výstupy". Znak "výstupů" obecně je, že po průběhu makra se jejich hodnota změní.

7.1 Vzdálenost dvou bodů

```
#define vzdalenostbodu(data_vstup3, data_vstup4, data_vystup3)
    ,kde jako data_vstup3 a data_vstup4 by se měl udávat typ float[3] a jde o vstup
    ;jako data_vystup3 by se měl udávat typ float a jde o výstup
```

Makro počítá vzdálenost dvou 3D bodů *data_vstup3* a *data_vstup4* a ukládá ji do proměnné *data_vystup3* podle analytického vzorce:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (25)$$

,kde d je vzdálenost bodu $[x_1, y_1, z_1]$ od bodu $[x_2, y_2, z_2]$

7.2 Velikost vektoru

```
#define velikostvektoru(vektor, velikost_vektoru)
    ,kde jako vektor by se měl udávat typ float[3] a jde o vstup
    ;jako velikost_vektoru by se měl udávat typ float a jde o výstup
```

Makro vypočítá velikost vektoru a používá se mimo jiné při úpravě vektoru na jednotkový. Velikost vektoru je počítána jako vzdálenost jeho hodnot od počátku.

7.3 Jednotkový vektor

```
#define jednotkovyvektor(vektor)
    ,kde jako vektor by se měl udávat typ float[3] a jde o vstupně - výstupní proměnnou
```

Makro normalizuje daný vektor na jednotkový. Je vypočítána jeho velikost a každá jeho složka je touto velikostí dělena.

7.4 Vektorový součin

```
#define vektorovysoucín(vektor1, vektor2, vyslednyvektor)
```

,kde jako *vektor1* a *vektor2* by se měl udávat typ float[3] a jde o vstupy

;jako *vyslednyvektor* by se měl udávat typ float[3] a jde o výstup

Vektorový součin dvou 3D vektorů.

7.5 Nejbližší bod přímky ke přímce

```
#define primkaNprimka(rovnice_bod42, rovnice_vektor42,  
rovnice_bod43, rovnice_vektor43, nejblicsibod43, vzdalenost)
```

,kde jako *rovnice_bod42*, *rovnice_vektor42*, *rovnice_bod43* a *rovnice_vektor43* by se měl udávat typ float[3] a jde o vstupy

;jako *nejblicsibod43* by se měl udávat typ float[3] a jde o výstup

;jako *vzdalenost* by se měl udávat typ float a jde o výstup

Makro vypočítá bod *nejblicsibod43* na přímce dané bodem *rovnice_bod42* a vektorem *rovnice_vektor42*, který je nejbližší k přímce dané bodem *rovnice_bod43* a vektorem *rovnice_vektor43*. Vzdálenost tohoto bodu od přímky je uložena do proměnné *vzdalenost*.

Hledání nejbližšího bodu je založeno na tom, že nejkratší spojnice mezi oběma přímkami má směrový vektor rovný vektorovému součinu směrových vektorů těchto přímek, a nejbližší bod musí ležet na této spojnici. Parametrický popis přímky v prostoru je:

$$\overline{BOD_P} = \overline{POSUN_P} + \overline{VEKTOR_P} \cdot t \quad (26)$$

,kde BOD_P je vypočítaný bod náležící přímce; $POSUN_P$ je známý bod náležící přímce, $VEKTOR_P$ je její směrový vektor; a t je parametr různý pro každý různý BOD_P na této přímce. Víme, že nejbližší bod leží na nejkratší spojnici, a zároveň na obou vyhodnocovaných přímkách. Proto dostáváme soustavu rovnic:

$$\overline{POSUN_P} + \overline{VEKTOR_P} \cdot t = \overline{POSUN_Q} + \overline{VEKTOR_Q} \cdot s \quad (27)$$

,kde $POSUN_Q$ je počáteční posunutí nejkratší spojnice, $VEKTOR_Q$ je její již dříve vypočítaný směrový vektor a s je její parametr. Pro třírozměrný prostor se tedy zdá, že neznámých je 5 - tři složky bodu $POSUN_Q$, parametr t a parametr s . Dvě složky bodu $POSUN_Q$ ale můžou být pro tento výpočet zafixovány na konstantní hodnoty. Pak je již získána soustava tří rovnic o třech neznámých. Postup se dá použít pro obě přímky a výsledky popřípadě zprůměrovat.

7.6 Průsečík dvou přímek

```
#define primkaXprimka(useckatest1_bod1, useckatest1_bod2,  
useckatest2_bod1, useckatest2_bod2, dim, prusecik, parametr_t)
```

,kde jako *useckatest1_bod1*, *useckatest1_bod2*, *useckatest2_bod1*, *useckatest2_bod2* by se měl udávat typ float[3] a jde o vstupy

;jako *dim* by se mělo udávat číslo 2 nebo 3 a jde o vstup

;jako *prusecik* by se měl udávat typ float[4] a jde o výstup

;jako *parametr_t* by se měl udávat typ float[2] a jde o výstup

Makro vypočítá průsečík *prusecik* dvou přímek v dim- rozměrném prostoru. První přímka je dána body *useckatest1_bod1* a *useckatest1_bod2*, a druhá je dána body *useckatest2_bod1* a *useckatest2_bod2*. Aby mohlo být makro modifikováno na průsečík s polopřímkami nebo úsečkami, je potřeba dále ukládat parametry zjištěné na přímkách do proměnné *parametr_t*.

K vyhledání průsečíku dvou přímek je použita stejná soustava rovnic (27) jako v kapitole 7.5, ale tentokrát jsou neznámými pouze parametry *t* a *s* těchto přímek, a známými jsou jak jejich posunutí, tak vektory:

$$\overline{POSUN}_p + \overline{VEKTOR}_p \cdot t = \overline{POSUN}_q + \overline{VEKTOR}_q \cdot s \quad (27)$$

Rovnoběžnost a totožnost je testována porovnáním jednotkových vektorů vektoru \overline{VEKTOR}_p a \overline{VEKTOR}_q na začátku makra. Protože jde o soustavu tří rovnic o dvou neznámých, je třeba vybrat takové dvě rovnice, u kterých vzniklé jmenovatele ve výrazech nejsou rovny 0. K tomu dojde v případě, že nemá rovnice právě jedno řešení. Jsou testovány všechny kombinace dimenzí, a pokud nejsou ani po tom nalezeny správné rovnice, jsou přímky vyhodnoceny jako mimoběžky. V opačném případě je vypočítán průsečík a třetí rovnice je použita k provedení zkoušky.

Proměnná *prusecik* je čtyřrozměrná. První tři prvky jsou jeho prostorové souřadnice a poslední je příznak platnosti průsečíku. V případě rovnoběžnosti, totožnosti, nebo mimoběžnosti přímek je tento průsečík nastavován na hodnoty -1 nebo 1. V případě správného nalezení průsečíku je tento příznak nastaven do 0.

7.7 Průsečík přímky a úsečky

```
#define primkaXusecka2(primkatest1_bod1, primkatest1_bod2,  
useckatest2_bod1, useckatest2_bod2, dim, prusecik, param_uloz)
```

,kde jako *useckatest1_bod1*, *useckatest1_bod2*, *useckatest2_bod1*, *useckatest2_bod2* by se měl udávat typ float[3] a jde o vstupy

;jako *dim* by se mělo udávat číslo 2 nebo 3 a jde o vstup

;jako *prusecik* by se měl udávat typ float[4] a jde o výstup

;jako *param_uloz* by se měl udávat typ float[2] a jde o výstup

Makro používá makro `primkaXprimka`, a omezuje parametr druhé přímky. Pokud parametr nenáleží do rozsahu $\langle 0, 1 \rangle$, bod je na své přímce, ale ne na úsečce, a je vyhodnocen jako neplatný.

7.8 Obecná rovnice roviny

```
#define obecnarovnice_rovina(upravenebody, normalovy_vektor,  
posunuti, dim)
```

,kde jako *upravenebody* by se měl udávat typ float[3][3] a jde o vstup

;jako *normalovy_vektor* by se měl udávat typ float[3] a jde o výstup

;jako *posunuti* by se měl udávat typ float a jde o výstup

;jako *dim* se udává číslo 3

Makro vypočítá ze tří bodů *upravenebody* rovinu danou normálovým vektorem *normalovy_vektor* a posunutím *posunuti*.

7.9 Průsečnice dvou rovin

```
#define rovinaXrovina(normalovy_vektor1, posunuti1,  
normalovy_vektor2, posunuti2, prusecnice_bod,  
prusecnice_vektor)
```

,kde jako *normalovy_vektor1* a *normalovy_vektor2* by se měl udávat typ float[3] a jde o vstupy

;jako *posunuti1* a *posunuti2* by se měl udávat typ float a jde o vstupy

;jako *prusecnice_bod* by se měl udávat typ float[4] a jde o výstup

;jako *prusecnice_vektor* by se měl udávat typ float[3] a jde o výstup

Makro vypočítá z roviny dané normálovým vektorem a posunutím *normalovy_vektor1* a *posunuti1* a roviny dané normálovým vektorem a posunutím *normalovy_vektor2* a *posunuti2* parametrickou rovnici jejich průsečnice danou směrovým vektorem *prusecnice_vektor* a bodem *prusecnice_bod*. Součástí makra je kontrola rovnoběžnosti těchto dvou rovin. Při rovnoběžnosti je proměnná *prusecnice_bod*[3] nastavena do 1, jinak je rovna 0.

7.10 Bod na rovině nejbližší k danému bodu

```
#define bodNrovina(bodpozice55, normala559, posunuti559,
```

nejblizsibod55)

,kde jako *bodpozice55* by se měl udávat typ float[3] a jde o vstup

;jako *normala559* by se měl udávat typ float[3] a jde o vstup

;jako *posunuti559* by se měl udávat typ float a jde o vstup

;jako *nejblizsibod55* by se měl udávat typ float[3] a jde o výstup

Makro vypočítá na rovině dané normálovým vektorem normálovým vektorem a posunutím *normala559* a *posunuti559* nejbližší bod *nejblizsibod55* k bodu *bodpozice55*.

7.11 Test bodu uvnitř trojúhelníku

```
#define bodXtrojuhelnik(bodtest666, vrcholy666, normala666,
posunuti666, jeuvnitr666)
```

,kde jako *bodtest666* by se měl udávat typ float[3] a jde o vstup

;jako *vrcholy666* by se měl udávat typ float[3][3] a jde o vstup

;jako *normala666* by se měl udávat typ float[3] a jde o vstup

;jako *posunuti666* by se měl udávat typ float a jde o vstup

;jako *jeuvnitr666* by se měl udávat typ int a jde o výstup

Makro zjistí, jestli je bod *bodtest666* uvnitř trojúhelníku daného body *vrcholy666*. Kvůli zrychlení programu je tomuto makru rovnou předávána dříve vypočítaná rovina daná normálovým vektorem a posunutím *normalovy_vektor666* a *posunuti666*. V případě, že bod neleží v této rovině, je příznak *jeuvnitr666* roven -100. V případě, že bod je uvnitř trojúhelníku, je příznak *jeuvnitr666* roven 1. Jinak je roven 0.

Bod je zjišťován následujícím postupem:

1. Test, jestli je bod na rovině trojúhelníku.
2. Zvolení bodu, který musí být mimo trojúhelník
3. Spojení bodu mimo a testovaného bodu úsečkou.
4. Pro všechny úsečky trojúhelníku je testováno protnutí s touto úsečkou
5. Pokud je počet protnutí liché, bod je uvnitř trojúhelníku. Pokud je sudý, je mimo.

ad 1) Souřadnice bodu jsou dosazeny do obecné rovnice roviny, a výsledek porovnán s nulou. Protože je dovolena určitá nepřesnost, je použito porovnávání s tolerancí. Pokud výsledek není nula, bod neleží na rovině, příznak je nastaven na -100, a makro končí.

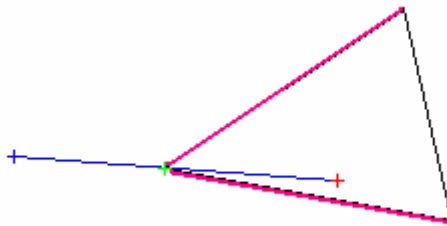
ad 2) Souřadnice tohoto bodu jsou zvolena jako dostatečně velká čísla. Jedno z nich je pak dopočítáno tak, aby bod ležel na dané rovině. Obecná rovnice roviny je:

$$ax + by + cz + d = 0 \tag{28}$$

,kde x,y,z jsou souřadnice bodu; a,b,c je normálový vektor a d je posunutí. Který rozměr bude dopočítán, je rozhodnuto tím, která první složka normálového vektoru není rovna nule. Kdyby byla rovna nule, došlo by k dělení nulou.

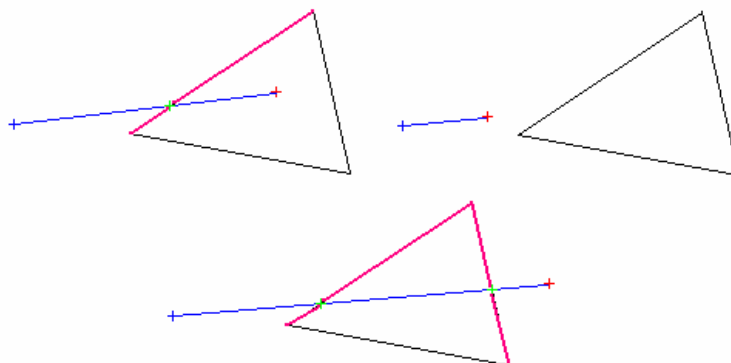
ad 4) K testování průsečíků jednotlivých úseček je použito makro *useckaXusecka3*, které je podobnou modifikací makra *primkaXprimka* v kapitole 7.6, jako makro *primkaXusecka2* popsané v kapitole 7.7. Místo jednoho parametru jsou ale omezovány oba, a tím je i z druhé přímky udělána úsečka. Příklad, kdy by makro nefungovalo, je na Obr. 36. Jde o krajní body obou fialově vyznačených úseček. Při průchodu přes tyto body mají vždy obě z úseček parametr 0 nebo 1. Problém je vyřešen tak, že je podmínka omezení upravena z

```
if(!((parametr_t[1]<=1)&(parametr_t[1]>=0))) {prusecik[3]=1;} \
if(!((parametr_t[0]<=1)&(parametr_t[0]>=0))) {prusecik[3]=1;} \
na
if(!((parametr_t[1]<=1)&(parametr_t[1]>=0))) {prusecik[3]=1;} \
if(!((parametr_t[0]<1)&(parametr_t[0]>0))) {prusecik[3]=1;} \
```



Obr. 36 Chyba vyhodnocení bodu v trojúhelníku

Na Obr. 37 je vidět důkaz funkčnosti tohoto postupu pro všechny možné případy, které mohou nastat. Tento postup se dá použít po úpravě i na kterýkoli n -stranný polygon, včetně polygonů s dírami. V této práci je model tvořen trojúhelníky, takže tato vlastnost je nevyužita.



Obr. 37 Princip vyhodnocení bodu v trojúhelníku

8 REKONSTRUKCE OBJEKTU

8.1 Zvolený datový formát

Z nasnímaných bitmap je třeba rozpoznat body převážně za použití obrazových transformací. Jedno měření má pro n postupných kroků šachovnice $4n+2$ bitmap. Z těchto dat budou výstupem 2 bitmapy s vyznačenými body (pro každou kameru zvlášť), ze kterých bude program pro 3D rekonstrukci číst. Tento způsob ukládání výsledku byl zvolen pro snadnou kontrolu jeho správnosti. Jinak by byl každý identifikovaný bod nacházející se na snímcích z jedné i druhé kamery uložen pouze do textového souboru ve formátu nejvhodnějším pro zcela odladěný program:

[ID bodu, x_1 , y_1 , x_2 , y_2]

,kde ID je identifikační číslo bodu jedinečné v rámci měření pro jeden úhel; x_1 a y_1 jsou pixelové souřadnice bodu na první kameře; x_2 a y_2 jsou jeho pixelové souřadnice na druhé kameře. Oddělovací znak jednotlivých proměnných by byl konec řádku ("Enter"). Pokud by bylo spoléháno na neporušenost souboru, stačilo by číst proměnné po pěti a automaticky tak poznat, o kterou proměnnou jde.

Zvolený způsob ukládání zkopíruje snímek objektu z nejvyšším rozlišením nasvícení, a přímo do něj nakreslí identifikované body jako pixely s barevnou hodnotou [ID, 0, 0]. Tím, že je obraz šedotónový, je umožněno číst takto označené pixely podmínkou

```
if ( ( znak1 != 0 ) & ( znak2 == 0 ) & ( znak3 == 0 ) )
```

,kde čtený pixel má hodnotu [znak1, znak2, znak3]. Je zřejmé, že tento způsob ukládání umožňuje jen 255 bodů a ID musí být indexováno od 1. Počet 255 nikdy nebyla překročen ani v nejvyšším rozlišení šachovnice. V případě, že by k tomu docházelo, by mohla být využita modrá barevná složka pixelu, a maximální počet tak zvýšen na dostatečných 255^2 bodů. Stejně jako se červená složka užívá při určení bodů, je zelená složka užita při určení osy otáčení - viz. kapitola 8.4.2.

8.2 Rozpoznání bodů

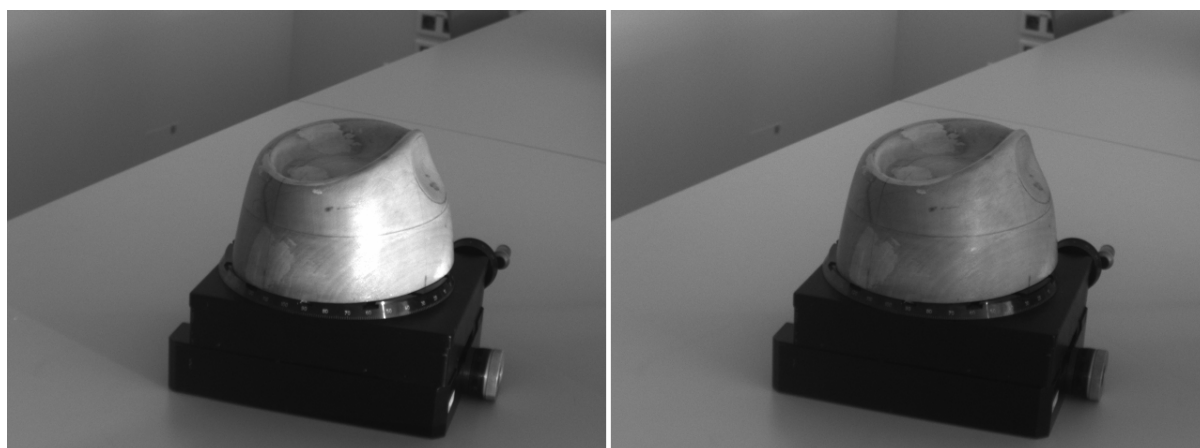
Pro rozpoznání i identifikaci bodů byla napsán software v jazyce C++ používající knihovnu OpenCV.

Při rozpoznání bodů je využit snímek zcela nasvíceného a zcela nenasvíceného objektu (Obr. 38 vlevo a vpravo). Dále je porovnáván snímek objektu nasvíceného šachovnicí (Obr. 39 vlevo) s jedním z těchto dvou obrazů. Jde o prahování se dvěma prahy, jehož výsledek nabývá tří hodnot - viz. vzorec (11). K rozpoznání bodů se používá pouze snímek nasnímaný s nejvyšším a druhým nejvyšším rozlišením vzoru. Nižší rozlišení slouží pouze ke vzájemné

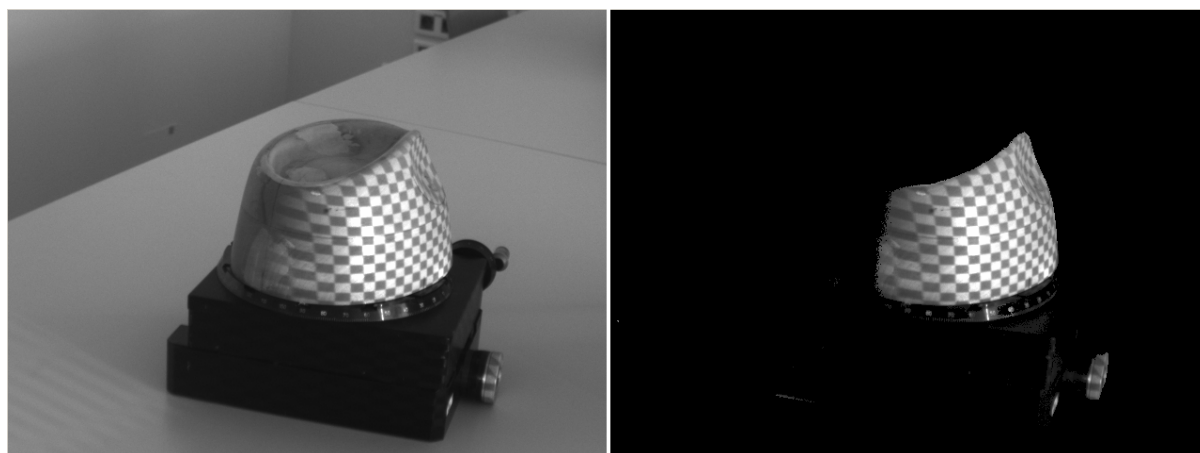
identifikaci bodů.

Na Obr. 39 vpravo je vidět, jak dobře může diferenční metoda eliminovat pozadí objektu. Na Obr. 40 je pak vidět obraz difference po aplikaci prahu. Protože formáty OpenCV omezují hodnoty typu `IplImage` od 0 do 255, a nebylo by tím možné vyhodnotit zápornou diferenci, muselo být pro uložení diferencí použito dvourozměrné pole proměnných typu `int`. Podobný problém nastal později u konvoluce, kde bylo použito dvourozměrné pole typu `float`.

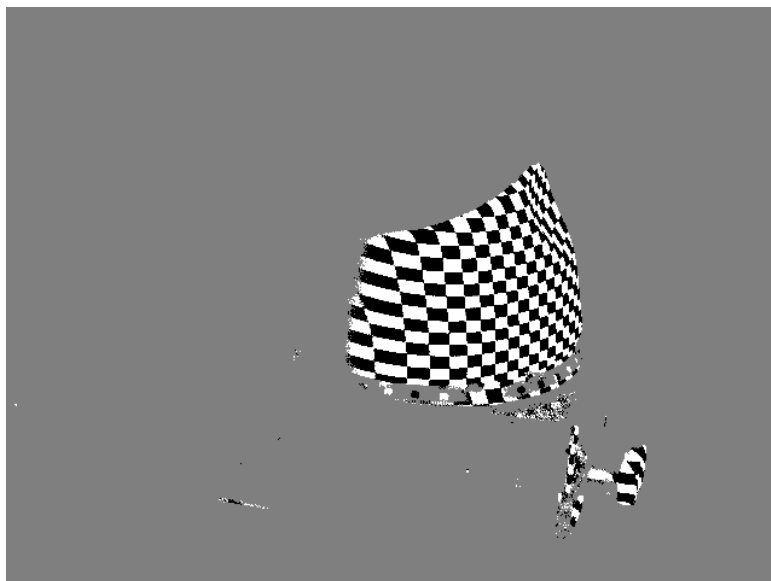
Na Obr. 41 je vidět obraz po aplikaci hranového detektoru a následném prahování šedé. Šedá se musí prahovat z důvodu, že významné jsou body šachovnice, ne pozadí. Pokud by byl ale vyhodnocován i obrys objektu, pak by šedá místa dostala význam hranic objektu. Konkrétně na obrázku je vertikální složka hranového detektoru.



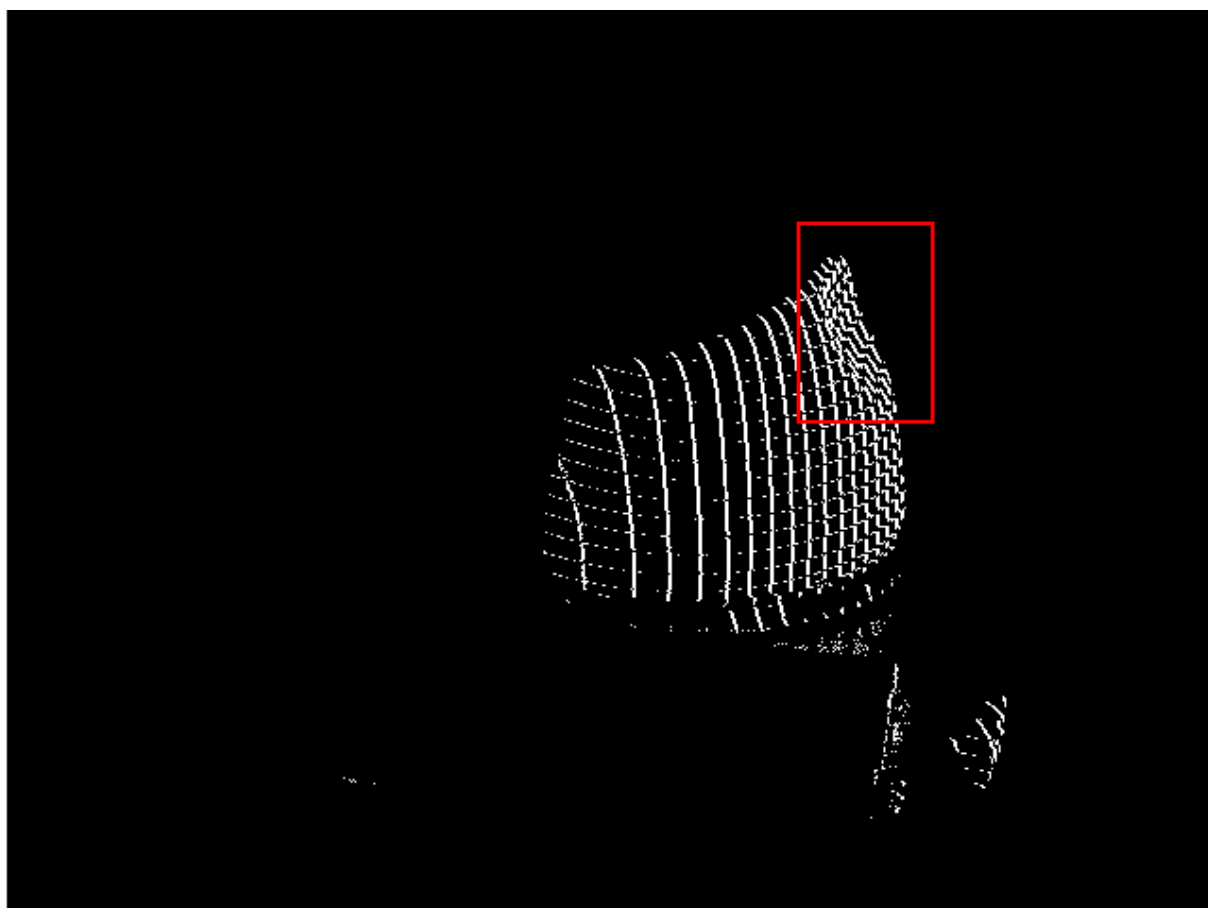
Obr. 38 Snímek nasvíceného a nenasvíceného objektu



Obr. 39 Snímky objektu nasvíceného šachovnicí a difference nasvícení



Obr. 40 Obraz po aplikaci prahování



Obr. 41 Obraz po aplikaci hranového detektoru

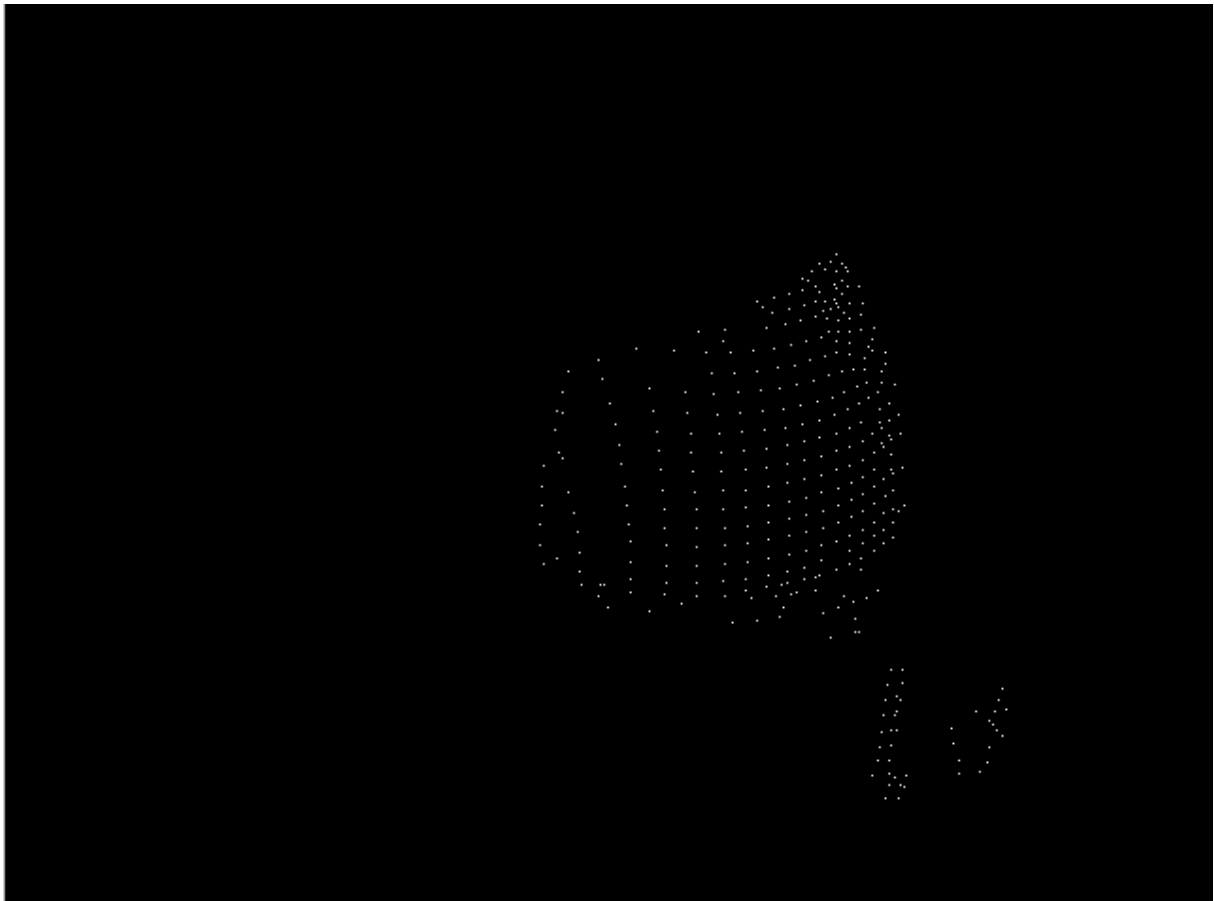
Obr. 41 je záměrně zvětšen, aby byla patrná červeně vyznačená část. Jde o to, že v této části dochází ke ztrátě informace, protože rozlišení kamer je příliš malé na to, aby mohl být v tomto místě jednoznačně identifikován bod. Řešením tohoto problému by mohlo být zvětšení

rozlišení tohoto obrazu před aplikací hranového detektoru.

Dále musí být vertikální i horizontální složka obrazů hran složena do jednoho obrazu. Složky jsou skládány podle vzorce (15). Na výsledném obrazu jsou nalezeny body pomocí skeletonizace a aplikace konvoluční masky:

$$Konv_4 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

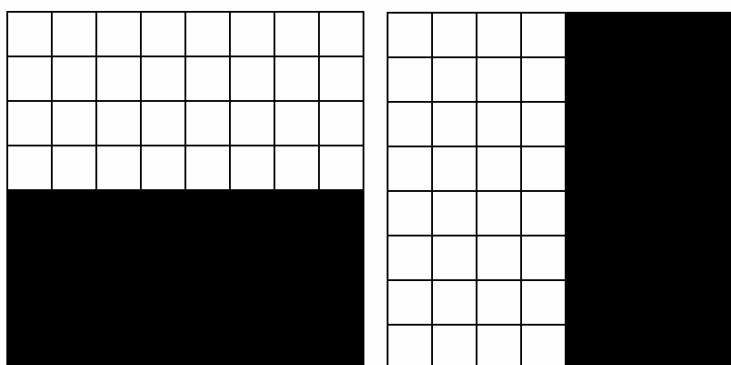
Výsledný obraz je na Obr. 42. Je vidět, že stále dochází ke ztrátě některých bodů. To už ale řeší vzájemná identifikace - body chybějící na obraze jedné z kamer nejsou uvažovány ve výsledné rekonstrukci. Body nepatřící k objektu mohou být z měření vyloučeny pomocí segmentace snímku. Je jasné, že největší oblast, která není šedá, je objekt. Zbytek by byl označen jako pozadí již před aplikací hranového detektoru, skeletonizace a konvoluční masky.



Obr. 42 Rozpoznané body objektu

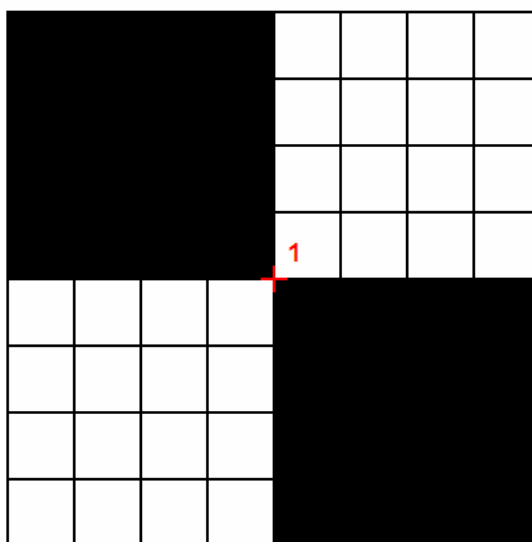
8.3 Vzájemná identifikace bodů

Identifikace bodů se provádí odděleně pro snímky jedné i druhé kamery. Je vysvětlena na obrázcích šachovnice s mřížkou, i když v této práci byla použita šachovnice bez mřížky. Mřížka je zde zobrazena pouze pro přehlednost. Na Obr. 43 vlevo a vpravo je první krok měření. Tyto obrázky samy v sobě nemají žádné body, ale mají význam pro pozdější identifikaci.



Obr. 43 Kód pro identifikaci bodů

Při dalším kroku mřížky je změřen jediný bod (Obr. 44). Ten je na příslušném obraze rozpoznán a je vyhledán bod na obraze s nejvyšším rozlišením, u kterého se shodují pixelové souřadnice se souřadnicemi tohoto bodu. Je označen číslem 1.



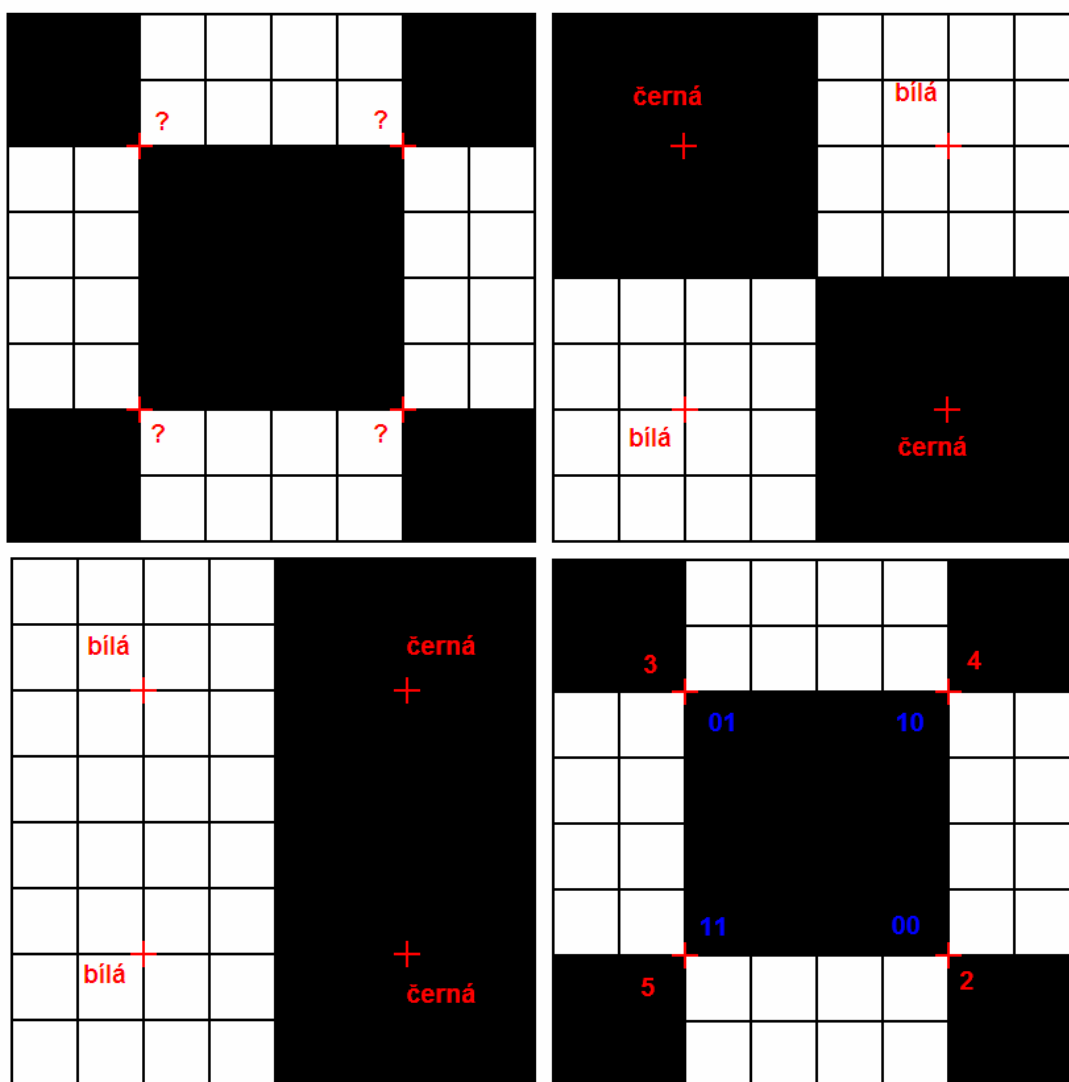
Obr. 44 První krok identifikace bodů

V dalším kroku jsou ale identifikovány už 4 nové body (Obr. 45 vlevo nahoře). Pro jejich identifikaci se porovnává barva předchozího snímku v místě pixelu (Obr. 45 vpravo nahoře). Na obrázku je vidět, že dva body jsou identifikovány v bílých polích šachovnice a dva v černých. Kód stále není unikátní, takže se pokračuje ve snímku s ještě nižším rozlišením (Obr. 45 vlevo dole). Je patrné, že nyní už je kód unikátní, a proto může být jednotlivým bodům

přiřazeno jedinečné ID a můžou být pod ním identifikovány. Na Obr. 45 vpravo dole je modře napsán kód každého bodu, kde první binární hodnota je brána z obrázku s nižším rozlišením o jeden krok a druhá z obrázku o dva kroky, a logická 0 reprezentuje černé pole a logická 1 bílé. Červeně je pak napsáno jedinečné ID.

Je třeba ošetřit, aby ID bylo přiřazováno na obou obrazech stejně. Když na obraze z jedné kamery bude například chybět bod 2, bude tímto způsobem celá identifikace posunuta o počet chybějících bodů. Proto je třeba vždy porovnávat, které body se objeví na obou kamerách. Ty, které se objeví jen na jedné kameře, budou z identifikace úplně vyřazeny.

Tento postup se opakuje pro všechna rozlišení šachovnice. Je zřejmé, že pro unikátnost kódu bude při vyšších rozlišeních muset být porovnáváno více snímků. Navíc nesmí být pro identifikaci používány snímky se sudým rozdílem kroku rozlišení od analyzovaného snímku. Když bude například analyzován snímek označený rozlišením 6, budou pro identifikaci použity snímky 5, 3, a 1. Snímky 2 a 4 mají stejné body přesně na rozhraních polí, a tím pádem by v takovém místě byla identifikace k ničemu.



Obr. 45 Druhý krok identifikace bodů

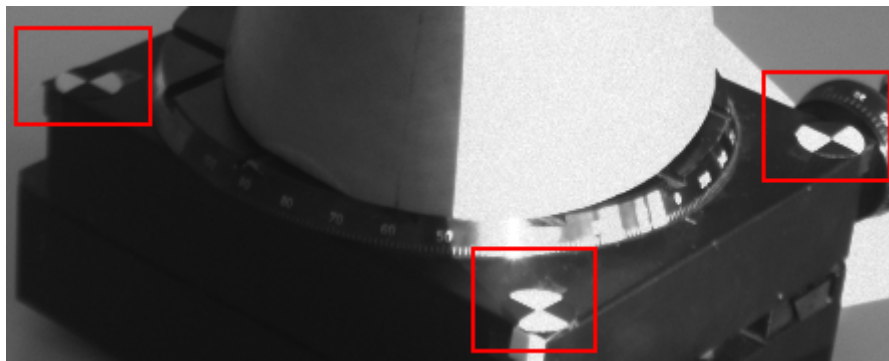
8.4 Transformace bodů

8.4.1 Transformace v rámci jednoho úhlu

Prakticky jde o postup obrácený jako u kalibrace. Jsou známy směrové vektory obou kamer, jejich roztažení zorného pole a 3D vektory odpovídající pixelovým osám x a y . Dále jsou spárovány souřadnice 2D bodu identifikované na jedné i druhé kameře. Souřadnice 3D bodu pak jsou počítány podle rovnice (22) v kapitole 5.5.

8.4.2 Zjištění osy natočení

U každého měření je znám úhel, o který je třeba body modelu tohoto úhlu natočit kolem osy rotace. Ta musí být zjištěna podle značek umístěných na podstavci, nebo v nejhorším případě podle značek vytvořených ručně na obou obrazech. Na postavec známých rozměrů 18.9 x 22.1 cm byly na rohy přilepeny terčíky o poloměru 1.2 cm (viz. Obr. 46). Je třeba vždy provádět měření tak, aby byly tři terčíky na snímku obou kamer.



Obr. 46 Terčíky ke snímání osy otáčení

Protože jsou známy vzdálenosti terčíků od sebe, může být při jejich použití prováděna i kalibrace kamer v jednom kroku s měřením. Tato možnost ale nebyla realizována.

Pixelové pozice terčíků mohou být z obrazu identifikovány tak, že by byly pořízeny snímky objektu, a zjišťováno, které detekované body se nemění. Kvůli tomu, že terčíky jsou šachovnice, by bylo možné je detekovat hranovým detektorem nebo konvoluční maskou. Vzájemná identifikace je provedena jednoduše pomocí jejich x -ové pixelové souřadnice.

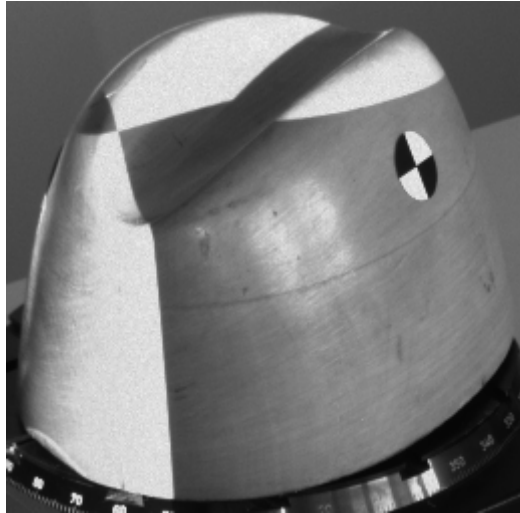
Je zjištěna obecná rovnice roviny daná těmito body transformovanými do 3D. Vektor této roviny je současně hledaným směrovým vektorem osy otáčení. Bod této osy je zase dán středním bodem úsečky mezi dvěma body na úhlopříčce podstavce, které budou mít díky způsobu jejich vzájemné identifikace ID vždy 0 a 2.

8.4.3 Otočení bodu podle osy

K rotaci bodu kolem osy v daném úhlu je použita zjednodušená transformační matice pro rotaci (viz. kapitola 4.1.3). Úhel je pevně zadán při měření a nastavován na podstavci ručně.

Kvůli časovým důvodům nebyla nijak řešena vůle trapézového šroubu na podstavci, a proto může být skutečný úhel $\pm 0.5^\circ$. K částečnému vyřešení tohoto problému by mohly sloužit terčíky nalepené přímo na objekt (viz. Obr. 47) tak, aby byl aspoň jeden z nich v záběru obou kamer na dvou po sobě jdoucích snímcích.

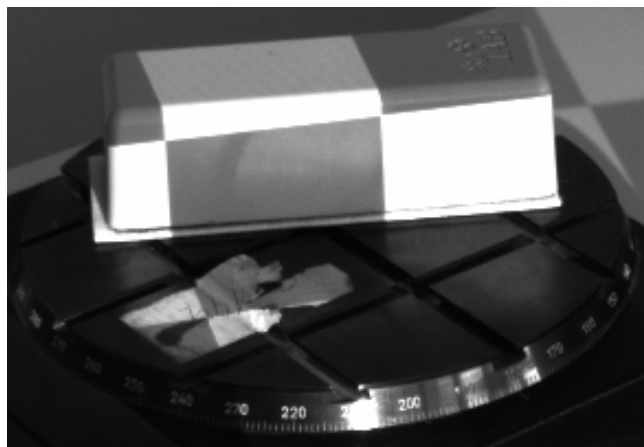
Pak by byl vyhodnocován rozdíl úhlů na prvním a druhém snímku, a pomocí tohoto rozdílu by byla skutečná hodnota rotace upravena.



Obr. 47 Terčíky k vyhodnocení posunutí osy objektu

8.4.4 Problém posunutí osy objektu

Jak je vidět na Obr. 48, osa objektu je vždy jinde než osa otáčení podstavce. K eliminaci problému je možno použít opět stejné značky jako na Obr. 47. Tentokrát by ale nebyl vyhodnocován rozdíl jejich úhlu od osy rotace, ale rozdíl jejich vzdáleností od této osy. Značky by bylo třeba umístit kolem celého objektu, aby bylo zjištěno, na jakém úhlu má vzdálenost od osy maximum a minimum. Pak by bylo při 3D rekonstrukci před rotací provedeno ještě posunutí objektu o tuto vzdálenost na tomto úhlu.



Obr. 48 Posunutí osy objektu

8.5 Zobrazení 3D modelu

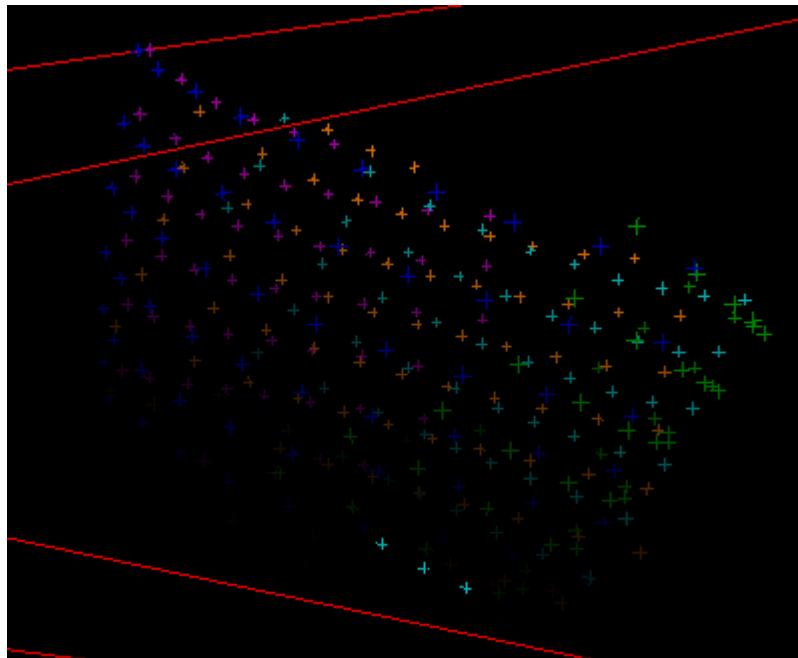
Pro zobrazování 3D modelu byl napsán software v jazyce C++ využívající knihovny OpenGL, SDL, a OpenCV. OpenCV je využita jen při čtení vstupních obrázků. Pro samotné zobrazení dat je využito knihoven OpenGL a SDL. Mohou v něm být zobrazeny pozice obou kamer a jejich směrové vektory a zorná pole, body objektu, 3D model objektu, jeho osa otáčení, těžiště, a kalibrační tabule. Je možné se v něm pohybovat, aby bylo možno snadno kontrolovat výsledek rekonstrukce ze všech úhlů. Otáčení je řešeno pomocí myši a ovládání klávesnice je:

[Esc] Ukončit

[W,S] Dopředu, dozadu

[A,D] Vlevo vpravo

Na Obr. 49 je snímek z tohoto programu, kde jsou vidět body rekonstruované z měření 5 různých úhlů natočení objektu označené různými barvami.



Obr. 49 Program pro vizualizaci dat

8.6 Algoritmus spojování bodů

Z transformovaných 3D bodů byly vytvořeny dílčí modely pro jednotlivé naměřené úhly. Nyní bude popsán algoritmus spojení těchto bodů do sítě trojúhelníků. Postup se dělí na vytvoření úseček a vytvoření polygonů. Jeho jednoduchost je dosažena transformací 3D souřadnic bodů na dva parametry. Při tvorbě dílčích modelů měření jednotlivých úhlů jsou tyto parametry pixelové souřadnice bodů na jedné z kamer, a není třeba nic transformovat. Další možnost je transformace na polární souřadnice a následné použití obou úhlů jako parametrů, což je popsáno v kapitole 8.7.2. Postup vytvoření úseček je následující:

1. Nalezení dvojice bodů, které mají od sebe nejmenší vzdálenost v rámci celé množiny vyhodnocovaných bodů.

2. Test, jestli se úsečka mezi těmito body neprotíná s jinou již dříve vytvořenou úsečkou, nebo není totožná s jinou již dříve vytvořenou úsečkou.

3. V případě, že výsledek testu 2 je záporný, je vytvořena nová úsečka mezi těmito dvěma body.

4. Postup je opakován, dokud jsou nacházeny nové úsečky

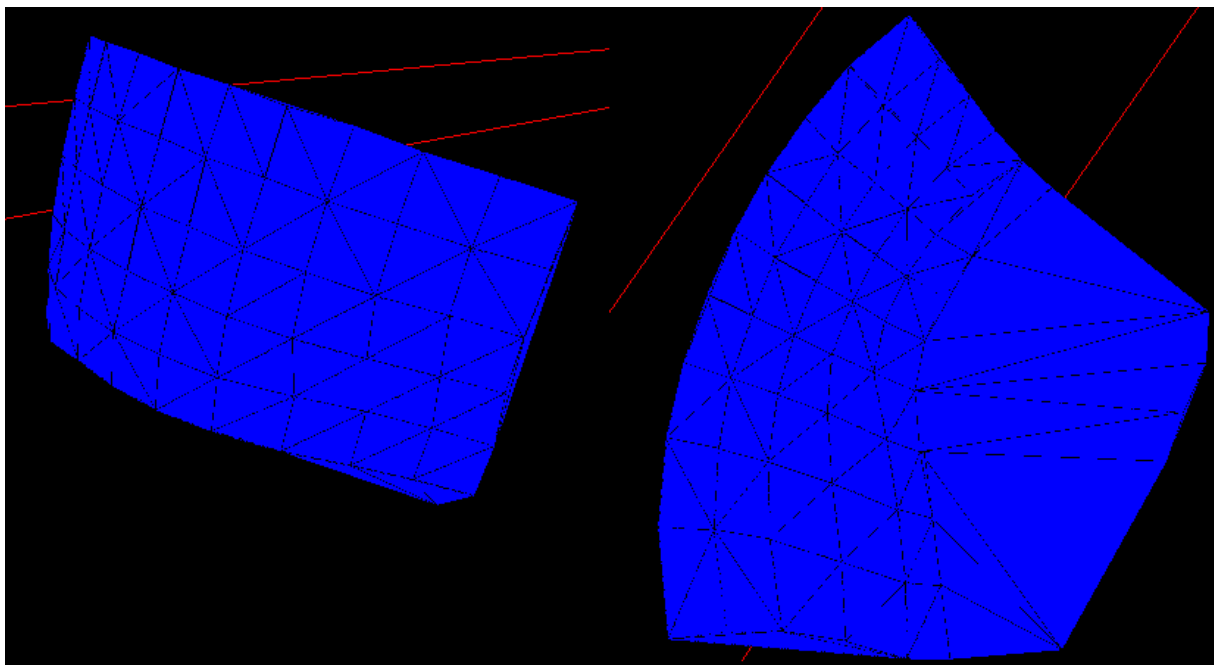
Když jsou vytvořeny všechny úsečky, musí být mezi jejich sítí vytvořeny polygony. Postup je:

1. Procházet všechny body a pro každý bod zjistit, jaké úsečky do něj směřují.

2. Protože jde o trojúhelníky, je z těchto hledána taková dvojice úseček, jejíž koncové body mezi sebou mají úsečku. Žádná z těchto tří úseček nesmí mít příznak průchodu 2 (vysvětleno níže).

3. Když jsou tyto tři úsečky nalezeny, je u každé úsečky zvýšen příznak průchodů. Je zřejmé, že každá úsečka může být členem pouze dvou trojúhelníků najednou. Proto nejsou úsečky, které dosáhnou dvou průchodů, dále vyhodnocovány.

4. Po průchodu všech bodů je algoritmus ukončen. Úsečky s příznakem průchodu 1 jsou označeny jako okrajové, což může být využito v některých metodách spojování modelu.

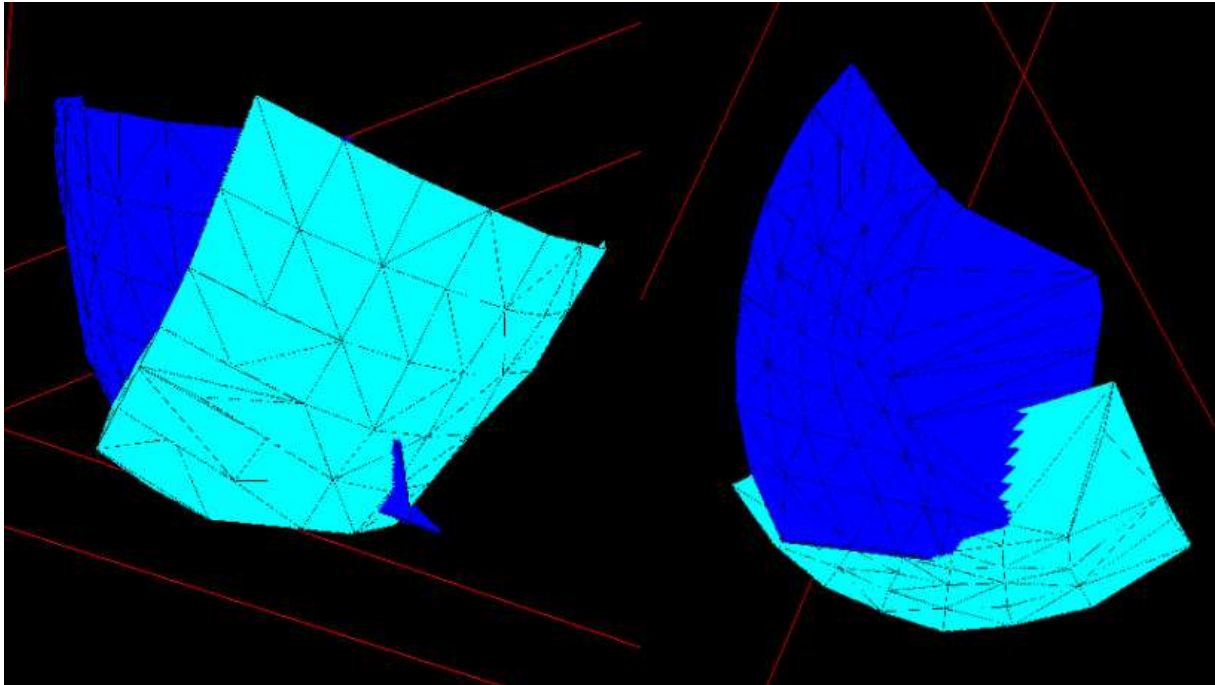


Obr. 50 3D model jednoho úhlu měření

Na Obr. 50 je vidět pospojovaný model pro měření jednoho úhlu. Je v něm patrná trojúhelníková síť.

8.7 Metody spojení 3D modelu

Navržené metody pro spojení výsledného 3D modelu se dělí na dvě hlavní skupiny - metody využívající dílčí modely, a metody počítající model celého 3D objektu pouze z bodů. Na obrázcích Obr. 51 je vidět, jak se měření jednotlivých úhlů překrývají.



Obr. 51 Překrytí měření dvou různých úhlů

Tento problém je třeba vyřešit některou z následujících metod. Jako první budou zvlášť popsány kroky, které jsou pro více metod společné.

8.7.1 Výpočet středu objektu

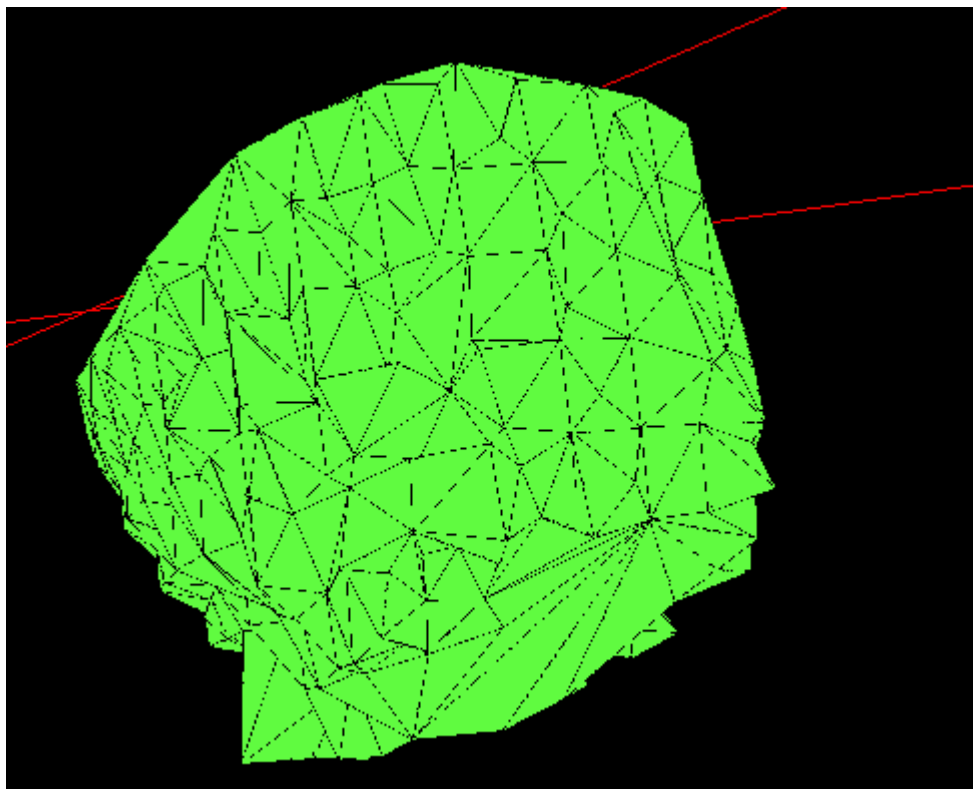
Bod, který je vypočítán, je pouze přibližným středem objektu. Při nasnímání z více rovnoměrně rozmístěných úhlů ale stačí. Jednoduše jsou sečteny všechny 3D souřadnice bodů a vyděleny počtem bodů. Střed objektu je důležitý pro většinu použitých metod.

8.7.2 Metoda polárních souřadnic

V této metodě jsou kartézské souřadnice všech 3D bodů převedeny na polární podle těchto vzorců získaných ze zdroje [7]:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arccos\left(\frac{z}{r}\right) \\ \varphi &= \arctan\left(\frac{y}{x}\right) \end{aligned} \quad (29)$$

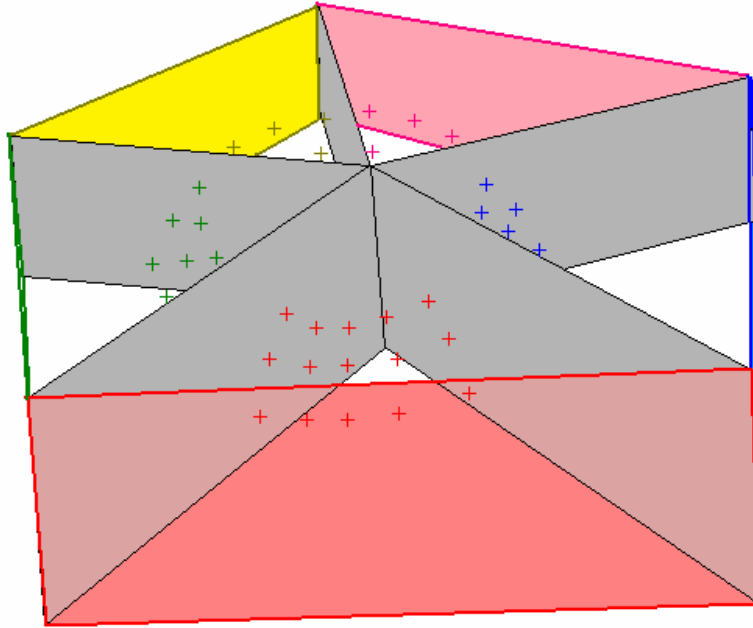
,kde 3D bod je dán kartézskými souřadnicemi $[x, y, z]$; r je poloměr; θ je inklinací úhel a φ je azimut. Úhly θ a φ jsou použity jako parametry algoritmu pro spojování bodů. Navíc na rozmezí 360° u obou úhlů nebudou trojúhelníky vytvořeny, což musí být vyřešeno dodatečným průběhem okrajových úseček a jejich bodů. Pro měření pouhých 5 úhlů bylo změřeno kolem 300 bodů, které je třeba navzájem pospojovat. Částečně proto je metoda pomalá.



Obr. 52 Celkový 3D model objektu

8.7.3 Metoda projekce na roviny

Body objektu jsou rozděleny na počet podmnožin stejný, jako byl počet měření. Jsou do jednotlivých podmnožin přiřazovány na základě rozdělení objektu podle osy otáčení viz. Obr. 53.



Obr. 53 Princip metody projekce na roviny

Tyto podmnožiny bodů jsou označeny různými barvami. Šedé polygony jsou dělicí a vymezují prostor, kam zasahují jednotlivé podmnožiny. Barevné polygony jsou projekční roviny, na které jsou odpovídající body promítnuty pomocí nalezení nejbližšího bodu na tuto rovinu. Po výpočtu obou směrových vektorů roviny z normálového dostáváme parametrickou rovnici roviny:

$$\overline{BOD} = \overline{BOD}_R + \overline{SMER1}_R \cdot t + \overline{SMER2}_R \cdot s \quad (30)$$

,kde BOD je bod náležící rovině; BOD_R je její posunutí; a SMER1_R a SMER2_R jsou její směrové vektory. Protože jde o 3D, dostáváme pro každý bod soustavu tří rovnic o dvou neznámých, z níž jsou pro každý bod vypočítány parametry t a s . Ty jsou použity jako parametry u algoritmu pro spojování bodů. Metoda eliminuje překryvy bodů jednotlivých měření a nemusí se složitě řešit průsečíky trojúhelníků. Na rozmezí rovin nebudou trojúhelníky vytvořeny, což musí opět být vyřešeno dodatečným průběhem okrajových úseček a jejich bodů.

Tentokrát je ale tento dodatečný průběh o hodně složitější - musí být vybrány projekční roviny na rozmezí předchozích rovin, a spojovat jen takové body, které jsou na jednom rozmezí. Další metoda je přiřadit jednotlivým rovinám rozsahy a přičíst k jednomu z parametrů takové číslo, aby se rozsahy nepřekrývaly. Algoritmus spojování by se pak provedl pouze jednou, a to pro všechny body. Rozmezí by mohlo být spojeno kombinací s předchozí metodou.

8.7.4 Spojení podle 3D vzdálenosti

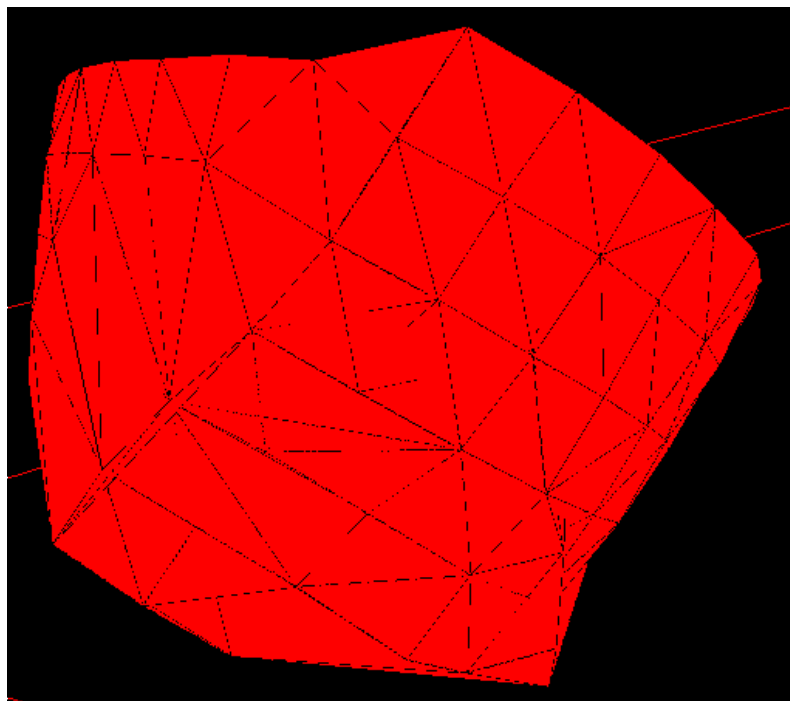
Na začátku této kapitoly je třeba si vysvětlit důvod, proč nemůže být použitý algoritmus pro spojování použit ve 3D. Algoritmus je totiž založen na hledání nejmenší vzdálenosti mezi

body, takže by byl správný jen jeho první a druhý krok provedený pro všechny body najednou. Další kroky by vytvářely ve skutečnosti neexistující trojúhelníky, a mezi body by vznikaly malé jehlany místo skutečného 3D modelu.

Podmínku správné funkce by bylo možné vymyslet jako test vzniku uzavřených těles na podobném principu, jako test bodu v polygonu viz. kapitola 7.11. Tím by se algoritmus značně zpomalil. Když je navíc uvaženo, že se provádí pro celou množinu naměřených bodů, je jeho časová náročnost nepřijatelná. Jde ale zatím o jediný algoritmus, který umožňuje správnou rekonstrukci 3D objektu s dírami.

8.7.5 Přitažení dvou dílčích modelů

Metoda je založena na zjištění bodů v místě překryvu několika měření, a jejich následném spojení do jednoho bodu zprůměrovaného z těchto tohoto bodu a bodu ze druhého dílčího modelu, který mu je nejbližší. Je vedena vždy přímka ze dříve vypočítaného středu objektu do analyzovaného bodu prvního modelu, a je testováno, jestli neprotnula trojúhelník druhého modelu. Průsečík je nalezen jako průsečík této přímky s rovinou trojúhelníku, a následně je testováno, jestli je uvnitř. Výsledek metody vypadá viz. Obr. 54. Je na ní třeba vyřešit mazání duplicitních trojúhelníků.

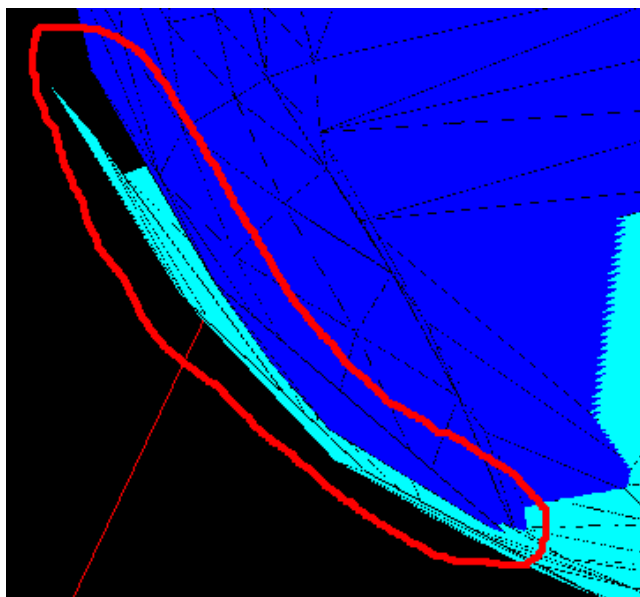


Obr. 54 Výsledek metody přitažení modelů

8.7.6 Řez dvou dílčích modelů

Metoda má nejsložitější implementaci z uvedených metod. Měla by fungovat tak, že se vypočítají všechny trojúhelníky, které protínají trojúhelníky druhého modelu, a být získána úsečka jejich protnutí. Tyto by měly být pospojovány a podle nich ořezán model. Metoda

nemůže nikdy fungovat z tohoto důvodu:



Obr. 55 Důkaz nefukčnosti metody řezu modelů

Na celém červeně vyznačeném rozsahu se nikdy neprotnou žádné trojúhelníky. Metoda by fungovala jen ve velice ideálních případech. Proto byla její implementace přerušena.

8.8 Další úkoly pro tvorbu modelů

Pokud by bylo v této práci pokračováno, měla by být práce zaměřena v první řadě na způsob eliminace chyby posunutím osy objektu od osy otáčení. Když jsou srovnány dílčí modely na Obr. 51 a kompletní model na Obr. 52, je vidět, že na místě překryvu je výsledek značně nevyhovující. Práce byla zaměřena v první řadě na funkčnost, a v návaznosti by měla být řešena přesnost.

Jako druhý úkol by byla zvolena správná detekce děr v objektu. Při prahování diferenčního obrazu již byl tento problém naznačen, viz. šedé pozadí na Obr. 40. Měl by být vysegmentován největší objekt na obraze, což bude objekt měřený. V něm by byly vyhodnoceny díry, kde se nemění hodnota nasvícení. 2D úsečky, které by takové pixely protínaly, by nebyly vytvořeny.

Na výsledném 3D modelu by do děr byly vedeny polopřímky z pozic kamer, a byl proveden výřez díry. Tato operace by měla velmi složitou implementaci, ale je to jediný možný způsob kromě kapitoly 8.7.4, jak změřit objekt s dírami.

9 ZÁVĚR

Úkoly práce byly tyto:

1) Nastudujte metody vytvoření (360 stupňů) 3D modelu objektu za pomoci nasvětlení vzorem.

2) Navrhněte měřicí stanoviště a zvolte vhodný vzor pro nasvícení.

3) Realizujte metody pro kalibraci systému.

4) Realizujte metody pro 3D rekonstrukci objektu. Pro uložení zvolte vhodný datový formát.

5) Vhodným způsobem vizualizujte výsledek rekonstrukce a zhodnoťte dosažené výsledky.

ad 1) Tento úkol je pouze teoretický a jeho splnění je doloženo kapitolami, které se zabývají používanými metodami.

ad 2) Protože byly k dispozici dvě kamery, byla pro měření zvolena pasivní metoda. Z důvodu, že není nutné snímat rychlé děje, byla jako vzor vybrána šachovnice s postupnou změnou rozlišení, které umožňuje snadnou vzájemnou identifikaci bodů a je málo náchylná na vnější vlivy.

ad 3) Kalibrace systému byla realizována. Ke kalibraci jsou používány trojice bodů kalibrační tabule, které jsou průměrovány. Pro přesnější kalibraci by bylo třeba použít metodu nejmenších čtverců.

ad 4) 3D rekonstrukce je založena na znalosti úhlu a měření osy otáčení objektu. Z Obr. 52 lze vidět, že rekonstrukce objektu byla úspěšná. Programové řešení rozpoznání bodů musí být upraveno.

ad 5) Objekt byl úspěšně zobrazen ve vlastním 3D enginu. Jako testovací objekt bylo použito kopyto na klobouk s obrysem 21·17 cm. Tyto délky byly měřením vyhodnoceny jako 21.92 · 15.9. Byl formulován závěr, že největší chyba je ve skládání modelů jednotlivých úhlů a je způsobena vlivem posunu osy otáčení objektu od osy otáčení na podstavci.

LITERATURA

- [1] BABINEC, T.. *Vyhodnocení tvaru objektů*. Bakalářská práce. Brno: FEKT VUT v Brně, rok neznámý.
- [2] Pasivní triangulace [online]. [cit. 6. ledna 2014]. Dostupné na www.midas.uamt.feec.vutbr.cz/APV/exercises-pdf/08_Pasivni_triangulace.pdf
- [3] Structured-light 3D scanner [online]. [cit. 18. dubna 2014] Dostupné na http://en.wikipedia.org/wiki/Structured-light_3D_scanner
- [4] Digital Light Processing [online]. [cit. 26. dubna 2014] Dostupné na http://cs.wikipedia.org/wiki/Digital_Light_Processing
- [5] Interferometry [online]. [cit. 26. dubna 2014] Dostupné na <http://www.answers.com/topic/interferometry>
- [6] Depth from focus / defocus [online]. [cit. 3. května 2014] Dostupné na http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdutorial.html
- [7] Spherical coordinate system [online]. [cit. 16. května 2014] Dostupné na http://en.wikipedia.org/wiki/Spherical_coordinate_system
- [8] White noise [online]. [cit. 17. května 2014] Dostupné na http://en.wikipedia.org/wiki/White_noise
- [9] HORÁK, K.. *Jasové transformace* [online]. [cit. 17. května 2014] Dostupné na http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php
- [10] HORÁK, K.. *Detekce hran a rohů* [online]. [cit. 17. května 2014] Dostupné na http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

3D	3 Dimensional, označení pro tří-dimenzionální prostor
2D	2 Dimensional, označení pro dvou-dimenzionální prostor
OpenGL	Open Graphics Library, standart definující multi-jazykové programovací rozhraní pro tvorbu počítačové grafiky
C++	programovací jazyk
Matlab	programovací jazyk pro výpočty a simulace, v této práci používaný hlavně v testovací fázi pro operace s obrázky
OpenCV	Open Computer Vision, knihovna pro počítačové vidění