

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UNIFORM MARKER FIELD NA VÁLCI

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. RADIM KŘÍŽ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UNIFORM MARKER FIELD NA VÁLCI

UNIFORM MARKER FIELD ON A CYLINDER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. RADIM KŘÍŽ

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2013

Abstrakt

Práce se zabývá rozšířením Uniform marker fields o možnost detekování na válci. V úvodu pojednává obecně o rozšířené realitě se zaměřením na systémy využívající markery. Diskutuje současné state-of-the-art systémy a jejich možnosti. Poté se podrobněji zabývá typem markerů Uniform Marker Field se zaměřením na jejich detekci. Zmiňuje i variantu pracující v odstínech šedi. Dále jsou v práci popsány některé zákonitosti projekce válce v reálném prostoru se zaměřením na vlastnosti, které lze využít k detekci markerů na plášti válce. Následuje návrh a popis konkrétní podoby detekčního algoritmu. Na závěr je implementace detektoru otestována a zhodnocena.

Abstract

This work presents a new extension for Uniform Marker Field, which is able to detect UMF on the cylinder. First part of the text deals with Augmented reality and focuses on systems using markers. It discusses the actual state-of-the-art systems and its possibilities. After that it focuses more deeply on the marker system Uniform marker field and its grayscale variants. Next part of the work describes properties of the cylinder projected in real space. Important properties for detecting are discussed in detail. Then the proposal and description of detection algorithm is presented. Implementation of algorithm is tested and evaluated on the very end of this thesis.

Klíčová slova

Uniform marker fields, Grayscale uniform marker fields, Rozšířená realita, Marker, Detekce merkeru, Válec, Detekce Elips, Detekce parabol, Detekce přímek, RANSAC, Metoda nejmenších čtverců, Hierarchické shlukování, Projektivní geometrie

Keywords

Uniform marker fields, Grayscale uniform marker fields, Augmented reality, Marker, Marker detection, Cylinder, Ellipse fitting, Parabola fitting, Line fitting, RANSAC, Least Square Error, Hierarchical clustering, Projective geometry

Citace

Radim Kříž: Uniform Marker Field na válci, diplomová práce, Brno, FIT VUT v Brně, 2013

Uniform Marker Field na válci

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana docenta Adama Herouta.

.....
Radim Kříž
22. května 2013

© Radim Kříž, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Rozšířená realita založená na markerech	5
2.1	Markery	7
2.2	Příklady markerů	8
3	Uniform marker fields	10
3.1	Detekce marker fields	11
3.2	Grayscale marker fields	13
4	Detekce Uniform marker fields na válci	14
4.1	Teoretické úvahy použité při vytváření detektoru UMF na válci	14
4.2	Použité algoritmy	20
5	Návrh a implementace detekčního algoritmu	23
5.1	Předpoklady a omezení pro vstupní data	24
5.2	Detekce hranových bodů na vzorkovací mřížce	24
5.3	Detekce skupin hranových bodů	25
5.4	Hledání parametrů přímek	26
5.5	Nalezení dvou hlavních skupin přímek a rozlišení mezi přímkami na plášti a prstenci	26
5.6	Nalezení úběžníku přímek na plášti	28
5.7	Shlukování podobných přímek na plášti	28
5.8	Detekce elips na plášti	29
5.9	Nalezení parametrů válce pomocí elips a algoritmu RANSAC	30
5.10	Hledání parametrů parabol na základě parametrů válce	31
5.11	Nalezení horizontu pomocí parabol a algoritmu RANSAC	33
5.12	Přepočet parametrů parabol na základě společného průsečíku na horizontu	34
5.13	Shlukování podobných parabol	35
5.14	Doplnění chybějících parabol	36
5.15	Vytvoření výsledné mřížky definující nalezené středy čtverců Uniform Marker Fields	37
6	Testování detektoru	39
6.1	Celkové testování	40
6.2	Testování vlivu úhlů pohledu na detekci	41
6.3	Testování počtu viditelných polí	44
6.4	Testování vlivu okolní scény	46

6.5	Testování rychlosti detektoru	47
6.6	Zhodnocení testů	48
7	Závěr	50
7.1	Možnosti budoucí práce	50
7.2	Celkové zhodnocení práce	50
A	Obsah CD	54
B	Manuál	56
B.1	Základní informace o programu	56
B.2	Potřebný software	56
B.3	Překlad	57
B.4	Spuštění	57
C	Konfigurační soubor	58
D	Ukázky testovacích dat	60
E	Plakát	63

Kapitola 1

Úvod

Rozšířená realita v dnešní době už není jen okrajové odvětví, kterým se zabývají pouze vědecké skupiny na univerzitách a technologičtí nadšenci. Stává se z ní regulérní obor využívající nejmodernější poznatky z hardwarové i ze softwarové oblasti výpočetních technologií.

Aby mohly systémy rozšířené reality korektně fungovat potřebují dostatek informací o okolním reálném světě. Tyto informace jsou zpravidla pozice objektu a umístění kamery ve scéně. Vstupní data jsou nejčastěji pouze ve formě obrazové informace z běžné kamery. Veškeré výpočty navíc musejí fungovat ideálně v reálném čase. Z uvedených podmínek je očividné, že vytvoření kvalitního systému představuje skutečnou výzvu.

Prozatím není známa metoda, která by byla uspokojivě schopná potřebné informace získat z prosté reálné scény. Proto si spousta řešení vypomáhá objekty se známými vlastnostmi, které jsou do scény umístěny. Tyto objekty se nazývají markery. Markery jsou zpravidla navrženy tak, aby umožňovali snadnou detekci v obraze a zároveň, aby bylo možno z nich vyčíst veškeré potřebné informace vyžadované konkrétním řešením.

Tato práce se zabývá Uniform marker fields (UMF) a detekcí tohoto typu markeru na neplanárním povrchu a to konkrétně na válci. Stávající detekční algoritmus pracuje pouze s planárními povrchy a neporadí si s výraznější deformací markeru, protože vychází z předpokladů a zákonitostí, které platí pouze pro planární strukturu. Představená detekce markeru na válci je patrně jedním z prvních pokusů o detekci markeru na neplanárním povrchu. Válec byl zvolen proto, že jeho tvar je po planární rovině druhý nejjednodušší. Ve směru kolmém na podstavě zakřivený není, čili je možné použít částečně upravený stávající postup. Ve směru kolmém k podstavě je zakřivení pravidelné. Právě uchopením tohoto zakřivení se zabývá rozšíření algoritmu.

Druhá kapitola obecně představuje rozšířenou realitu, poskytuje krátký pohled do historie a odpovídá na otázku, jakým způsobem a z jakého důvodu rozšířená realita vznikla a co může nabídnout. Dále jsou zde představeny některé současné významné systémy pracující s rozšířenou realitou a markery. V této části je také představeno možné rozdělení výzkumných odvětví rozšířené reality. V rámci uvedeného rozdělení je definována oblast, do které spadá tato práce.

Třetí kapitola se poměrně detailněji zabývá systémem rozšířené reality využívající markery Uniform marker fields a zaměřuje se především na detekční algoritmus markeru ve scéně. Zmiňuje také vylepšení v podobě šedotónové varianty (UMF).

Čtvrtá kapitola pojednává především o teorii, která byla využita při vytváření detekčního algoritmu. Zabývá se problematikou projekce válce do reálné roviny z hlediska projektivní geometrie a zkoumá způsoby jakými je možno nalézt parametry potřebné pro detekci markerů na jeho plášti. Tato kapitola tvoří jádro této práce. V závěru kapitoly jsou popsány

významnější algoritmy použité v detektoru.

V páté kapitole je popsán krok po kroku návrh detekčního algoritmu doplněný o názorné ukázky z výstupu testovacího programu. Jsou zde uvedeny už konkrétní matematické vztahy a modely využití v reálné implementaci.

Testování implementace detektoru je popsáno v šesté kapitole. Jsou zde uvedeny výsledky a závěry z několika různých testů zaměřených na různé aspekty detektoru. Testovány byly i různé typy UMF. Výstupy z jednotlivých testů jsou vyneseny v přehledných tabulkách.

V poslední kapitola nastiňuje možnosti budoucí práce a zhodnocuje celou práci a její výsledky.

Tato práce plynule navazuje na semestrální projekt, ze kterého přebírá především teoretický úvod v podobě kapitol [2](#), [3](#) a některé úvodní části kapitoly [4](#).

Kapitola 2

Rozšířená realita založená na markerech

Rozšířená realita se dá chápat jako variace na virtuální prostředí, respektive na virtuální realitu. Ve virtuální realitě je člověk zcela obklopen počítačem vytvořeným prostředím, které zpravidla s reálným světem žádným způsobem nespojuje. Oproti tomu rozšířená realita pracuje přímo s reálným světem, který se snaží nějak rozšířit nebo obohatit. Zpravidla se do prostředí reálného světa vkládají počítačem generované objekty, které uživateli zprostředkovávají nové, jiným způsobem těžko dosažitelné, informace a vjemy. Příkladem může být navigace s rozšířenou realitou od společnosti Wikitude¹ nebo projekt společnosti Google GLASS².

První rozhraní, které lze považovat za rozšířenou realitu představil již roku 1965 Ivan Sutherland [12]. Zařízení však značně předběhlo svou dobu a bylo limitováno tehdejšími dostupnými technologiemi. Za první konferenci zaměřenou na rozšířenou realitu je považována IWAR (International Workshop on Augmented Reality), která proběhla v roce 1998 v San Franciscu. Dnes je systém virtuální reality předmětem intenzivního zkoumání a některé aplikace již nacházejí reálná využití v lékařství, strojírenství, výuce, zábavě, ale i každodenním životě. Vybrané příklady reálných aplikací jsou na obrázku 2.1.

Podle R. Azumy [1] každý systém virtuální reality splňuje následující obecné charakteristiky:

- Kombinuje reálný svět s virtuálním
- Reaguje v reálném čase
- Pracuje ve 3D prostoru

Z předešlého je zřejmé, že vytvoření aplikace, která využívá rozšířenou realitu, představuje skutečnou výzvu a nutně je potřeba skloubit nejnovější technologie a poznatky hned z několika oblastí informačních a výpočetních technologií.

Dle práce F. Zhou a kolektivu [17] v současné době probíhá intenzivní výzkum především v těchto oblastech:

- Hardware a software pro vytvoření virtuálního světa

¹VTT Technical Research Centre of Finland – www.wikitude.com

²Google GLASS – www.google.com/glass/start



Obrázek 2.1: Vlevo nahoře Ultimate Display I. Sutherlanda, vpravo nahoře navigace s rozšířenou realitou od Wikitude , vlevo dole servisní manuál pro auta značky BMW s rozšířenou realitou – pohled přes speciální brýle, lokalizace markeru ve scéně a vykreslení virtuálního objektu pomocí aplikace ARToolkit.

- Sledovací a rozpoznávací techniky, které reagují na změny a podněty z reálného světa a dodávají data pro vytváření odpovídajícího virtuálního světa
- Kalibrační a registrační techniky, které co nejpřesněji zasazují virtuální objekty do reálného světa
- Displeje a projekční zařízení, kde uživatel sleduje výslednou scénu. Kromě běžných displejů se může jednat například o speciální helmy
- Výpočetní hardware, který dokáže dostatečně rychle provádět potřebné výpočty a zároveň je schopen komunikovat se vstupními a výstupními zařízeními
- Techniky interakce mezi uživatelem a virtuálním obsahem scény

Dnes je nejvíce zkoumanou právě oblast sledování a rozpoznávání. Úkolem je nejčastěji potřeba získat relativní pozici kamery vzhledem k reálnému světu. Tuto oblast lze dále dělit na tři typy:

- Techniky založené na specifických senzorech (například magnetické senzory v rukavici)
- Techniky založené na obrazu z klasické RGB kamery
- Hybridní techniky, které používají více různých metod (GPS souřadnice společně s obrazem)

Nejaktivněji zkoumanou je technika založená na sledování a rozpoznávání v obraze z klasické RGB kamery. Využívá se zde především poznatků z oborů zpracování obrazu a počítačové vidění. Aby bylo možné úspěšně scénu dostatečně rychle a přesně kalibrovat, je potřeba

se v obraze, který nám dává kamera, vyznat. To často znamená vyhledávat potřebné a předem známé vlastnosti. Tyto vlastnosti obrazu nejčastěji bývají dvojího typu:

- Přírozené – nacházejí se přirozeně v reálném světě. Zpravidla jsou to významné body nebo oblasti v obraze jako hrany, rohy, texture, atd.
- Uměle přidané – častokrát mají podobu speciálních dvourozměrných vzorů tak zvaných markerů. Markery v obraze mají známé tvary a vlastnosti, které umožňují co možná nejrychlejší a nejpresnější lokalizaci v obraze a co nejvíce usnadňují výpočet požadovaných informací.

Právě markery a jejich možnosti použití jsou oblastí rozšířené reality, kterou se zabývá tato práce.

2.1 Markery

Mnoho systémů počítačového vidění požívaného v praxi využívá dvourozměrné vzory k přenosu informací. Tyto vzory jsou vytvořeny, tak aby byly jednoduše dekódovatelné a rozpoznatelné v obraze. Příkladem mohou být známé čárkové kódy na zboží nebo modernější QR (Quick Response) kódy a jejich varianty. Příklady jsou na obrázku 2.2.



Obrázek 2.2: Příklady reálně používaných vzorů zaměřených na přenos informace [6]

Tyto vzory jsou však určeny pouze k přenosu informace a nikoli k lokalizaci v obraze. Pro využití v aplikacích rozšířené reality proto nejsou vhodné. Slouží však jako inspirace pro vytváření speciálních markerů pro rozšířenou realitu. Některé metody a postupy určené k detekci těchto vzorů v obraze jsou také často úspěšně využívány při detekci markerů.

Pro kódování se zpravidla používá binárního obrazce, který se skládá z jednoduchých geometrických vzorů jako čtverec, obdélník, atd. Často se jednotlivé části vzorů se od sebe identifikují prahováním. Nejčastěji se využívá maximálního možného rozdílu v jasů, tedy barvy bílá a černá. Tímto se jednotlivé části snadno identifikují a minimalizují se chyby způsobené rozdílným osvětlením a různými vlastnostmi kamer.

Hlavní důvody proč QR kódy a jiné vzory určené k přenosu informace nejsou vhodné jako markery pro virtuální realitu jsou dva:

- Jejich tvary nejsou zamýšlené ani vhodné k robustním výpočtům 3D pozice markeru vzhledem ke kameře
- Typicky vyžadují velkou plochu v obraze a limitující je i vzdálenost od kamery

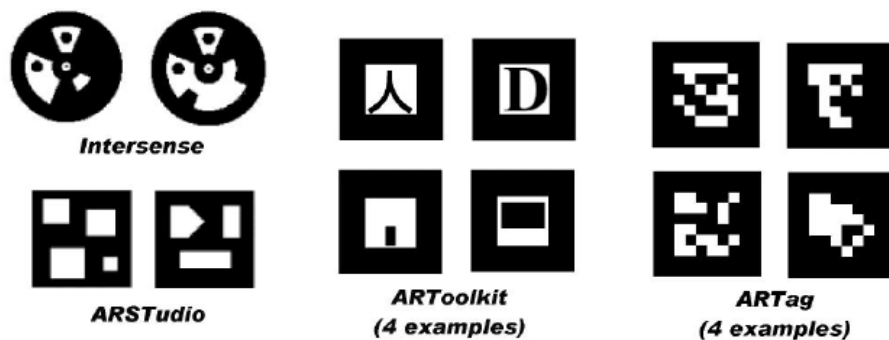
U markerů v rozšířené realitě obecně stačí 4 známé body k určení pozice kamery a dostatek informací k tomu, aby šli od sebe odlišit jednotlivé markery v dané sadě. Dokonce se vyplatí minimalizovat množství nesené informace, tak aby marker byl co možná nejjednodušší. Protože čím méně pixelů v obraze je využito markerem, tím více prostoru zbývá pro okolní reálnou scénu a navíc se tím často i zjednodušuje samotná detekce.

Markery pro rozšířenou realitu jsou navrhovány především s ohledem na tyto vlastnosti:

- Musí obsahovat minimálně čtveřici bodů k co nejpřesnějšímu určení polohy kamery. Tyto body by měli být co možná nejdále od sebe kvůli minimalizaci chyby zkreslení
- Musí obsahovat dostatek informací k jednoznačnému odlišení jednotlivých markerů v sadě
- Musí mít takový tvar a barevnost, aby se dali snadno detekovat a nedali se jednoduše zaměnit z jinými prvky scény. Tedy minimalizace falešně pozitivní chyby

2.2 Příklady markerů

Bylo vyvinuto několik systémů pro vytváření a detekci markerů pro rozšířenou realitu. Několik příkladů markerů těchto systémů je na obrázku 2.3.



Obrázek 2.3: Ukázky některých typů planárních markerů. Cirkulární Intersense markery se využívají výhradně k nalezení pozice. Ostatní typy markerů jsou speciálně navrženy pro rozšířenou realitu.

Dva významné systémy pro práci markery a rozšířenou realitu ARToolkit [10] a ARTag [6] jsou zde popsány podrobněji. Na jejich základní myšlenky navazuje tato práce, dalším podobným projektem je například ALVAR [11].

Porozumění uvedeným myšlenkám je důležité k pochopení systému Uniform marker fields, kterému se věnuje následující část této práce.

2.2.1 ARToolkit

ARToolkit je velmi populární volně dostupný systém pro vytváření aplikací využívající rozšířenou realitu založenou na markerech. Obecně je uznáván jako state-of-the-art v této oblasti.

Jeho jednotlivé markery mají široký černý čtvercový okraj a uvnitř se nachází množství různých vzorů. Okraj slouží pro detekci čtveřice bodů. Z těchto bodů se vypočte matice homografie mezi markerem a kamerou. Pomocí homografie se poté identifikuje vnitřní oblast markeru, ze které se vytváří příznakový vektor. Tento příznakový vektor se dále pomocí

korelace porovnává s vektory uloženými v databázi knihovny. Získává se pravděpodobnostní faktor ke každému vzoru a vybere se nejvíce odpovídající. Tímto způsobem se marker identifikuje.

Tento přístup se ukazuje jako velmi robustní a jednoduchý, má však dvě hlavní slabiny.

Velmi záleží na správném zvolení prahu při detekci okraje a při vytváření příznakového vektoru z vnitřní oblasti. Hodnota prahu se mění s osvětlením a vlastnostmi použité kamery.

Kvůli různému osvětlení a natočením vzoru, je pro každý vzor v markeru potřeba 12 různých prototypů (4 typy natočení a pro každé 3 typy osvětlení). Rychlost a velikost knihovny je velmi ovlivněna počtem vzorů. Pokaždé je potřeba porovnat každý vzor s každým, což pro větší množství vektorů znamená nezanedbatelný čas nutný ke zpracování.

2.2.2 ARTag

ARTag je rozšířením přístupu knihovny ARToolkit, který se snaží řešit výše zmíněné problémy. Největším rozdílem je kódování vnitřní části, kde se nyní nenachází libovolný motiv, ale matice o rozměrech 6×6 buněk symbolizující 10 bitový kód zakódovaný pomocí metody FEC (Forward error-correction coding)[15], kvůli odolnosti vůči chybám.

Markery jsou opět detekovány podle širokého okraje, který definuje body pro nalezení homografie a pozice kamery. Tentokrát se však okraje nehledají pomocí prahování, ale pomocí metod založených na hledání hran v obraze. Problematická volba prahu zde tedy odpadá. Vypočtená homografie je poté použita k rozdělení vnitřní části na jednotlivé buňky matice. Pomocí rozdílných barev buněk se následně dekoduje informace, která identifikuje daný marker.

Rozšíření ARTag je robustnější, rychlejší a umožňuje efektivněji kódovat více markerů než při použití klasického ARToolkitu. Konkrétně matice o rozměrech 6×6 umožňuje teoreticky zakódovat 210 markerů. Avšak z toho je 23 markerů reálně nepoužitelných, proto je tento počet celkem 1001. Použitím bílého okraje matice markeru namísto černého lze celkový počet reálně použitelných markerů zdvojnásobit.

Kapitola 3

Uniform marker fields

V předchozích kapitolách byly představeny současné systémy pracující s markery. Pokud je potřeba pracovat s více markery ve scéně, musíme existovat možnost, jak tyto markery vzájemně odlišit. A navíc, má-li být marker z představených systémů korektně detekovaný, musí být vidět celý a nesmí jej nic překrývat. Markery ARTagu mohou být sice částečně překryté díky redundantním čtvercům ve vnitřní části markeru, ale toto je pouze důsledek použití robustní technologie a nikoli přímý záměr. V případě volného pohybu kamery ve scéně často nastává právě situace, kdy je některý marker vidět pouze částečně.

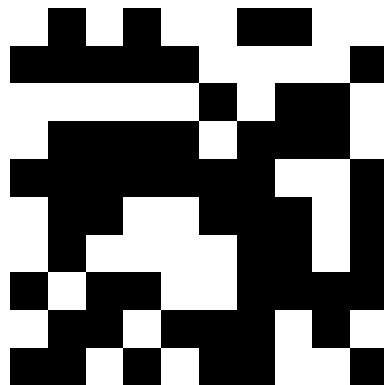
Obecně platí, že při použití více markerů ve scéně je nutné, aby byl viděn alespoň jeden z těchto markerů celý a tento marker musí být dostatečně velký na to, aby se z něj dala spočítat dostatečně přesná matice homografie určující relativní pozici kamery ve scéně vůči markeru.

Tyto požadavky jdou v zásadě proti sobě. Ideální marker tedy musí být dostatečně malý na to, aby se vešel do zorného pole kamery a zároveň dostatečně velký na to, aby poskytl požadovanou informaci.

Těmto požadavkům se snaží vyhovět technologie Uniform marker fields [13]. Termín Marker field představuje planární strukturu, která se skládá ze samostatných podoblastí. Každá podoblast může vystupovat jako klasický plnohodnotný marker se všemi náležitými vlastnostmi. Tyto podoblasti se v rámci Marker field struktury velmi výrazně překrývají a tím umožňují jejich identifikaci a to jak z pohledu z blízka tak i z větší vzdálenosti. Algoritmu také nečiní problém případy, kdy některé části celého Marker field nejsou zcela viditelné. Slovo uniform značí, že jsou příznaky použité pro detekci lokalizovány uniformě po celé ploše struktury. Názorná ukázka struktury Marker field je na obrázku 3.1.

Jednotlivé podoblasti Marker field jsou založeny na *4-orientable n^2 -windows arrays* [3]. Což jsou dvourozměrné matice $n \times n$ s binárními hodnotami. Jako nejlepší velikost těchto matic se pro praktické použití v Uniform marker fields ukázal rozměr 4×4 . Každá matice (podoblast) je přítomna maximálně jednou v celé struktuře a bere v úvahu i všechny 4 možné rotace. Tímto je zajištěno, že každá podoblast nese všechny potřebné informace, aby mohla fungovat jako samostatný marker.

Avšak vytvoření Uniform Marker Fields je velmi výpočetně náročné s exponenciální složitostí a je z časových důvodů nereálné k němu přistupovat hrubou silou. Proto byl vyvinut paralelní genetický algoritmus, který je schopen v kombinaci s výkonným superpočítačem tyto Uniform Marker fields vytvářet.



Obrázek 3.1: Ukázka části binárního markeru systému Uniform marker fields.

3.1 Detekce marker fields

Pro účely této práce je nejdůležitější částí celého systému UMF detekce markeru v obraze. Oproti dříve uvedeným markerům (ARToolkit 2.2.1, ARTag 2.2.2) Uniform Marker Fields nevyužívají pro lokalizaci a identifikaci markeru speciální příznaky, ale veškerá informace se vyčítá z binární šachovnice podoblastí. Tímto je zajištěno, že k úspěšnému zpracování markeru stačí nejmenší možná plocha.

Algoritmus detekce se snaží být co možná nejefektivnější, aby mohl být použit například na mobilních zařízeních v reálném čase.

Detekční algoritmus pro korektní detekci předpokládá, že významná část obrazu musí být pokryta Uniform Marker Fields. Toto může být v některých případech pro algoritmus limitující.

Detekční algoritmus obsahuje čtveřici následujících kroků. Tyto kroky představují významnou inspiraci pro představovaný detekční algoritmus na válci a některé z kroků jsou přebrány téměř kompletně.

3.1.1 Extrakce edgelů

Edgely definující hrany v obraze jsou zde popsány pomocí dvojice bodů. Algoritmus neprovádí kompletní hranovou detekci obrazu, ale vyhledává hranové body pouze na řídké mřížce. Tímto se velmi snižuje výpočetní náročnost procesu.

Při detekci hranových bodů \mathbf{p}_0 se používá adaptivní prahování založené na pohyblivém průměrovacím okénku. Následně je použit Sobelův operátor na orientační zjištění směru gradientu hrany.

Poté se hledá ve směru kolmém na gradient s_0 cyklicky další hranový bod \mathbf{p}_i . Tento postup ukazuje rovnice 3.1.

$$\tilde{\mathbf{p}}_{i+1} = \mathbf{p}_0 + iw\hat{s}_i \quad (3.1)$$

Pokud je takový bod nalezen přepočte se na základě nalezeného bodu směrnice přímký 3.2.

$$s_{i+1} = \mathbf{p}_{i+1} - \mathbf{p}_0 \quad (3.2)$$

Na této přímce se poté postupně hledají další body a stejný postup se aplikuje pro oba směry, respektive pro kladná i záporná i . Platí, že čím více se takovýchto bodů nalezne a jejich vzájemná vzdálenost je delší, tím lze považovat edgel za přesnější. Výsledný edgel, respektive přímka kopírující hranu šachovnice markeru, je poté definována na základě dvou nejkrajnějších nalezených bodů.

3.1.2 Zjištění pozice dvou hlavních úběžníků

Dalším krokem je nalezení dvou hlavních úběžníků, které definují průsečíky všech nalezených edgelů z předchozího kroku.

Nejprve jsou edgely podle úhlu směrnic rozděleny do dvou skupin náležících dvou hlavním úběžníkům. V praxi je dále často nutné před samotným hledáním úběžníků použít metod, které jsou schopny odstranit ze skupiny i případné chybné vstupní edgely. Příkladem takové metody může být algoritmus RANSAC.

Ze skupin se následně vypočte hledaná dvojice úběžníků \mathbf{v}_1 a \mathbf{v}_2 , což jsou body, ve kterých se všechny edgely z dané skupiny protínají. Edgely jsou vlastně přímky \mathbf{l}_i popsány pomocí obecné rovnice a bod \mathbf{v} je v homogenních souřadnicích. Pak platí rovnice 3.3.

$$\forall i : \mathbf{v} \cdot \mathbf{l}_i = 0 \quad (3.3)$$

Parametry jednotlivých přímek tvoří v reálné projektivní rovině trojrozměrný vektorový prostor bez počátku. Body reálné projektivní roviny korespondují s nadrovinami, které procházejí počátkem. Takže úběžník nalezneme nalezením nadroviny přes všechny nalezené přímky.

Parametry přímek jsou nastaveny tak, že přímky s větší vzdáleností mezi body mají odpovídajícím způsobem zvětšeny hodnoty definičního vektoru. A tím je zvětšen jejich vliv na výsledek. Normála nadroviny je nalezena jako směr s nejmenší variancí eigen dekompozice korelační matice C podle rovnice 3.4.

$$C = (\mathbf{l}_0 \dots \mathbf{l}_N)(\mathbf{l}_0 \cdot \mathbf{l}_N)^T \quad (3.4)$$

Matice C má rozměr 3×3 a je symetrická, takže dekompozice je velmi efektivní.

3.1.3 Nalezení parametrů dvou vějířů přímek

Nejdříve je spočtena přímka představující horizont \mathbf{h} , na kterém leží oba nalezené úběžníky \mathbf{v}_1 i \mathbf{v}_2 . Jednotlivé přímky vějíře korespondující s hranami markeru se určují podle rovnice 3.5, kde $\hat{\mathbf{x}}$ značí normalizovaný vektor.

$$\mathbf{l}_i = \hat{\mathbf{l}}_{base} + (ki + q)\hat{\mathbf{h}} \quad (3.5)$$

\mathbf{l}_{base} značí vhodně zvolenou počáteční přímku, nejlépe přímku procházející středem obrazu a daným úběžníkem. Část rovnice $(ki + q)$ je vypočtena pro všechny edgely. Tyto hodnoty jsou shlukovány do jednotlivých shluků a je jim přiřazeno i . Poté je nalezeno optimální k a q pro každý shluk pomocí lineární regrese.

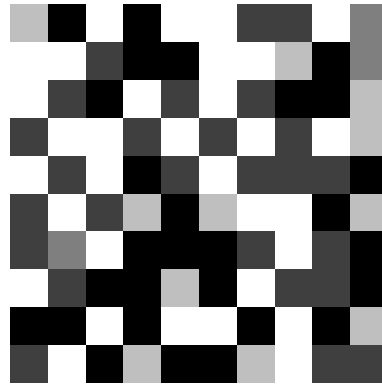
3.1.4 Extrakce modulů Uniform marker fields

Známe-li parametry obou vějířů přímek, stačí nyní jen detekovat průsečíky přímek, které vedou přesně mezi jednotlivými přímkami vějíře. Hodnoty jasu těchto bodů jsou vyčteny z obrazu a rozliší se pomocí adaptivního prahování. Poté jsou přečteny všechny možné matice

tvořící markery požadovaného rozměru. Vypočtené hodnoty tvoří klíč do hašovací tabulky. Pokud klíč existuje, marker je nalezen.

3.2 Grayscale marker fields

Grayscale marker fields (Uniform Marker fields ve stupních šedi) [14] představují rozšíření výše diskutovaných Uniform marker fields. Ukázka je na obrázku 3.2.



Obrázek 3.2: Ukázka části Grayscale marker fields.

Hlavní rozdíl spočívá v použití markeru s pěti různými odstíny šedé místo binární černobílé. Ostatní obecné vlastnosti zůstávají shodné. U těchto markerů není možné detekovat informaci přímo z barevnosti markeru, kvůli robustnosti vůči osvětlení, vlastnostem kamery a podobným proměnným vlivům. Namísto toho se využívá vlastností hranového gradientu mezi jednotlivými poli šachovnice. Avšak ze stejného důvodu, se nevyužívá číselná hodnota gradientu, ale pouze základní informace. Každé hraně se přiřadí číslo z množiny $\{-1, 0, 1\}$. Kde hodnoty -1 a 1 určují směr gradientu a číslo 0 značí, že barva sousedních polí je shodná, čili není zde hrana.

Detekční algoritmus je velmi podobný detekčnímu algoritmu pro Uniform marker fields. Rozdíl je pouze v posledním kroku, kde je nutné správně klasifikovat hrany, a v následné detekce markeru v rámci celého UMF.

Kromě uvedeného typu Šedotónového UMF používáme v této práci i Speciální šedotónový UMF, který navíc splňuje podmínku, která napomáhá úspěšnosti detekce na válci. Podmínkou je, že pokud takový typ markeru připevníme na plášť válce, dva nad sebou ležící čtverce nebudou mít nikdy stejnou barvu. Toto zajistí, že hrany kolmé na osu válce budou vždy celistvé a tedy detekce těchto hran bude přesnější.

Kapitola 4

Detekce Uniform marker fields na válci

V této kapitole se dostáváme k hlavnímu tématu práce. Cílem je detekování UMF a jeho šedotónové varianty na válci. Vstupem detekčního algoritmu je tedy obraz s válcem, který má na svém plášti UMF nebo případně jeho šedotónové varianty. Vzor na válci tedy vytváří přímky na plášti válce, které jsou kolmé na jeho podstavu, a prstence, které leží v rovinách rovnoběžných s podstavou válce.

Výstupem tohoto algoritmu je detekovaný marker a to tak, že přes válec je definována síť, jejíž průsečíky určují střed jednotlivých čtverců UMF. Výsledný algoritmus by měl pracovat jako rozšíření stávajícího detekčního algoritmu.

Detekce markerů na válci představuje zcela jistě výzvu i z toho důvodu, že během studia materiálů k této práci nebyla objevena žádná jiná práce, která by se zbývala podobným problémem. Všechny dosud představené detekční algoritmy předpokládali planární plochu markeru.

V praxi by mohl algoritmus umožnit markery detekovat například na sloupech nebo na potrubí. Válce obalené markerem by také jistě našli využití v různých aplikacích rozšířené reality. Příkladem by mohli být různé virtuální předměty, jejichž tvar a textura by se namapovala na válec, nebo třeba sloupy ve virtuálním stavění, které by mohly měnit texturu a barvu.

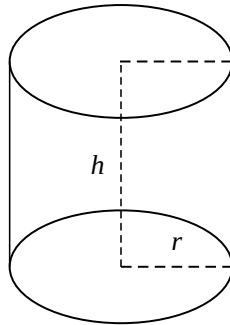
V kapitole jsou dále popsány obecné úvahy a teorie využitá dále v návrhu finálního detektoru. V poslední části kapitoly jsou také popsány významnější použité algoritmy.

4.1 Teoretické úvahy použité při vytváření detektoru UMF na válci

V této sekci je nejprve popsán válec a některé jeho základní vlastnosti, které jsou později využity při návrhu detektoru. Poté jsou v sekci zmíněny užitečné aspekty chování projekce válce z pohledu projektivní geometrie. A konečně závěr kapitoly se zabývá již konkrétními vlastnostmi, respektive jejich aproximacemi, které jsou dále využity v návrhu detekčního algoritmu.

4.1.1 Válec

Válec je jedním ze základních prostorových geometrických těles [2]. Obecný válec je určen dvěma rovnoběžnými podstavami a pláštěm. Všechny pomyslné přímky tvořící plášť jsou vzájemně rovnoběžné. V našem případě předpokládáme rotační válec, který má obě podstavy kruhové a pomyslné přímky tvořící plášť jsou na podstavu kolmé. Přímka h , která prochází středem obou podstav, se nazývá osa rotace. Ukázka rotačního válce je na obrázku 4.1.



Obrázek 4.1: Vyobrazení rotačního válce. Na obrázku je vyznačena osa rotace h a poloměr válce r .

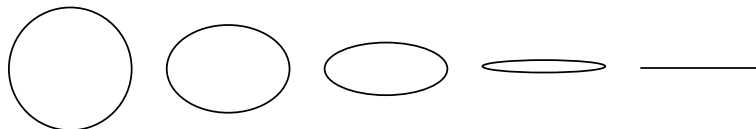
Poznamenejme, že výsledkem průniku pláště válce a libovolné roviny rovnoběžné s podstavami je kružnice, tuto kružnici nazvěme prstenec. Známe-li parametry prstence můžeme zpětně určit parametry roviny protínající válec. Těto vlastnosti budeme později využívat.

4.1.2 Projekce kružnice v prostoru

Předpokládejme, že v prostoru existuje rovina ρ_1 a kružnice k , která této rovině náleží. Dále existuje projekční rovina ρ_2 , do které kružnice k promítá. Budou-li se postupně měnit úhly mezi rovinami ρ_1 a ρ_2 od 0° do 90° , dostáváme následující významné varianty.

- Při 0° získáme shodný obraz kružnice k (respektive elipsu, která má obě poloosy stejně dlouhé).
- Zmenšuje-li se postupně úhel dostáváme obraz kružnice k jako elipsu jejíž vedlejší poloosa se postupně zkracuje, ale hlavní poloosa zůstává vždy stejně dlouhá s délkou odpovídající poloměru k .
- Při 90° je délka vedlejší poloosy nulová, čili kružnice k degradovala na úsečku.

Na obrázku je 4.2 je názorná ukázka změny tvaru projekce elipsy.



Obrázek 4.2: Projekce elipsy v prostoru. Úhel pohledu roste zleva (0°) doprava (90°).

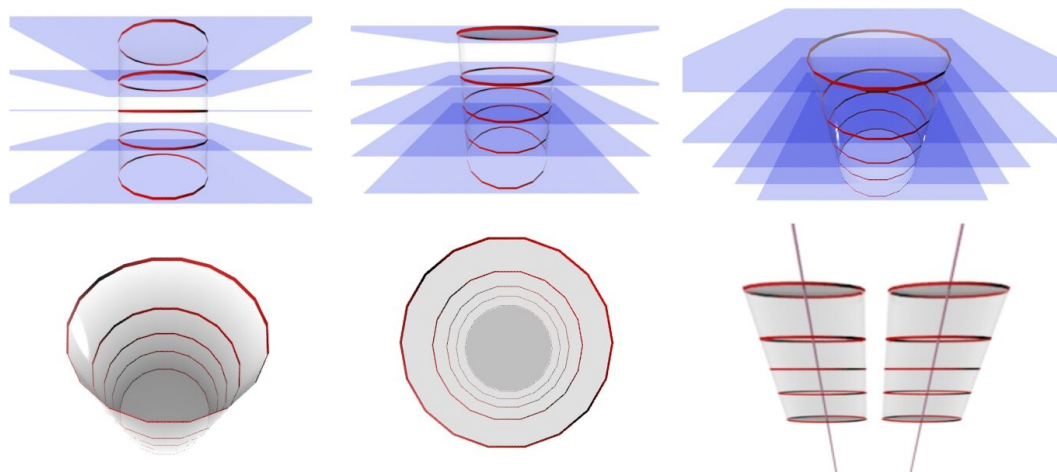
Tuto úvahu zúročíme později při hledání prstenců na válci a speciálního prstence – horizontu, který jak později ukážeme, lze definovat elipsou, která degeneruje právě na přímku.

4.1.3 Projekce válce v prostoru

V projektivní geometrii se válec na reálné projekční rovině jeví jako rotační komolý kužel. Tento komolý kužel lze protáhnout na rotační kužel, jehož vrchol leží v nekonečnu. Tento bod určený vrcholem kužele je zároveň úběžníkem všech pomyslných přímk tvořících plášť a místem, kde pomyslné prstence na plášti degenerují v bod.

Pozorujeme-li válec na projekční rovině, zmíněné kruhové prstence se díky výše zmíněné deformaci promítají na elipsy, kterým poměr mezi hlavní a vedlejší poloosou klesá s blížícím se úběžníkem. Jednotlivé elipsy (prstence) na plášti válce leží v rovnoběžných rovinách.

Avšak pravý úhel mezi osou rotace a hlavními poloosami elips není zaručen – respektive nastává pouze ve zvláštních případech, kdy je osa rotace válce ve středu projekční roviny. Například rotační osy dvou vedle sebe stojících válců směřují jistě do jednoho úběžníku. Prstence obou válců leží v odpovídajících rovnoběžných rovinách a jejich poloosy směřují do vlastního úběžníku. Výše zmíněné úvahy jsou vyobrazeny na obrázku 4.3.

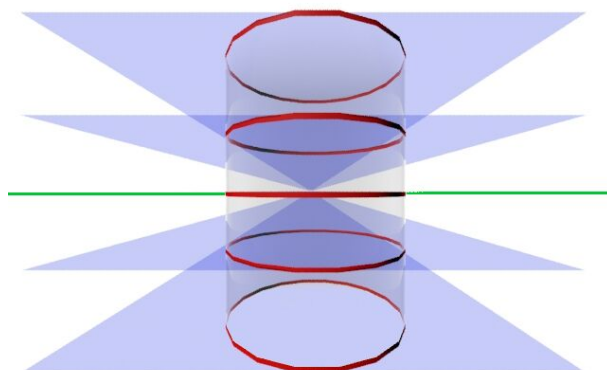


Obrázek 4.3: Na obrázku je vidět projekce válce v prostoru. Všimněme si jakým způsobem se promítají elipsy (prstence) a jimi procházející roviny a také si povšimněme dvou válců vedle sebe, jak jejich rotační osy směřují do úběžníku. Poznamenejme, že perspektiva je zde značně přehnaná. Červeně jsou vyobrazeny prstence a modře roviny které je protínají.

Na každém takovém válci existuje právě jedna elipsa (prsteneček), která na pozorované projekční rovině degeneruje na úsečku. Tato úsečka leží v projekční rovině na přímce, která se nazývá horizont. Horizont tvoří nadrovinu, ve kterém se sbíhají veškeré roviny rovnoběžné s podstavami válce. Tedy i roviny, které na válci definují zmíněné prstence. Připomeňme, že jednotlivé prstence stále leží v rovnoběžných rovinách, které jsou rovnoběžné s podstavou válce. Úvahy jsou znázorněny na obrázku 4.4

4.1.4 Hledání prstenců na válci

Nyní uvedeme úvahu, která později v detektoru umožní určit parametry horizontu.



Obrázek 4.4: N a obrázku jsou zobrazeny prstence červeně a jimi procházející roviny směřující do horizontu modře. Horizont je označen zeleně.

Parabola, podobně jako elipsa nebo kružnice, patří mezi kuželosečky [2], které lze popsat obecnou rovnicí kuželoseček 4.1.

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (4.1)$$

Kde reálné koeficienty a až f tvoří šestici možných parametrů kuželosečky. Obecně je tedy potřeba šestice bodů k jednoznačnému definování obecné kuželosečky, respektive paraboly. V praxi se však častěji setkáváme s kanonickým tvarem rovnice paraboly v normální poloze, který můžeme použít je-li osa paraboly rovnoběžná s osou y 4.2.

$$(x - x_0)^2 = 2p(y - y_0) \quad (4.2)$$

Kde vrchol paraboly V má souřadnice $[x_0, y_0]$ a parametr p definuje míru a směr otevření paraboly. Pro $p = 0$ degraduje parabola na přímku.

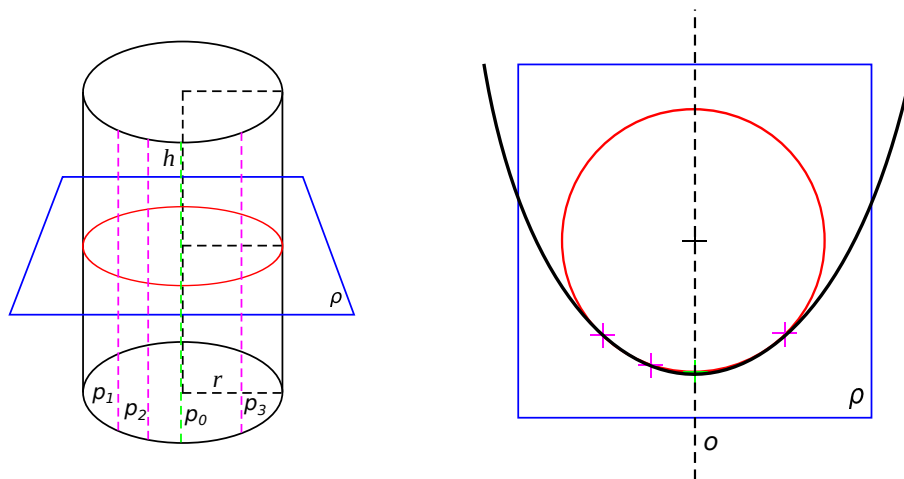
Tato rovnice má jen tři parametry, čili stačí pouze 3 body k jednoznačnému definování paraboly.

Dále vezmeme tři libovolné vzájemně neshodné přímky p_1, p_2, p_3 , které leží na straně pláště válce přivrácené k pozorovateli a libovolnou rovinu ρ rovnoběžnou s podstavou válce. Dále existuje přímka p_0 na plášti válce, která leží v rovině kolmé na projekční rovinu a zároveň této rovině náleží osa rotace válce. Nyní můžeme určit průsečíky roviny ρ s trojicí zvolených přímek, s prostřední přímkou p_0 a s osou rotace válce. Situace je zachycena na obrázku 4.5.

Máme-li takto nastavenou situaci můžeme přímkou o tvořenou průsečíkem roviny s prostřední přímkou p_0 a s osou rotace asociovat s osou y . Máme-li takto nastaveny podmínky můžeme v rovině pomocí třech nalezených průsečíků s přímkami p_1, p_2, p_3 definovat parabolu. Poznamenejme, že tyto tři body definují i prstence na povrchu válce. Nalezená parabola bude částečně kopírovat prstenek elipsy a bude otevřená ve směru k ose rotace válce.

Stejná situace nastává pro všechny roviny rovnoběžné s podstavou. V každé takové rovině je takto možné definovat parabolu se stejnými parametry.

Pokud se všechny roviny kolmé na podstavu válce protínají v horizontu, budou se také shodné paraboly v jednotlivých rovinách protínat na horizontu v jednom respektive



Obrázek 4.5: Ukázka situace na válci při detekování paraboly. Vlevo je situace z celkového pohledu. Trojice přímek p_1, p_2 a p_3 je zobrazena fialově čerchovaně. Zeleně čerchovaně je prostřední přímka p_0 . Modře je rovina ρ . Černě čerchovaně je osa válce h (r značí poloměr podstavy válce). Červeně je kružnice tvořená průnikem válce a roviny ρ – prstenec. Vpravo je nastíněna situace z pohledu roviny ρ . Jednotlivé průsečíky si barevně odpovídají s pravou stranou. Černě čerchovaně je přímka o rovnoběžná s osou elipsy (myšlená osa y) a černě tučně je výsledná parabola.

dvou společných bodech. Toto platí pro všechny paraboly se stejnými parametry, náležící různým rovinám rovnoběžným s podstavou. Právě této vlastnosti využívá detektor k určení nadroviny (přímky) tvořící horizont. Ukázka z výstupu algoritmu, kde lze názorně pozorovat paraboly protínající se na horizontu ve stejných průsečících, je na obrázku 5.13.

4.1.5 Aproximace parametrů

V předchozím textu jsou uvedeny některé vlastnosti a úvahy, které jsou použity pro detekci UMF na válci. Jediným zdrojem informací o scéně s válcem, kterou k detekci můžeme využít, je obraz z běžné kamery. Jak si ale později ukážeme, ve skutečnosti převádíme tuto obrazovou informaci pouze na skupiny hranových bodů.

Detekce těchto hranových bodů může být v principu nepřesná. Z tohoto důvodu jsme nuceni improvizovat a mnohé vlastnosti vědomě aproximovat. Ovšem vždy je kladena maximální snaha o co nejmenší možnou chybovost. V této části textu jsou zmíněny hlavní aproximace a zjednodušení, které jsou vědomě použity v návrhu detektoru.

U všech zjednodušení nebo zanedbání deformací způsobených projekcí v reálném prostoru jsme experimentálně ověřili, že chyba je při projekční deformaci, se kterou se můžeme setkat u běžných kamer, dostatečně malá a nežádoucí účinek lze považovat pro výsledný detektor za zanedbatelný.

V úvodní části detektoru se snažíme mezi detekovanými a hranovými body nalézt takové, které definují prstence na válci. Tyto prstence detekujeme nejprve jako regulární elipsy a přitom zanedbáváme skutečnost, že i ony samotné jsou deformovány projekcí.

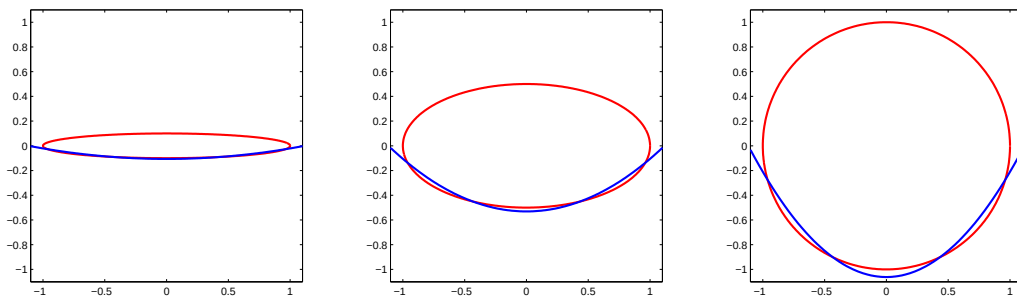
Dalším příkladem aproximace reálné projekce je například to, že všechny hlavní poloosy elips jsou rovnoběžné. Toto předpokládá algoritmus RANSAC v kroku 5.9. Ve skutečnosti však směřují do vlastního úběžníku a nejsou tedy u všech elips zcela rovnoběžné. V reálné

situaci je však odchylka malá a v implementaci je řešená rozsahem prahu.

Algoritmy na hledání parametrů elips mají tedy k dispozici pouze poměrně málo informací, které jsou navíc často postiženy chybami. Z tohoto důvodu nemusí detekce elipsy vždy skončit korektně.

Z důvodu zvýšení přesnosti detekce jsou v jednom kroku detekčního algoritmu nahrazeny elipsy, reprezentující prstence na válci, za paraboly. Důvodem je právě možnost využití výše zmíněného vztahu mezi parabolami a horizontem. V praxi to znamená reálné zpřesnění detekce. Parabola samozřejmě nemůže aproximovat celou elipsu, ale v případě detektoru pracujeme jen s přivrácenou stranou válce, čili nás zajímá pouze přivrácená polovina elipsy. Tuto polovinu elipsy je parabola schopná poměrně dobře aproximovat s větší chybou pouze u krajů, kde však při reálném použití selhávají i detekované elipsy. Proto tato chyba není pro výsledný detektor příliš významná a výhody aproximace pomocí parabol vítězí nad nedostatky.

Názorná ukázka nahrazení elips parabolami je na obrázku 4.6. Názorně ukazuje, že čím větší je poměr mezi hlavními poloosami elipsy, tím lépe parabola aproximuje přivrácenou stranu elipsy. V případě, že elipsa degraduje na přímku, tak je i parabola degraduje na přímku a chyba je nulová. Naopak je-li elipsa zároveň kružnicí je chyba maximální.



Obrázek 4.6: Obrázek demonstrující chybu nahrazení elipsy parabolou. Parametry paraboly jsou získány pomocí metody nejmenších čtverců z poloviny bodů elipsy. Toto přesně simuluje situaci, která nastává v detekčním algoritmu. Z obrázku je patrné, že s rostoucím poměrem mezi dvěma poloosami elipsy se chyba zmenšuje. Elipsa je znázorněna červeně, parabola modře.

Při nahrazování elips parabolami nevyužíváme zcela přesně výše zmíněné úvahy založené na třech shodných bodech pro každou rovinu. Z důvodu nepřesnosti se snažíme z každé množiny dat vydefinovat parabolu co nejpřesněji s použitím všech těchto dat. Výsledný horizont je potom definován dvojicí bodů, která vznikne průměrováním všech průsečíků jednotlivých parabol. Podrobněji bude postup detekce horizontu popsán dále.

Při detekci elips a hledání parabol také předpokládáme, že osa rotace válce je kolmá k rovinám tvořeným parabolami respektive k jejich hlavním poloosám. Jak jsme si však ukázali výše, nemusí to být pravda. Reálné testování však ukázalo, že výchylka je při reálné perspektivě kamery pro detekci nepodstatná.

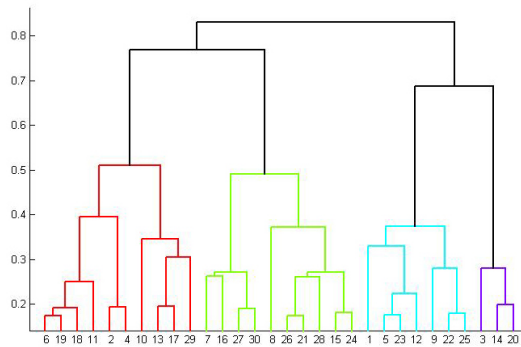
4.2 Použité algoritmy

V této kapitole jsou podrobněji popsány některé významnější algoritmy, které jsou použity při návrhu implementace detektoru.

4.2.1 Hierarchického shlukování na předem neznámý počet shluků

Metoda hierarchické shlukování [16] patří mezi jednu z metod shlukování prvků do reprezentativních shluků na základě definované vzdálenosti. Základem a výstupem hierarchického shlukování je struktura zvaná dendrogram (příklad dendrogramu je na obrázku 4.7). Dendrogram má tvar stromu, kde listy jsou jednotlivé prvky před začátkem shlukování a kořen tvoří jeden shluk, který pojímá všechny původní prvky.

Algoritmus v každé iteraci sloučí dva nejbližší uzly grafu v nový uzel. Takto pokračuje iterativně dokud nezůstane pouze kořen. Z dendrogramu lze poté vyčíst, jak byli prvky (shluky) postupně přidávány a jak se zvětšovaly vzdálenosti mezi nimi.



Obrázek 4.7: Ukázka dendrogramu. Jednotlivé různě barevné větve oddělují definované shluky.

Předpokládejme, že z předem dané množiny prvků vytváříme rozumně oddělené shluky, ale neznáme předem jejich celkový počet. Máme-li hotový dendrogram, můžeme sledovat, jak se postupně zvětšovaly vzdálenosti mezi shlukovanými uzly. V případě rozumně oddělených shluků nastane v jedné iteraci výrazný skok, který nastává v momentě, kdy se algoritmus pokouší sloučit dvojici shluků, které jsou již finálními. V případě nalezení takového skoku je možno prohlásit, že v této iteraci jsou nalezeny hledané shluky a výstupem jsou všechny existující shluky v dané výšce stromu – dendrogramu.

Hierarchického shlukování využíváme v detektoru celkově dvakrát a to při shlukování nalezených přímek a parabol.

4.2.2 RANSAC

Algoritmus RANSAC je známá iterační metoda [18], která obecně slouží k hledání prvků vyhovujících definovanému modelu v zadané sadě prvků. Sada prvků obsahuje kromě vyhovujících bodů (inliers) i šumové body (outliers), které modelu nenáleží. RANSAC je obecně nedeterministický a dospěje k požadovanému výsledku pouze s určitou pravděpodobností. Tato pravděpodobnost roste s počtem provedených iterací.

Počet kroků algoritmu lze předem odhadnout podle vzorce 4.3.

$$1 - p = (1 - w^n)^k \Rightarrow k = \frac{\log(1 - p)}{\log(1 - w^n)} \quad (4.3)$$

Kde p je pravděpodobnost, že algoritmus skončí korektně, většinou se volí poměrně vysoká například 0.9 a více (jinak by ani algoritmus nemělo smysl používat). Pravděpodobnost $1 - p$ tedy odpovídá tomu, že algoritmus neskončí korektně. Výraz w odpovídá poměru výskytu inliers vůči všem prvkům, tento poměr lze předpokládat z apriori znalosti vstupních dat. Výraz n je počet prvků potřebných k odhadnutí modelu a k je hledaný počet iterací. Výraz $(1 - w^n)^k$ tedy odpovídá pravděpodobnosti, že algoritmus nikdy nevybere správné prvky k vytvoření modelu a to je zřejmě ekvivalentní pravděpodobnosti, že algoritmus nenajde řešení. Tento výsledek ovšem předpokládá, že můžeme jeden prvek vybrat do modelu vícekrát a toto většinou nenastává. Proměnná k tedy reprezentuje horní hranici počtu iterací, do které má smysl iterovat.

Důležitým krokem algoritmu je definice hledaného modelu. Parametry modelu se poté odhadují z potřebného počtu náhodně vybraných bodů. Je nutné také definovat rozhodovací funkci, která určuje zda je zkoumaný prvek inlier nebo outlier vůči modelu. Toto se zpravidla definuje jako pevně stanovený práh nad nějakou vzdáleností bodu od odhadnutého modelu.

Algoritmus probíhá zjednodušeně takto:

- Iterujeme přes odhadnutý počet iterací
- Náhodně vybíráme prvky z dané množiny a odhadujeme model
- Pro všechny prvky množiny zkusíme, zda sedí do odhadnutého modelu (jsou inliers)
- Zapamatujeme si počet inliers
- Po provedení požadovaného počtu iterací algoritmus končí a výsledkem je model s největším počtem inliers
- Ze všech inliers modelu přepočítáme znova celý model a tím jej zpřesníme

Algoritmus RANSAC využíváme při hledání parametrů válce pomocí elips a při hledání korektních parabol. Toto bude podrobněji popsáno později.

4.2.3 Hledání parametrů elips

Obecnou elipsu můžeme definovat délkou hlavní a vedlejší poloosy, souřadnicemi středu a úhlem natočení. Pro nalezení obecné elipsy je tedy nutné určit 5 parametrů, čili je potřeba alespoň pětice různých bodů. Mnohé algoritmy však pracují s obecnou rovnicí kuželosečky, která má šest parametrů, a potřebují tedy šestici různých bodů.

Hledání elips a obecně základních geometrických tvarů patří mezi zásadní úkoly oblasti zpracování obrazu a počítačového vidění. Dosud bylo představeno značné množství přístupů k detekci elips a některé z nich zde krátce představíme.

Jeden typ přístupu [9] k určení parametrů elipsy využívá Hougovy transformace, tyto přístupy jsou sice velmi robustní, ale jsou výpočetně a paměťově velmi náročné.

Byly představeny i metody elipsy pomocí algoritmu RANSAC, který je schopen nalézt parametry elipsy i ve velmi zašuměných datech [4].

V současné době jsou nejvyužívanějším přístupem detektory elips na bázi metody nejmenších čtverců [8]. Algoritmy zpravidla hledají v množině bodů parametry obecné kuželosečky podle rovnice 4.1.

Na výsledné parametry jsou poté aplikována omezení, která co možná nejlépe zajistí, že nalezená kuželosečka je elipsou nebo není. Možná omezení pro koeficienty rovnice 4.4 jsou například tato:

$$a + c = 1, f = 1, a^2 + \frac{1}{2}b^2 + c^2 = 1 \quad (4.4)$$

Nejvhodnějším nalezeným algoritmem pro využití v implementaci tohoto detektoru je Direct Least Square Fitting of Ellipses [7]. Tento algoritmus vychází z metod založených na nejmenších čtvercích, ale na rozdíl od nich přímo hledá elipsu, takže odpadá následné omezení nalezených parametrů. Oproti ostatním uvedeným algoritmům má především tyto výhody:

- Vždy definuje elipsu a to i pro velmi špatná data
- Je prokazatelně robustnější než u jiných způsobů
- Výpočet je efektivnější než u jiných způsobů

Tento přístup je v aktuálně považován za etalon mezi detektory elips a je implementován i ve známé knihovně OpenCV ¹, kterou mimo jiné využívá implementace detektoru.

¹Opensource Computer Vision Library: opencv.org

Kapitola 5

Návrh a implementace detekčního algoritmu

V úvodu kapitoly jsou uvedeny některá omezení a předpoklady, které algoritmus vyžaduje pro správnou funkčnost.

Hlavní část kapitoly poté tvoří postupný popis jednotlivých kroků návrhu implementace detekčního algoritmu. Některé části jsou převzaty z původního planárního detektoru, u těchto kroků se text věnuje převážně potřebným úpravám a samotný algoritmus je nastíněn poměrně stručně s odkazem na původní práci [13]. Naopak části specifické pouze pro detekční algoritmus na válci jsou probrány poměrně podrobně.

Detekční algoritmus se skládá z následujících kroků:

- Detekce hran na vzorkovací mřížce
- Detekce skupin hranových bodů
- Hledání parametrů přímek ze skupin hranových bodů
- Nalezení dvou nejčastějších směrů přímek a rozlišení mezi přímkami na plášti a prstenci
- Nalezení úběžníku přímek na plášti
- Shlukování podobných přímek na plášti, tak aby každou hranu reprezentovala právě jedna přímka
- Hledání parametrů elips definujících prstence na plášti válce ze skupin hranových bodů
- Nalezení parametrů válce pomocí vybraných vhodných elips algoritmem RANSAC
- Hledání parametrů parabol ze skupin hranových bodů původních elips na základě dříve nalezených parametrů válce
- Nalezení horizontu a společného průsečíku parabol na horizontu pomocí vybraných vhodných parabol algoritmem RANSAC
- Přepočítání všech parabol na základě společného průsečíku
- Doplnění případných chybějících parabol
- Vytvoření výsledné mřížky definující nalezené středy čtverců markeru na povrchu válce

Popis kroků je vždy doplněn o názornou ukázkou v podobě obrázků z výstupu implementace algoritmu. Vybrané obrázky jsou voleny tak, aby co nejlépe popisovali zvolený krok. Je vybrána dvojice příkladů. První příklad představuje válec v ideální poloze se vzorem ideální šachovnice jako markeru, tento příklad ukazuje ideální stav. Druhý příklad má jako vzor šedotónový UMF. Válec je také natočen. Tato scéna ukazuje složitější příklad, kde podstata některých kroků algoritmu lépe vynikne. Nežádoucí vlivy pozadí a falešných hran byly také potlačeny, pro co nejnázornější ukázkou.

Algoritmus je však samozřejmě schopen pracovat i za složitějších podmínek. V kapitole 6 je implementace detektoru testována na velkém množství různých podmínek a v příloze D jsou vybrané obrázky s názornými ukázkami výsledných detekcí z testovacích dat.

5.1 Předpoklady a omezení pro vstupní data

Vstupní data detektoru tvoří jeden obrázek z běžné kamery nebo fotoaparátu. Obrázek může být v barevný nebo černobílý. Algoritmus pracuje pouze s jasovou informací v obraze a převádí vstupní obraz na odstíny šedi v rámci předzpracování vstupu.

Má-li být detekce korektní je potřeba zajistit tuto trojici předpokladů:

- Na vstupním obrázku se předpokládá právě jeden válec s vyobrazením markeru na plášti
- Marker musí na plášti válce tvořit přímky kolmé na podstavu válce a prstence rovnoběžné s podstavou válce
- UMF je možno korektně detekovat pouze v případě, když je možné správně identifikovat alespoň jednu podoblast o rozměrech 4×4

Při nedodržení těchto podmínek detekce selže.

Současný návrh a implementace detektoru pracuje jen s jedním konkrétním snímkem a nijak nevyužívá časovou informaci. Přidání a využití časové informace se nabízí jako možné vylepšení v budoucnu. Například by se mohlo vyzkoušet zakomponování Kalmanova filtru do detektoru pro sledování jednotlivých čtverečků markeru. Tato nová informace by mohla přinést zpřesnění a zvýšení robustnosti detekce. Zlepšení by mělo být patrné například při náhlém překrytí části markeru, kdy na okamžik ztrácíme obrazovou informaci o markeru.

Omezení v podobě rušivých falešných hran pozadí, počtu viditelných políček a nutné velikosti částí markeru jsou podrobně testována a vyhodnocena v kapitole 6.

5.2 Detekce hranových bodů na vzorkovací mřížce

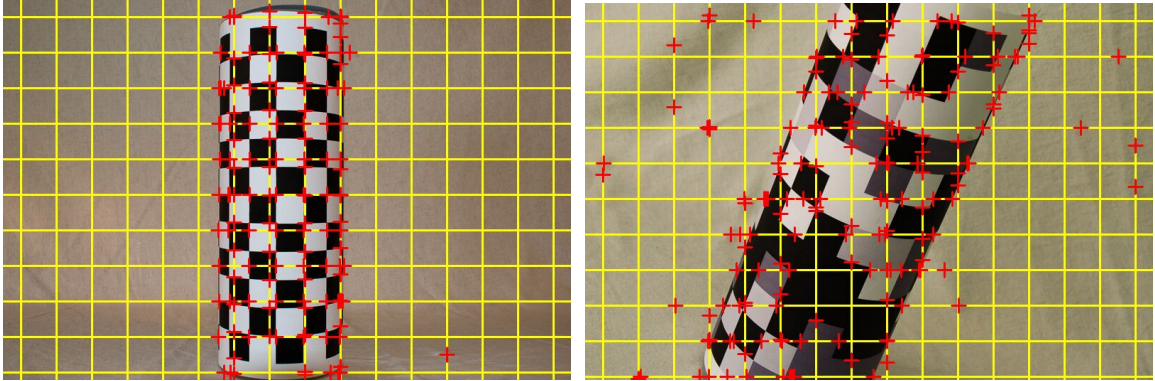
Algoritmus pro detekci hranových bodů je převzat z původního detekčního algoritmu Uniform Marker Fields. Jedním z cílů uvedeného postupu je snaha o využití minimálního počtu bodů a tedy minimalizace počtu přístupů do paměti dat zdrojového obrazu. Doplníme, že algoritmus pracuje s obrazem v odstínech šedi, čili hranové body rozlišuje podle skokové změny jasu.

Algoritmus detekuje hranové body pouze na zvolených skenovacích přímkách s pravidelným rozstupem v horizontálním a vertikálním směru. Hustota skenovacích přímek se ovlivňuje pevně stanoveným parametrem.

Samotné hranové body se na skenovacích přímkách detekují pomocí adaptivního prahování. Pro každou přímku se vytváří buffer hodnot několika předchozích bodů. Zda je aktuálně

testovaný bod hranovým se zjišťuje prahováním průměrné hodnoty aktuálního stavu buferu a hodnoty daného bodu. Poznamenejme, že Šedotónové UMF vyžadují citlivější práh na detekci hran než černobílý UMF nebo šachovnice.

Ukázka detekce hranových bodů je na obrázku 5.1.



Obrázek 5.1: Ukázka detekce hranových bodů. Hranové body jsou zobrazeny červeně, skenovací přímky žlutě. Všimněte si, že díky citlivějšímu hranovému detektoru, který Šedotónové UMF vyžadují, je na druhém obrázku detekováno několik falešných bodů.

5.3 Detekce skupin hranových bodů

Tato část detekčního algoritmu je také z velké části přebrána z původního detektoru. Bylo nutné pouze provést úpravy výstupu, aby jej bylo možné využít nejen k hledání přímek, ale i elips. Také bylo nutné upravit parametry a způsob predikce dalšího hranového bodu tak, aby bylo možné detekovat jak přímé hrany, tak i hrany zakřivené.

Vstupem tohoto algoritmu jsou detekované hranové body z předchozího kroku. Výstupem je pro každý vstupní bod skupina bodů sledujících jednu souvislou hranu, jejíž směr se skokově nemění. Tyto hranové body obsahují všechny potřebné informace definující rozložení UMF na válci a navíc nemusíme pracovat se všemi body obrazu, což značně urychluje výpočty. Poznamenejme, že skupiny hranových bodů, které obsahují méně než pět bodů prohlašujeme za málo přesné a automaticky zahazujeme.

Postup detekce probíhá následujícím způsobem:

- Ze vstupního bodu se hrubě určí směr gradientu hrany pomocí odezvy na křížový Sobelův filtr. Další hranový bod se predikuje ve směru hrany ve vzdálenosti zvoleného kroku od zdrojového bodu. Nyní se ve směru kolmém na zjištěný směr hrany stanoví interval, ve kterém se hledá další hranový bod.
- V tomto intervalu se předpokládá právě jedna nebo žádná hrana. Test zda interval nějakou hranu obsahuje probíhá prahováním rozdílu hodnot na koncích intervalu. V případě, že je jejich rozdíl dostatečně velký vyhledáme předpokládanou hranu v intervalu pomocí binárního vyhledávání.
- Je-li nalezen, kromě zdrojového bodu, alespoň jeden další bod. Směr gradientu se definuje jako směrový vektor přímky definované pomocí dvojice posledních nalezených

bodů. Směrový vektor hrany se tedy může postupně měnit a takto je umožněna detekce i oblé hrany elipsy.

Ukázka detekce hranových bodů je na obrázku 5.2.



Obrázek 5.2: Ukázka detekce skupin hranových bodů. Hranové body jsou zobrazeny červeně.

5.4 Hledání parametrů přímek

Ze všech nalezených skupin bodů se vypočtou parametry přímek které lze nejlépe proložit zvolenou skupinou bodů. K nalezení parametrů se využívá metoda nejmenších čtverců, která minimalizuje vzdálenost mezi zdrojovými body a proloženou přímkou. Detektor používá implementaci algoritmu z knihovny OpenCV.

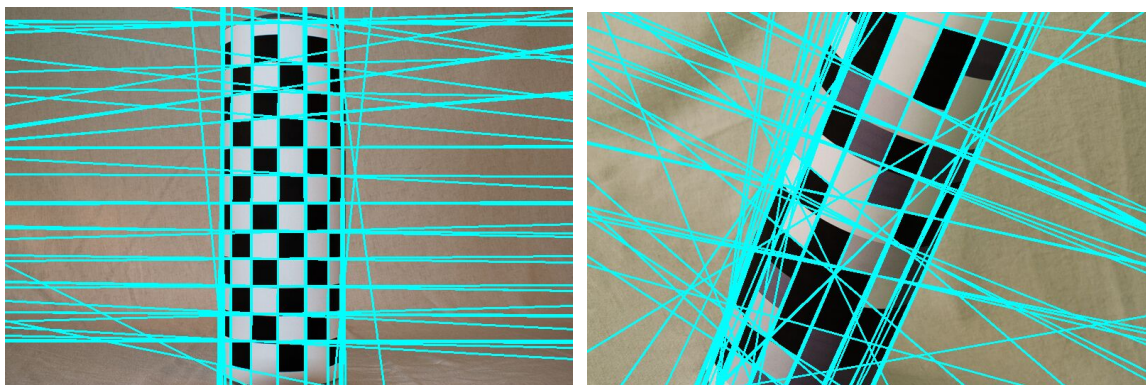
Pro každou přímkou si algoritmus ukládá následující data:

- Parametry vycházející z obecné rovnice přímky
- Skóre, respektive počet bodů, ze kterých je přímka definována
- Průměrnou vzdálenost a směrodatnou odchylku od definované přímky přes všechny její body
- Seznam zdrojových bodů

Ukázka detekce nalezených přímek na válci je na obrázku 5.3.

5.5 Nalezení dvou hlavních skupin přímek a rozlišení mezi přímkami na plášti a prstenci

V tomto kroku se hledá ze čtyř testovaných směrů jeden hlavní, do kterého směřují vektory přímek, které mají po součtu nejvyšší skóre. I zde je postup inspirován původním detekčním algoritmem. Střední přímky testovaných směrů jsou horizontální, vertikální, s náklonem vůči ose x 45° a 135° . Směry nerozdělují množinu všech možných směrů na čtyři disjunktní množiny, ale mají vůči sobě částečné přesahy. Jedna přímka tedy může spadat do jedné nebo maximálně dvou skupin.



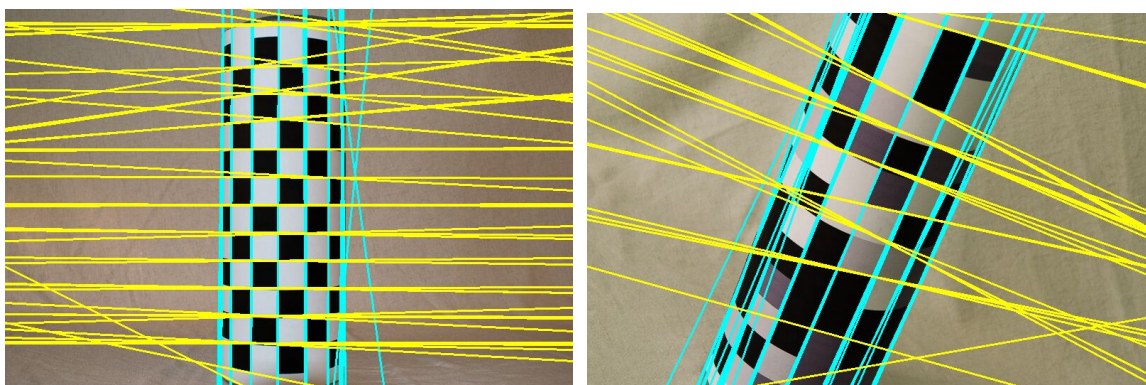
Obrázek 5.3: Ukázka všech nalezených přímek ze skupin hranových bodů. Přímky jsou zobrazeny světle modře.

Je-li nalezen hlavní směr, potom směr na něj kolmý obsahuje přímky druhého hlavního směru markeru. Zbývá rozlišit, který z této dvojice směrů reprezentuje přímky pláště, a který přilehlou (viditelnou) stranu elips tvořící prstence na válci.

Můžeme předpokládat, že přímky proložené přilehlou stranou elips spadají do směru kolmého k směru definovaného přímkami pláště, respektive tvoří hlavní směr a směr s přímkami pláště je kolmý na něj. Tento předpoklad nemusí platit pro přímky tvořené malou skupinou bodů po krajích válce, kde elipsa tvořící prstenec prudce mění směr. Tyto přímky však nejsou vhodné pro detekci finální elips, protože zpravidla neobsahují dostatečně mnoho informací a bývají nepřesné. Proto jejich případná ztráta neznamená problém.

Rozlišení přímek pláště od elips tvořící prstence se provede na základě průměrné vzdálenosti bodů od definované přímky a směrodatně odchylky. Přímky definované hranovými body z přímek na plášti mají zřejmě menší rozdíly ve vzdálenosti bodů od výsledné přímky než přímky definované pomocí hranových bodů přivrácené strany elips, protože se projeví zakřivení elips.

Ukázka výstupu je na obrázku 5.4.



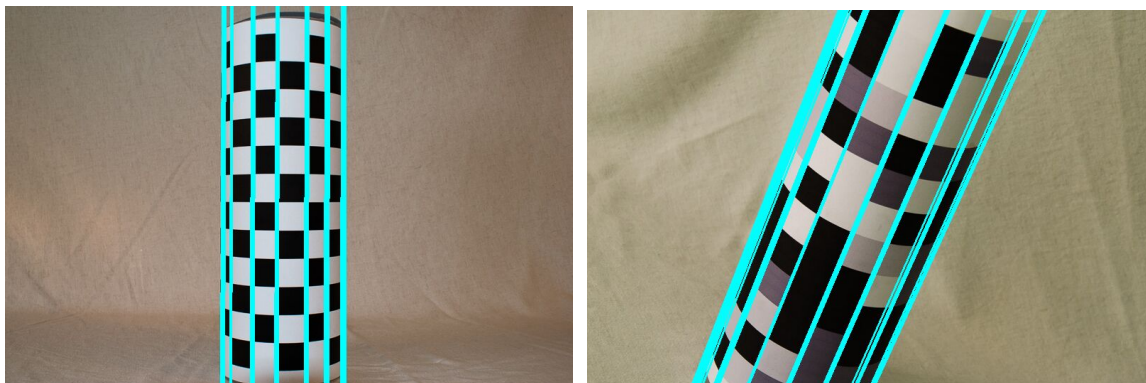
Obrázek 5.4: Ukázka rozlišení, které skupiny bodů definují přímku a které prstence (elipsu). Světle modré čáry definují přímky a žluté čáry prstence.

5.6 Nalezení úběžníku přímek na plášti

Tato část je prakticky kompletně přebrána z původního detektoru. Pouze bylo potřeba vytvořit wrapper, který je schopen převádět reprezentace přímek mezi původním detektorem a detektorem UMF na válci. Vstupem nalezení úběžníku jsou tedy přímky na plášti válce, které sledují hrany tvořené vzorem UMF.

Algoritmus detekce úběžníku je podrobně popsán v sekci 3.1.2 v rámci popisu původního detekčního algoritmu UMF.

Výsledkem algoritmu není jen nalezený úběžník, ale i upravené zdrojové přímky, které procházejí úběžníkem. Ukázka je na obrázku 5.5.



Obrázek 5.5: Ukázka nalezení úběžníku přímek. Upravené přímky jsou zobrazeny světle modře.

5.7 Shlukování podobných přímek na plášti

V předchozím kroku dostaneme sice již správné přímky, ale je jich zpravidla příliš mnoho a je potřeba je upravit tak, aby jednu hranu tvořenou markerem reprezentovala právě jedna přímka.

Metoda vytvoření vějíře přímek, kterou využívá původní detektor nelze využít. Pro určení vějíře totiž využívala znalosti obou úběžníků z planární plochy. V případě detekce na válci, jsme schopni zjistit pouze jeden takový úběžník a navíc vějíř je deformován zakřivením válce.

Z tohoto důvodu je k nalezení takového vějíře použita metoda hierarchického shlukování na obecně neznámý počet clusterů. Podrobněji je obecný algoritmus hierarchického shlukování popsán v kapitole 4.2.

V případě přímek je kritérium určující podobnost jednotlivých shluků vzdálenost průsečíků dvojice porovnávaných přímek, reprezentující shluky, s přímkou kolmou na osu válce a procházející středem obrazu.

Před spuštěním shlukování ještě odstraníme ze vstupní množiny přímek ty, co nespádají do trychtýře určeného přímkou určující sklon válce. Tuto přímku získáme jako výstup po algoritmu RANSAC na elipsách 5.9.

Algoritmus v každé iteraci vyhledá dvojici nejpodobnějších shluků a sloučí je v jeden. Každý shluk je reprezentován přímkou vzniklou váženým průměrem všech členských přímek shluku. Algoritmus si pamatuje aktuální a předchozí vzdálenost shluků a to i z předchozí

iterace. Z povahy hierarchického shlukování je zřejmé, že vzdálenosti stále rostou. Důležité však je, že poměr mezi posledními dvěma vzdálenostmi je prakticky konstantní, dokud se algoritmus nepokusí sloučit dva již finální shluky. Následně je průběh opět téměř konstantní, protože vzdálenosti mezi finálními shluky jsou opět řádově podobné. Samozřejmě předpokladem je, že vzdálenostní rozdíl mezi jednotlivými shluky je mnohem větší, než mezi prvky (přímkami), které patří do jednoho shluku.

Hledáme tedy iteraci uvedeného algoritmu, ve kterém nastane skokové zvýšení poměru rozdílu dvou předchozích vzdáleností – čili hledáme stav, ve kterém je poměr mezi aktuálním a předešlým kritériem maximální.

Protože UMF vyžaduje viditelnou velikost podoblasti 4×4 , je minimální počet výsledných shluků (přímek) 5.

Ukázka výstupu shlukování je na obrázku 5.6.



Obrázek 5.6: Ukázka shlukování finálních přímek. Finální přímký jsou zobrazeny světle modře.

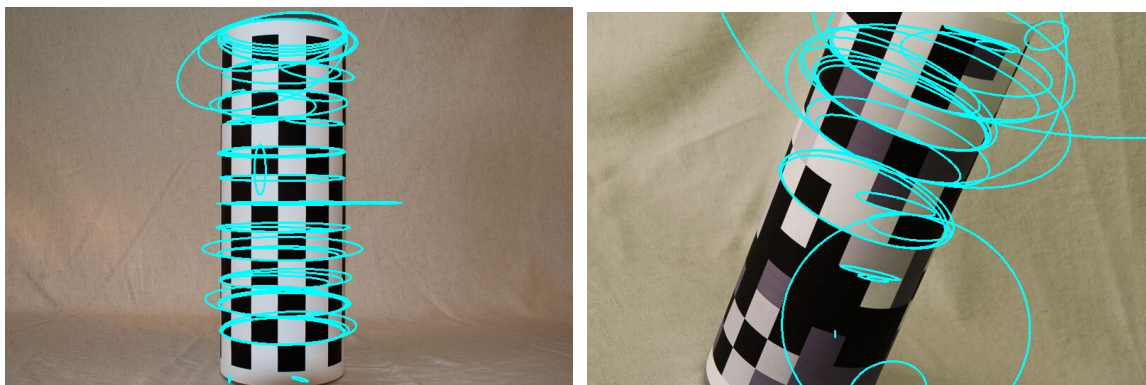
5.8 Detekce elips na plášti

Dosud jsme se zabývali přímkami tvořící plášť. V tomto kroku se z druhé skupiny přímek z předchozího kroku ze sekce 5.5 hledají parametry elips tvořící prstence na plášti. Parametry elips jsou určeny z jednotlivých skupin bodů pomocí algoritmu *Direct least-squares fitting of ellipses* podrobněji popsaného v kapitole 4.

Nalezené elipsy jsou zpravidla zatíženy poměrně velkým počtem chybně určených elips. Tento problém je způsobem nejvíce následujícími vlivy:

- Nepřesností zdrojových bodů
- Projekcí elips v prostoru – tuto zanedbáváme, jak je popsáno v kapitole 4
- Problém je i to, že hledáme elipsy pouze na základě bodů z přilehlé strany a máme tak k dispozici pouze polovinu elipsy

O korekci chybně detekovaných elips se stará následující krok detekce 5.9. Výsledek detekce všech potenciálních elips je na obrázku 5.7.



Obrázek 5.7: Ukázka všech detekovaných elips. Elipsy jsou zobrazeny světle modře.

5.9 Nalezení parametrů válce pomocí elips a algoritmu RANSAC

Nalezené elipsy z předchozího kroku jsou, jak bylo uvedeno, zatížené značnými chybami. Pro nalezení hledaných parametrů válce je tedy zapotřebí použití algoritmu RANSAC, který dokáže určit nejvhodnější elipsy k výpočtu parametrů. Algoritmus RANSAC je použit ve stejné formě jak je popsán v sekci 4.2.

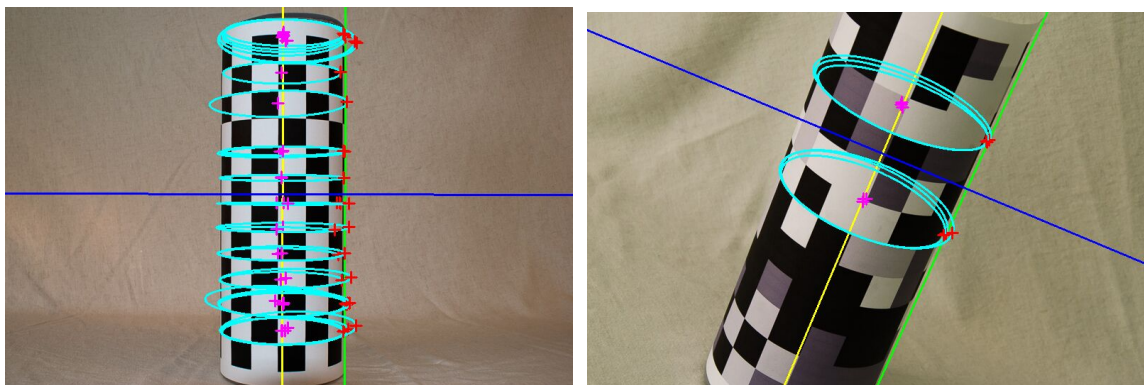
Hledané parametry válce jsou tyto:

- Osa válce, která prochází středy všech elips tvořících prstence na válci. Ve skutečnosti nalezená přímka nemusí být osou válce, ale pro potřeby algoritmu toto zjednodušení situace postačuje. Více informací o problému je v kapitole 4
- Přímka, která je rovnoběžná se všemi hlavními poloosami elips. V tomto případě zanedbáváme skutečnost, že i hlavní poloosy elips směřují do vlastního úběžníku, jak jsme popsali v kapitole 4
- Přímka udávající sklon projekce válce v reálné projekční rovině

Modely tří hledaných přímek jsou definovány pomocí jedné náhodně zvolené elipsy tímto způsobem:

- Model osy válce je definován jako přímka určená úběžníkem přímek na plášti a středem vybrané elipsy
- Model přímky určující rovnoběžnost všech hlavních poloos elips je definován přímkou určenou hlavní poloosou definované modelové elipsy
- Model přímky udávající sklon válce je definován jako přímka určená úběžníkem a hlavním vrcholem elipsy

Nejúspěšnější model je určen nejvyšším skóre po součtu jednotlivých skóre všech elips, které jsou inliers modelu. Výsledek tohoto kroku je na obrázku 5.8.



Obrázek 5.8: Ukázka výstupu algoritmu RANSAC nad všemi elipsami. Elipsy, které vyhovují modelu jsou zobrazeny světle modře. Vínově jsou zobrazeny středy elips a červeně jejich hlavní vrcholy. Žlutá čára odpovídá rotační ose válce, zelená čára odpovídá přímce, která udává sklon válce, a modrá čára odpovídá přímce udávající směrnicí úhlu hlavních poloos elips. Všimněte si, že s porovnáním s šachovnicovým vzorem poskytují šedotonové UMF pouze velmi malou hranovou informaci, ale i ta stačí k dostatečně přesnému určení parametrů válce.

5.10 Hledání parametrů parabol na základě parametrů válce

Dalším krokem detekčního algoritmu je hledání parametrů parabol a to ze stejných skupin bodů, ze kterých se v předchozích krocích hledaly parametry elips. Pro výrazné zjednodušení detekce parabol se využívá osa válce nalezená pomocí elips a algoritmu RANSAC z předchozím kroku 5.9.

Důvodem detekce parabol je fakt, že detekované elipsy z předchozího kroku trpí velkou chybovostí a při experimentování se často stávalo, že elipsy byly schopné správně určit pouze velmi malou část prstenců na plášti válce a bylo tedy nemožné jednoduchým způsobem z těchto dat vypočítat zbývající chybějící elipsy. Avšak osu rotace válce a ostatní parametry modelu RANSAC lze zpravidla určit ze získaných elips velmi dobře.

Známe-li parametry osy rotace válce, pak tuto osu můžeme s užitím transformací zaměnit s osou y . To umožňuje využít kanonické rovnice 5.1 pro hledání elipsy, jejíž osa je rovnoběžná s osou y obrazu, tato rovnice má tyto výhody oproti obecné rovnici kuželoseček.

- Výsledkem rovnice je vždy korektní parabola
- Rovnice má pouze tři parametry (souřadnice Vrcholu a parametr paraboly) oproti šesti parametrům obecné rovnice kuželoseček.

$$(x - x_0)^2 = p(y - y_0) \quad (5.1)$$

Tato problematika je podrobněji rozepsána v kapitole 4.

Také můžeme bez ztráty přesnosti předpokládat, že vrchol paraboly leží právě na ose rotace, ve skutečnosti leží na přímce na plášti, kolmé k projekční rovině. Ale tyto dvě přímky lze, při běžné perspektivě obrazu, bez dopuštění se znatelné chyby zaměnit. Takto získáme jednu souřadnici vrcholu. Ve finále tedy hledáme pouze dva parametry upravené kanonické rovnice 5.2.

$$x^2 = p(y - y_0) \Rightarrow y = \frac{1}{p}x^2 + y_0 \quad (5.2)$$

Parametry rovnice hledáme ze zadaných bodů pomocí metody nejmenších čtverců [5]. Problém můžeme definovat soustavou rovnic 5.3, kde $c_1 = \frac{1}{p}$ a c_0 jsou hledané parametry. Výrazy y_n a x_n jsou jednotlivé zdrojové body, kterými chceme proložit hledanou parabolou.

$$\begin{aligned} y_0 &\doteq c_0 + c_1 x_0^2 \\ y_1 &\doteq c_0 + c_1 x_1^2 \\ &\vdots \\ y_n &\doteq c_0 + c_1 x_n^2 \end{aligned} \quad (5.3)$$

Tento fakt lze zapsat maticově, tak jak je uvedeno v rovnici 5.4.

$$\mathbf{y} \doteq \mathbf{Z}\mathbf{c}, \text{ kde } \mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} 1 & x_0^2 \\ 1 & x_1^2 \\ \vdots & \vdots \\ 1 & x_n^2 \end{pmatrix}, \text{ a } \mathbf{c} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \quad (5.4)$$

Nyní hledáme takové parametry c_0 a c_1 , pro které je součet druhých mocnin rozdílů levých a pravých stran soustavy rovnic 5.3 minimální. Tento součet je znázorněn rovnicí 5.5.

$$\sum_{i=0}^n (y_i - c_0 - c_1 x_i^2)^2 \quad (5.5)$$

Nyní můžeme napsat rovnici řešení 5.6 pomocí výše definovaných matic 5.4 a z této rovnice vyjádříme hledaný vektor \mathbf{c} .

$$\mathbf{Z}^T \mathbf{Z} \mathbf{c} = \mathbf{Z}^T \mathbf{y} \Rightarrow \mathbf{c} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y} \quad (5.6)$$

Pomocí výše popsané metody nalezneme parametry paraboly pro každou skupinu bodů.

Aproximace elips parabolami je výhodné. Kromě potřeby hledat méně parametrů, což eliminuje chybovost, je výhodné nahradit elipsy parabolami především proto, že umožňují nalezení horizontu, jak bylo uvedeno v kapitole 4. Díky tomuto je možné lépe dopočítat chybějící paraboly a celkově zpřesnit vypočtené parametry parabol.

Jediným problémem je skutečnost, že prstence na válci tvoří skutečně elipsy a nikoli paraboly. Toto je také důvod proč přímá detekce parabol, bez znalosti osy rotace válce nemá smysl. Jsou-li však známy parametry osy válce, jsou nalezené paraboly schopné velmi dobře elipsy aproximovat, což bylo experimentálně potvrzeno. Větší chyby se paraboly dopouštějí pouze u hlavních vrcholů elips (na krajích prstenců), kde se křivka elipsy prudce stáčí. Zde se však dopouštějí velké chybovosti i samotné elipsy, proto je tento nedostatek akceptovatelný.

Aby bylo možné provést předchozí zjednodušení a hledat jen dva parametry, je potřeba transformovat zdrojové body podle rotační osy válce tak, že zaměníme osu y obrazu za osu rotace. Transformace bodů je definována transformační maticí, která se skládá s afinních transformací posunu do počátku souřadnic a rotace o úhel osy rotace vůči ose y obrazu. Matice se počítá pouze jednou a je stejná pro všechny body, takže je transformace poměrně

efektivní. Vypočtena je i zpětná transformační matice pro transformaci bodů respektive parabol zpět.

Rovnice 5.7 ukazuje příklad výpočtu transformačních matic. Kde α značí úhel mezi osou válce a osou y obrazu a parametry xd a yd určují posunutí na ose x respektive y do počátku souřadnic.

$$\text{Rotace: } \mathbf{R} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.7)$$

$$\text{Translace: } \mathbf{R} = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix}$$

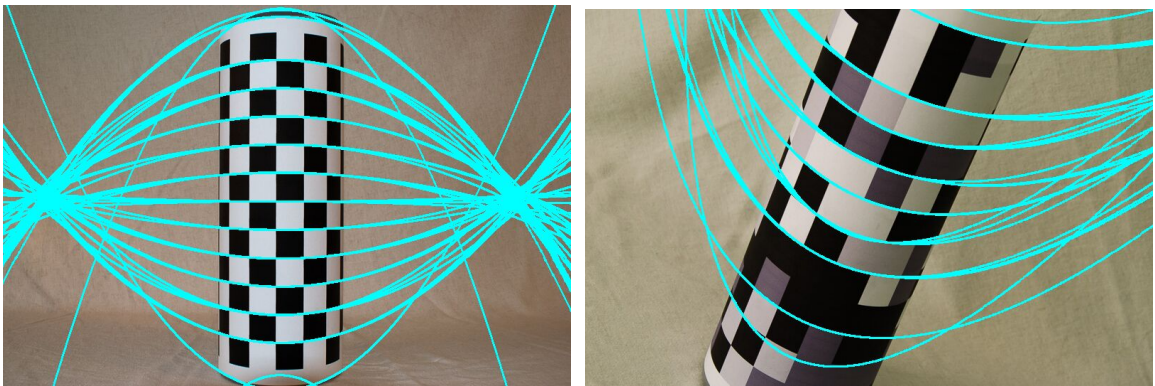
$$\text{Výsledná transformace: } \mathbf{M} = \mathbf{RT}$$

Poznamenejme, že pokud chceme body transformovat pomocí uvedených matic je potřeba body nejprve převést do homogenních souřadnic.

Algoritmus nalezení parabol se tedy skládá ze dvou respektive tří následujících částí:

- Transformace zdrojových bodů podle rotační osy válce
- Nalezení dvou hledaných parametrů parabol pomocí metody nejmenších čtverců
- Při zobrazení parabol, nebo při hledání průsečíků s nimi je nutné jejich body transformovat zpět

Ukázka nalezených parabol je na obrázku 5.9.



Obrázek 5.9: Ukázka všech nalezených parabol. Paraboly jsou zobrazeny světle modře.

5.11 Nalezení horizontu pomocí parabol a algoritmu RANSAC

I přesto, že jsou paraboly značně méně zatíženy chybami než elipsy, je zpravidla nutné odstranění několika outliers. Samotný algoritmus je opět implementován podle popisu v kapitole 4.2.2. Model pro tento RANSAC má dvě části.

Prvním kritériem je vzdálenost od přímky ve 2D prostoru, která je definovaná pomocí dvojice náhodně vybraných parabol. Z každé paraboly je definován bod, který má na ose x hodnotu nalezené souřadnice vrcholu paraboly z předešlého kroku. Na ose y je vynesena hodnota parametru paraboly.

Toto kritérium vychází z experimentálně zjištěné skutečnosti, že parametry parabol se lineárně mění se vzdáleností od horizontu, kde je hodnota parametru nulová.

Vhodně zvolený práh určuje vzdálenost od takto vniklé přímky, do které jsou stejně definované body z ostatních parabol považovány za inliers. Toto kritérium má především za úkol eliminovat paraboly se zjevně špatným rozevřením nebo natočením.

Druhým a důležitějším kritériem je vzdálenost průsečíků na horizontu. Pomocí náhodně zvolené dvojice parabol, lze definovat horizont jako přímku procházející dvojicí průsečíků těchto elips. Zde je potřeba dávat pozor a testovat, jestli má zvolená dvojice parabol právě dva průsečíky. Víme-li, že jsou osy všech parabol shodné mohou nastat tyto možné varianty:

- Žádný průsečík – paraboly nemají společný vrchol a zároveň mají shodný parametr zakřivení. Nebo mají rozdílné parametry zakřivení a parabola dále od středu má stejné znaménko a větší parametr udávající zakřivení. A nebo mají různé znaménko a parabola otevřená směrem dolů je pod parabolou otevřenou směrem nahoru.
- Jeden průsečík – paraboly mají společný vrchol, ale rozdílný parametr zakřivení.
- Paraboly jsou shodné a tedy mají nekonečně mnoho průsečíků.
- Dva průsečíky – paraboly nemají společný vrchol a zároveň mají rozdílné parametry zakřivení a parabola dále od středu má stejné znaménko a menší parametr udávající zakřivení, nebo mají rozdílné znaménko parametru a parabola otevřená směrem nahoru je pod parabolou otevřenou směrem dolů. Tento stav očekáváme a pouze tato dvojice parabol může vytvářet model.

Mají-li paraboly dva průsečíky, pak tato dvojice bodů definuje modelový horizont. Vzdálenost průsečíku ostatních parabol s horizontem vůči referenčnímu průsečíku dvojice vybraných parabol je prahována pevným prahem, který rozhoduje zda je daná parabola vyhovující inlier.

Rovnice 5.8 ukazuje způsob výpočtu průsečíku paraboly a přímky, který se využívá při výpočtu druhého kriteria.

$$\text{Přímka} : ax + by + c = 0 \quad (5.8)$$

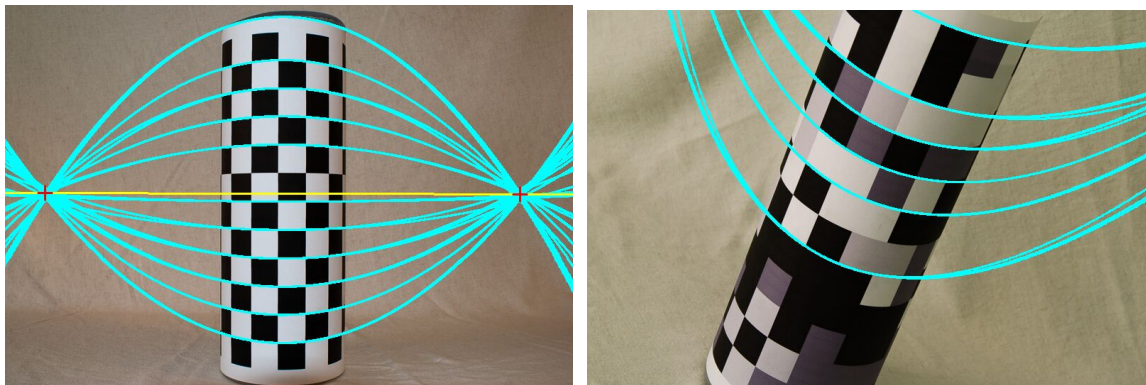
$$\text{Parabola} : y = px^2 + y_0$$

$$\text{Výsledná kvadratická rovnice průsečíku} : (bp)x^2 + (a)x + (by_0 + c) = 0$$

Názorná ukázka nalezení horizontu pomocí elips a algoritmu RANSAC je na obrázku 5.10.

5.12 Přepočítání parametrů parabol na základě společného průsečíku na horizontu

Je-li znám průsečík všech parabol na horizontu (viz. sekce 5.11) a zároveň jedna ze souřadnic vrcholu, zbývá pro určení paraboly pouze jeden parametr a tím je druhá souřadnice



Obrázek 5.10: Ukázka horizontu a vhodných parabol nalezených pomocí algoritmu RANSAC. Modelu vyhovující paraboly jsou zobrazeny světle modře. Červeně jsou zobrazeny průsečíky parabol na horizontu a žlutě je zobrazen nalezený horizont.

vrcholu. Z důvodu nepřesnosti zdrojových bodů odstraní RANSAC algoritmus značný počet nevyhovujících parabol. Avšak tyto body, lze poměrně úspěšně použít k přepočítání druhé souřadnice vrcholu původní paraboly, která je již mnohem přesnější.

Podobně jako v sekci 5.10 je využita metoda nejmenších čtverců k nalezení takové paraboly, jejíž vrchol leží na středové přímce válce a zároveň prochází definovaným bodem na horizontu. Oproti předchozímu hledání parabol však nyní hledáme pouze jeden parametr. Vyjdeme z rovnice 5.2. Známe-li bod, kde parabola prochází horizontem, lze z rovnice vyjádřit parametr y_0 podle rovnice 5.9.

$$y_h = \frac{1}{p}x_h^2 + y_0 \Rightarrow y_0 = y_h - \frac{1}{p}x_h^2 \quad (5.9)$$

Kde x_h a y_h jsou známé parametry bodu na horizontu. Nyní dosadíme do původní rovnice 5.9 a dostáváme finální rovnici, podle které budeme hledat parametr p 5.10.

$$y = \frac{1}{p}x^2 + y_h - \frac{1}{p}x_h^2 \Rightarrow y - y_h = \frac{1}{p}(x^2 - x_h^2) \quad (5.10)$$

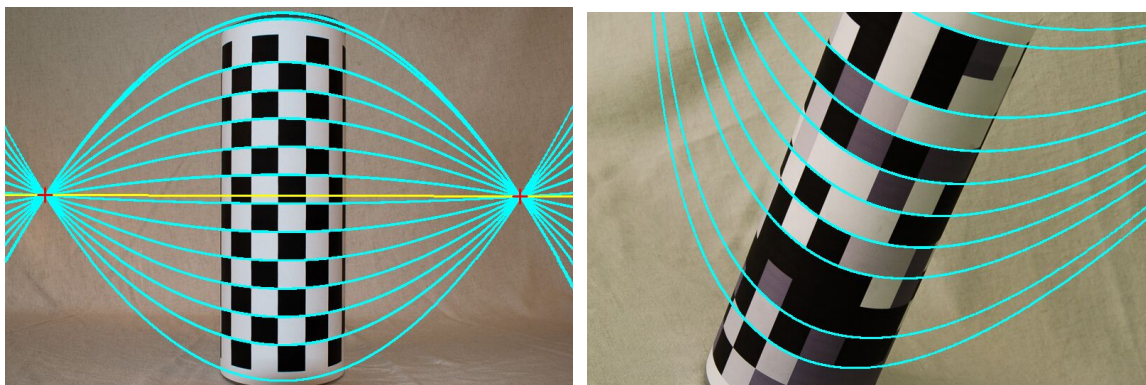
Dále můžeme stejným způsobem jako v kroku 5.10 vytvořit matice pro výpočet. Nastavení matic pro výpočet je znázorněno rovnicí 5.11. Samotný výpočet poté provedeme opět stejným způsobem a získáme optimální parametr p . Parametr y_0 získáme jednoduchým dosazením nalezeného p .

$$\mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} (x_0^2 - x_h^2) \\ (x_1^2 - x_h^2) \\ \vdots \\ (x_n^2 - x_h^2) \end{pmatrix}, \mathbf{a} \mathbf{c} = (c_0) \quad (5.11)$$

Ukázka nalezených parabol je na obrázku 5.11.

5.13 Shlukování podobných parabol

V této části se sloučí podobné paraboly zřejmě určující stejný prsteneček do jedné reprezentující paraboly, která je kombinací všech členských parabol shluku pomocí váženého průměru



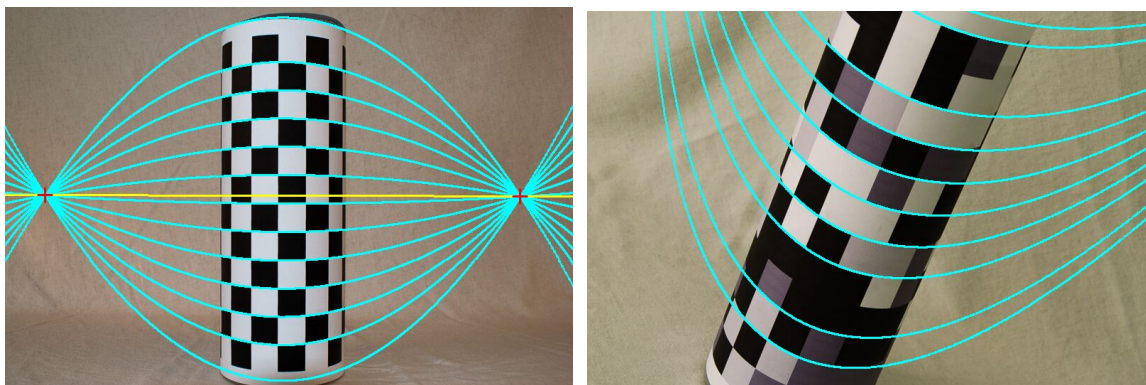
Obrázek 5.11: Ukázka přepočítaných parabol na základě společného průsečíku na horizontu. Paraboly jsou zobrazeny světle modře. Červeně jsou zobrazeny průsečíky parabol na horizontu a žlutě je zobrazen nalezený horizont. Všimněte si, že ačkoli algoritmus RANSAC v předešlém kroku využil znatelně méně vhodných parabol, tak díky přepočtu na základě znalosti získaného průsečíku parabol na horizontu jsme schopni rekonstruovat a upravit téměř všechny nalezené paraboly.

podle skóre jednotlivých parabol.

Podobně jako u shlukování přímek se jedná o shlukování na neznámý počet shluků a využíváme opět metody hierarchického shlukování 4.2.

Kritérium podobnosti jednotlivých elips je zde vzdálenost jejich vrcholů.

Ukázka výsledných parabol je na obrázku 5.12.



Obrázek 5.12: Ukázka výstupu po shlukování podobných parabol. Paraboly jsou zobrazeny světle modře. Červeně jsou zobrazeny průsečíky parabol na horizontu a žlutě je zobrazen nalezený horizont.

5.14 Doplnění chybějících parabol

Po provedení všech předchozích kroků se v mnoha případech stává, že pro jeden a více prstenců neexistuje žádná parabola, která by jej popisovala. V této části detektoru je představen algoritmus, který se pokouší vzniklé díry zaplnit a dopočítat chybějící paraboly.

Prakticky je potřeba dopočítat pouze pozici vrcholu hledané paraboly, všechny ostatní parametry paraboly lze vypočítat ze znalosti společných průsečíků na horizontu a parametrů osy rotace.

Algoritmus předpokládá, že existuje alespoň jedna trojice prstenců, kterou definují právě tři těsně nad sebou ležící paraboly. Tedy, že mezi parabolami neleží ani jeden prsteneček, který není definován žádnou parabolou. V případě, že tato podmínka není splněna, algoritmus selže a tuto skutečnost není schopen odhalit.

Algoritmus se snaží doplnit chybějící paraboly pomocí skutečnosti, že rozdíly dvou, vedle sebe ležících vzdáleností je vždy stejný. Pravidlo lze zapsat tak, že pro libovolné tři bezprostředně vedle sebe ležící paraboly platí pravidlo 5.12, kde p_x označuje parabolu, indexy i, j určují výchozí pozici pro libovolný prsteneček a d je hledaný rozdíl, který je vždy stejný.

$$|p_i - p_{i-1}| - |p_{i-1} - p_{i-2}| = |p_{j-1} - p_{j-1}| - |p_{j-1} - p_{j-2}| = d \quad (5.12)$$

Pravdivost této hypotézy byla empiricky testována na značné množině příkladů.

Algoritmus funguje následujícím způsobem:

- Algoritmus iteruje přes všechny trojice nad sebou ležících parabol a vypočítá parametry parabol
- Pomocí rovnice testuje, zda na určených pozicích leží do určité zvolené vzdálenosti paraboly
- Rovnice, do které padne nejvíce parabol se stane referenční a jedna z trojice počátečních určí výchozí bod
- Parametry rovnice jsou ještě zpřesněny přepočítáním ze všech korektních elips pomocí průměrování
- Následně pomocí rovnice doplníme na souřadnice paraboly. Pokud je na dané souřadnici parabola dostatečně blízko, použijeme ji místo dopočítané paraboly a stane se zní nová referenční pozice

Samozřejmě platí, že čím více správných elips má algoritmus k dispozici, tím bude výsledek lepší a použitelnější. Při malém procentu korektních elips algoritmus často selhává.

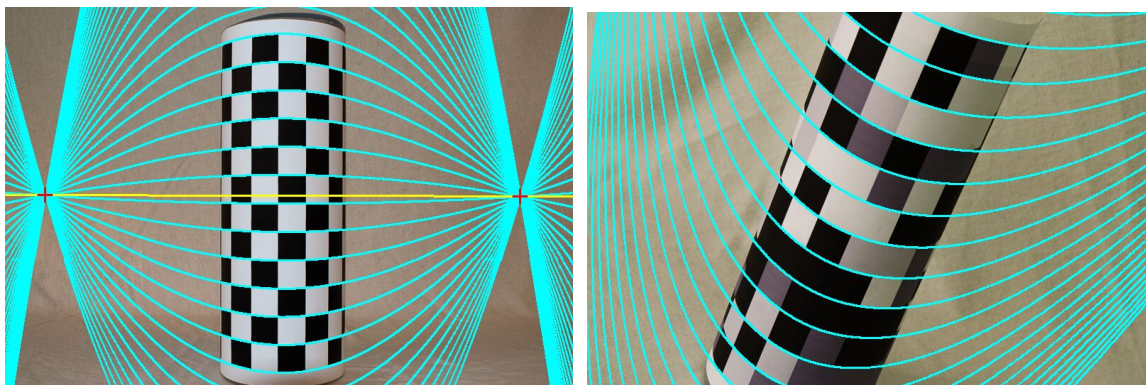
Ukázka doplněných parabol je na obrázku 5.13.

5.15 Vytvoření výsledné mřížky definující nalezené středy čtverců Uniform Marker Fields

Toto je poslední krok celého algoritmu detektoru. Vstupem jsou nashlukované přímky a doplněné paraboly z předchozích kroků.

Nejdříve je potřeba nalézt parametry přímek a elips, který leží mezi jednotlivými nejbližšími dvojicemi. Přímky se vypočtou zprůměrováním parametrů dané dvojice. U parabol se definuje poloha vrcholu, přesně uprostřed dané dvojice a zbylé parametry se dopočítají z průsečíků na horizontu podobně jako v kroku 5.14.

U takto vzniklých parabol a přímek se hledají vzájemné průsečíky vyřešením rovnice 5.8. Zde má význam pouze průsečík, který je blíže vrcholu paraboly druhý průsečík je zpravidla zcela mimo viditelný obraz.

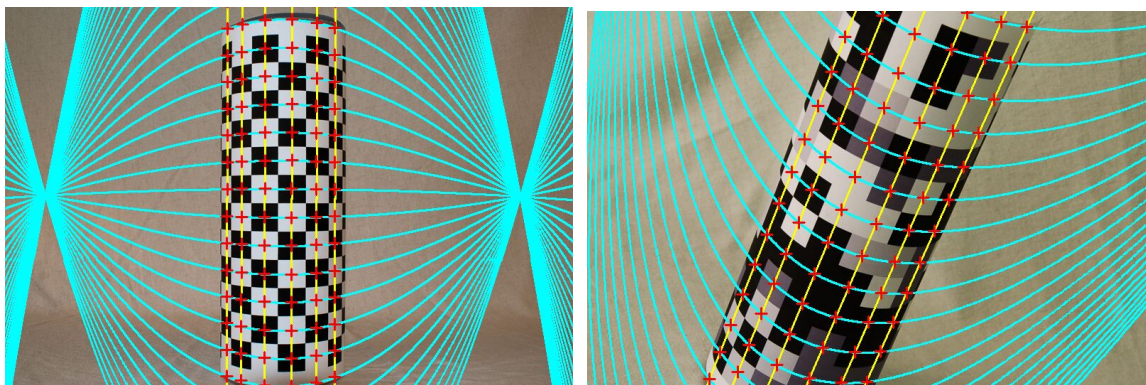


Obrázek 5.13: Ukázka výstupu po doplnění parabol. Paraboly jsou zobrazeny světle modře. Červeně jsou zobrazeny průsečíky parabol na horizontu a žlutě je zobrazen nalezený horizont. Porovnejte obrázky s výsledkem předchozího kroku. výsledek doplnění parabol je znatelný především na druhém obrázku.

Poznamenejme, že přímky se musí pomocí dříve vytvořené transformační matice transformovat do prostoru parabol. Poté je možno nalézt průsečíky a tyto průsečíky je nutné transformovat zpět.

Průsečíky se poté ukládají do seznamu seznamů po řádcích. Dalším postupem je vyčtení hodnot na pozicích průsečíků z obrazu, jejich prahování na danou hodnotu a hledání vnořeného vzoru UMF. Tato část však již není implementována.

Ukázka výsledné mřížky je na obrázku 5.14.



Obrázek 5.14: Ukázka výsledné mřížky po vykonání celého algoritmu. Středové paraboly jsou zobrazeny světle modře. Středové přímky jsou zobrazeny žlutě. Výsledné body, které by měli korespondovat se středy polí šachovnice jsou zobrazeny červeně.

Kapitola 6

Testování detektoru

Tato část práce se zabývá testováním algoritmu. Bylo provedeno několik různých testování, které prověřili různé aspekty algoritmu. Rozměr testovacích obrázků je 800×530 pixelů. Při testování detekčních vlastností byla použita čtveřice různých vzorů – markerů.

Šachovnice Jedná se o běžnou šachovnici pravidelně se střídajících černých a bílých čtverců. Tento vzor nepředstavuje žádný marker, ale je velmi vhodný pro testování algoritmu, protože díky pravidelnosti odpadá vliv pseudonáhodné podoby markeru. Představuje ideální případ markeru a očekává se, že detekce pro něj bude probíhat nejlépe

Uniform Marker Field Černobílý marker mající vlastnosti popsány v kapitole 3. Tento marker obsahuje nejméně hranových informací, a proto se očekává, že pro něj dopadnou testy nejhůře.

Šedotónový UMF UMF v odstínech šedi, tak jak je popsán v kapitole 3.2.

Speciální šedotónový UMF Jedná se o speciální případ Šedotónového UMF, pro který platí, že nikdy nemá vedle sebe dvě stejně barevná pole ve směru osy válce. Kritické hrany pro detektor prstenců na válci (elips) má tedy k dispozici co nejvíce hranové informace. Tento předpoklad by měl zajistit lepší detekci elips a tedy i přesnější nalezení parametrů válce jako je jeho rotační osa a horizont.

Poznamenejme také, že při testování vzoru šachovnice a klasických UMF je potřeba použít vyšší nastavení prahu (konkrétní hodnota 25) pro detekci hran než u šedotónových typů UMF (konkrétní hodnota 15).

Při vyhodnocování úspěšnosti detekce daného snímku sledujeme následující prvky:

- Poměr korektně detekovaných přímků na plášti a celkového počtu viditelných přímků. Přímků na okraji válce, které jsou díky deformaci velmi blízko sobě, rozlišujeme jako různé pokud je vzdálenost mezi nimi alespoň deset pixelů.
- Poměr korektně detekovaných prstenců (parabol) a celkového počtu viditelných prstenců.
- Korektní detekce alespoň jedné podoblasti UMF o velikosti 4×4 . Jedná se o binární vlastnost s hodnotami úspěch nebo neúspěch. Tato vlastnost prakticky určuje, jestli byla detekce úspěšná či nikoli. Počet detekovaných prstenců a přímků pak definuje, do jaké míry úspěšná nebo neúspěšná byla.

Špatně detekované přímký nebo prstence nesledujeme. V případě, že narušují správnou detekci, tak se projeví, při testování nalezení podoblasti UMF. Pravděpodobnost, že náhodně vytvoří falešnou nekorektní podoblast, je velmi malá.

Dalším prvkem, který se nabízí sledovat, jsou souřadnice správně určených středů vzoru markeru. Tato informace, však nemá vypovídající hodnotu v případě, že byly jednotlivé přímký a prstence detekovány zcela špatně. Detekce středů vzoru markeru je v takovém případě takřka náhodná. Proto bylo jako měřítko úspěšnosti detekce raději zvoleno právě pozorování detekce hran, které tvoří markery.

Kromě testování úspěšnosti a vlastností detekčního algoritmu se jedna část kapitoly věnuje výpočetní rychlosti algoritmu. Přesto, že optimalizace rychlosti detekce nebyla primárním cílem práce, mohou získané údaje vypovědět o možnostech optimalizace algoritmu v rámci další budoucí práce.

6.1 Celkové testování

V tomto testování byl použit pro každý typ markeru vzorek nejméně padesáti různých snímků. Pro každý typ markeru byla vygenerována šestice unikátních příkladů. Uvedené údaje neplatí pro vzor šachovnice, kde bylo použito jen dvacet testovacích snímků a dva příklady vzoru, které se vzájemně liší počtem a velikostí jednotlivých políček šachovnice. Vzorek šachovnice je vždy stejný a více testů by nemá smysl.

Každý snímek tvoří unikátní scénu, kde je válec s markerem v různém natočení a různých vzdálenostech od kamery. Na každém snímku je vždy pouze jeden válec s nepoškozeným vzorem na plášti a bez výrazných rušivých vlivů pozadí. Snímky také nebyly úmyslně rozmazány a poznamenány jinou chybou nebo šumem.

Cílem tohoto testu je ukázat schopnost algoritmu detekovat válec na nepoškozených datech v různých případech, natočeních, sklonech, úhlech pohledu a podobně. Tento test je prakticky nejdůležitější ze všech provedených testů. Prokazuje celkové detekční schopnosti algoritmu a zároveň ukazuje, který typ markeru je pro detekci na válci nevhodnější. Shrnuté výsledky jsou v tabulce 6.1.

Vzor	Úspěšnost [%]	Přímký [%]	Prstence [%]
Šachovnice	100	83	99
UMF	40	55	56
Šedotónové UMF	85	85	89
Spec. Šedotónové UMF	96	92	98

Tabulka 6.1: Tabulka s výsledky celkového testování. Úspěšnost je určena celkovým počtem testovacích dat a dat, kde byl správně nalezena alespoň jedna podoblast UMF. Přímký a prstence jsou určeny počtem viditelných a korektně nalezených přímek nebo prstenců.

Celkové testování potvrdilo předpoklad. Šachovnicový vzorek dává celkově nejlepší detekční výsledky. Úspěšnost je 100 %, což znamená validní výsledek ve všech případech. Detekce prstenců přesahuje 99 %. Největší problémy algoritmu činí detekce přímek na plášti. Tento problém se projevuje u všech typů markerů.

Za šachovnicí těsně následuje speciální Šedotónový UMF, jehož výsledky jsou také velmi dobré a úspěšnost se drží nad solidními 95 %. Detekci prstenců má od šachovnicového vzoru pouze o procento menší a v detekci přímek na plášti šachovnicový vzorek dokonce překonává.

Oproti vzoru šachovnice je tento typ markeru již reálně použitelný v aplikacích, proto je tento výsledek důležitější než u šachovnice.

Klasický šedotónový UMF již zaostává o zhruba 10 % jak v úspěšnosti detekce, tak v detekci prstenců. Je vidět, že úprava speciálních šedotónových UMF má svá opodstatnění a je schopná celkově zlepšit detekci.

Klasický UMF výrazně zaostává za ostatními. Z testování vyplývá, že zřejmě neobsahuje dostatečné množství hranové informace a tedy není pro detekci na válci vhodným typem markeru. Z tohoto důvodu je klasický černobílý UMF v dalších testech vynechán.

Test také poukázal na to, že detekce prstenců má větší úspěšnost, než detekce přímek na plášti. Detekce přímek, jak již bylo zmíněno, problematicky shlukuje přímký na okrajích válce a často sdruží i přímký patřící dvou různým hranám. Dalším problémem je poměrně silná závislost shlukování na správně určených parametrech válce pomocí elips. Eliminace tohoto problému může být tématem pro zlepšení algoritmu.

6.2 Testování vlivu úhlů pohledu na detekci

V rámci tohoto testu byl pozorován vliv úhlu pohledu na úspěšnost detekce. Pro všechny výše uvedené typy markerů, kromě klasického UMF, byla postupně pořízena sada fotografií o různém úhlu pohledu na marker.

Celkový počet testovaných úhlů pohledu je deset. Rozsah úhlů pohledu je od 0° do zhruba 80° . Jednotlivé skoky mezi úhly činí tedy přibližně 9° . Z důvodu nedostatečné přesnosti vybavení použitého pro testování se však skoky mohou vůči sobě částečně lišit. Úhel 80° přibližně odpovídá úhlu pohledu, ve kterém už není marker dostatečně vidět a detektor selže ve všech případech. Jednotlivé úhly si pro každý typ testovaného markeru odpovídají, aby bylo možno provést porovnání mezi nimi.

V každém úhlu pohledu pro jednotlivé typy markerů byla vytvořena skupina fotografií s rozdílnými příklady daného typu markeru a rozdílnými hustotami polí. Jejich hodnoty byly pro daný úhel pohledu zprůměrovány, aby se eliminoval vliv konkrétního vzoru a projevil se pouze typ daného typu markeru. Pro vzor šachovnice se testuje opět jen dvojice šachovnic s rozdílnou hustotou polí.

Data jsou vyobrazena v přehledných grafech, kde jsou zakresleny sledované vlastnosti detekce pro každý úhel pohledu. Sledované vlastnosti jsou stejně jako v předchozím testu – konkrétně úspěšnost nalezení alespoň jedné podoblasti UMF a počet korektně detekovaných přímek a prstenců.

6.2.1 Testování vlivu úhlů pohledu na detekci šachovnice

Testováním šachovnicového vzoru zkusíme, jak nejlépe může algoritmus fungovat při změnách úhlu pohledu. Díky vzoru šachovnice odpadávají jakékoli vlivy zvoleného markeru. Tímto testem, tedy získáme nejlepší možný výsledek jakého je algoritmus schopen dosáhnout. Tento výsledek poté můžeme porovnat s testem na reálně použitelných typech UMF.

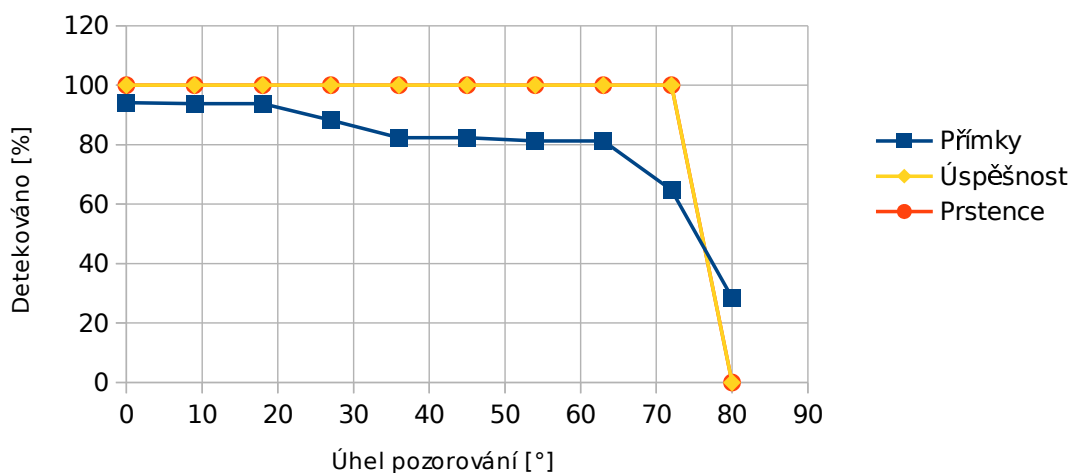
Výsledky pro vzor šachovnice jsou vyneseny do grafu 6.1.

Jak se předpokládalo šachovnice dopadla nejlépe z trojice testovaných vzorů. Prstence jsou pomocí parabol detekovány korektně i extrémních úhlech. Stejně tak úspěšnost algoritmu dosahuje 100 % i v extrémnějších úhlech. Algoritmus selhává až ve chvíli, kdy úhel pohledu dosáhl kritické hranice a marker na plášti válce je již nerozpoznatelný.

U přímek dělají problém krajní málo viditelná pole, která jsou vlivem deformace příliš malá. Za tento problém může především krok shlukování, který neodhalí rozdílnost krajních

Test vlivu úhlu pozorování

Šachovnice



Obrázek 6.1: Graf s výsledky testování vlivu úhlu pohledu pro vzor šachovnice. Poznámka: Procentuální úspěšnost detekce prstenců a podoblasti UMF se zcela překrývá. Nejdůležitější sledovaným kritériem je úspěšnost, která udává zda, lze nalézt alespoň jednu podoblast UMF.

přímek. S tímto problémem se však potýkají všechny testované vzory.

6.2.2 Testování vlivu úhlů pohledu na detekci Šedotónového UMF

Dalším testovaným vzorem na vliv úhlu pohledu je Šedotónový UMF. Klasický černobílý UMF jak bylo zmíněno výše přeskakujeme kvůli příliš špatným výsledkům už v úvodním celkovém testu.

Výsledky jsou vyneseny na grafu 6.2.

Tento typ markeru dopadl, podle předpokladu, nejhůře z testované trojice. Ve větších úhlech pohledu algoritmus není schopen z poskytované hranové informace správně určit elipsy reprezentující prstence a zjistit tak potřebné parametry válce, na kterých stojí všechny následující kroky algoritmu. Toto byla nejčastější chyba, na které algoritmus selhával.

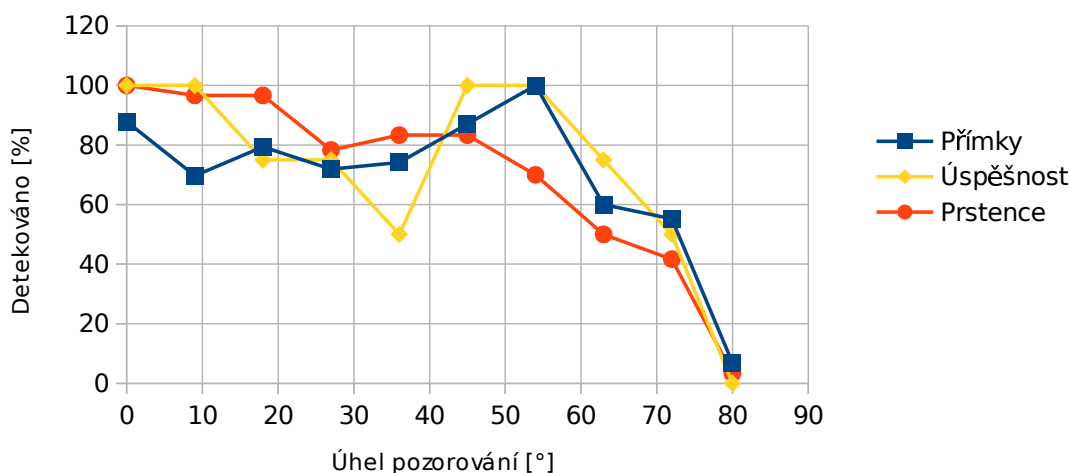
Nekonzistence výsledků poukazuje na silný vliv konkrétního vygenerovaného příkladu markeru. U běžných šedotónových UMF se hranová informace může mezi různými příklady poměrně silně měnit a především detekce prstenců je na množství hranové informace velmi citlivá.

6.2.3 Testování vlivu úhlů pohledu na detekci Speciálního Šedotónového UMF

Výsledky testu tohoto typu markeru jsou nejdůležitější, protože se jedná o reálně použitelný typ markeru. V tomto testu zaostává za ideální šachovnicí pouze nepatrně. Úspěšnost detekce podoblasti UMF se dokonce vyrovnává vzoru šachovnice. V úspěšnosti detekce prstenců a přímek je mírně horší, ale stále jsou výsledky o mnoho lepší než u běžného Šedotónového UMF.

Test vlivu úhlu pozorování

Šedotónové UMF



Obrázek 6.2: Graf s výsledky testování vlivu úhlu pohledu pro šedotónový UMF. Nejdůležitější sledovaným kritériem je úspěšnost, která udává zda, lze nalézt alespoň jednu podoblast UMF.

Výsledky jsou vyneseny na grafu 6.3.

Větší chybovost ve větších náklonech je možné vysvětlit úbytkem použitelné hranové informace. Čím větší je náklon tím méně použitelné hranové informace algoritmu zůstává a zvyšuje se počet špatně detekovaných šumových elips.

Dalším problémem je zmenšující se pozorovaná vzdálenost mezi jednotlivými prstenci při větších úhlech. V extrémních případech je vzdálenost tak malá, že je algoritmus vyhodnotí jako jeden prsteneček namísto dvou různých.

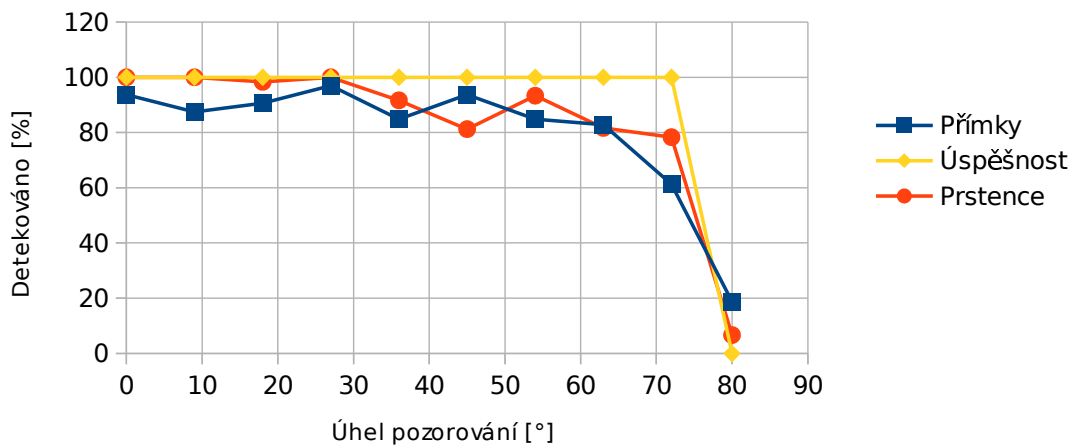
Celkově se, ale opět se tedy potvrdila hypotéza, že Speciální Šedotónové UMF jsou pro detekci na válci nejvhodnější z testovaných typů UMF.

6.2.4 Shrnutí testování vlivu úhlu pohledu

Z testování vlivu úhlu pohledu na úspěšnost detekce můžeme vyvodit několik závěrů.

- Potvrdila se hypotéza, že z testovaných UMF je Speciální šedotónový nejvhodnějším kandidátem na detekci na válci
- Algoritmus se ukázal jako poměrně robustní vůči různým úhlům pohledu. Sice se zmenšuje schopnost parabol přesného kopírování hran prstenců, ale přesto algoritmus selhává až v extrémních úhlech, kdy je vzor na plášti válce stěží rozpoznatelný i lidským okem. Snížení přesnosti vyplývá ze zmenšování poměru mezi hlavní a vedlejší poloosou elips, čili se elipsy blíží kružnici. Kružnice má největší chybovost při nahrazení parabolou, jak vyplývá z úvah v kapitole 4.
- Potvrdilo se zjištění z celkového testu a to, že největší slabinou algoritmu je správné nalezení přímek na plášti válce. Tento problém však s úhlem pohledu nesouvisí a vyskytuje se ve všech podmínkách.

Test vlivu úhlu pozorování Speciální Šedotónové UMF



Obrázek 6.3: Graf s výsledky testování vlivu úhlu pohledu pro speciální šedotónový UMF. Nejdůležitějším sledovaným kritériem je úspěšnost, která udává zda, lze nalézt alespoň jednu podoblast UMF.

Dalším pozorovaným jevem je snižující se přesnost detekce s rostoucím úhlem pohledu. Společně se zvyšujícím se úhlem pohledu, se zmenšuje poměr mezi hlavními poloosami elips, které tvoří prstence. Čím menší je poměr mezi poloosami elipsy, tím větší je chyba nahrazení elipsy parabolou, jak je ukázáno v kapitole 4. Toto vede ke zmíněné snížené přesnosti detekce prstenců a většímu sklonu k chybám při větších úhlech.

6.3 Testování počtu viditelných polí

Tento test má ukázat vliv počtu viditelných polí markeru na úspěšnost detekce a zároveň ukazuje i závislost úspěšnosti algoritmu na vzdálenosti od válce, protože čím blíže jsme válci, tím méně polí je vidět v zorném poli kamery a obráceně.

Výsledky jsou zobrazeny pro každý druh vzoru (markeru) ve zvláštní tabulce. Tabulky jsou v tomto případě přehlednější než grafy, je v nich možno přesně sledovat počty detekovaných prstenců a přímek.

Jednotlivé indexy v tabulkách mají tyto významy:

- ID 1 – matice 4×4 viditelných polí, což odpovídá pěti přímkám a pěti parabolám. Nejmenší smysluplná část přesně odpovídá velikosti jedné podoblasti UMF.
- ID 2 – matice 4×5 viditelných polí, což odpovídá šesti přímkám a pěti parabolám
- ID 3 – matice 5×4 viditelných polí, což odpovídá pěti přímkám a šesti parabolám
- ID 4 – matice 5×5 viditelných polí, což odpovídá šesti přímkám a šesti parabolám
- ID 5 – matice 6×6 viditelných polí, což odpovídá sedmi přímkám a sedmi parabolám
- ID 6 – matice 7×7 viditelných polí, což odpovídá osmi přímkám a osmi parabolám

- ID 4 – Celé válce

Každý index byl testován na čtveřici různých příkladů daného typu markeru tak, aby se projevil vliv konkrétního vzoru na úspěšnost detekce. Opět kromě vzoru šachovnice, kde toto nemá smysl. Jednotlivé získané hodnoty byly poté pro každý index seskupeny.

V tomto testu s zaměřujeme se na testování detekce malých částí markerů. Velké části markerů prakticky testujeme ve všech ostatních provedených testech.

6.3.1 Testování vlivu počtu viditelných polí na detekci vzoru šachovnice

Vzor šachovnice v testu opět uspěl nejlépe ze všech typů testovaných markerů a kromě problému s krajními přímkami detekoval úspěšně. I v tomto případě výsledky šachovnice definují nejlepší možný výsledek, jakého je algoritmus schopen při ideálním vzoru. Výsledek slouží především pro porovnání s následujícími testy Šedotónových UMF.

Celkové výsledky jsou v tabulce 6.2

Šachovnice									
ID	Úspěšnost			Přímky			Prstence		
	S	D	%	S	D	%	S	D	%
1	1	1	100	5	5	100	5	5	100
2	1	1	100	6	6	100	5	5	100
3	1	1	100	5	5	100	6	6	100
4	1	1	100	6	6	100	7	7	100
5	1	1	100	7	7	100	7	7	100
6	1	1	100	7	6	86	13	13	100
7	1	1	100	8	7	88	17	17	100
Souhrn	7	7	100	44	42	95	60	60	100

Tabulka 6.2: Tabulka testování vlivu počtu viditelných polí markeru na detekci pro vzor šachovnice. Kvůli jednotvárnosti výsledků se může zdát, že tento test a tabulka jsou zbytečné, ale slouží především k porovnání s následujícími testy Šedotónových UMF. Legenda: S – Skutečný správný počet, D – Počet detekovaný algoritmem

Test prokázal, že, je-li hranová informace dostatečná, algoritmus je schopen i pro malé počty viditelných čtverců vzoru pracovat velmi přesně.

6.3.2 Testování vlivu počtu viditelných polí na detekci Šedotónového UMF

Šedotónový UMF dopadl v rámci detekcí prstenců a přímek poměrně dobře. V úspěšnosti detekce podoblasti UMF však úplně propadl. Největším problémem v tomto testu byla opět špatná detekce elips pro prstence a následně špatný odhad parametrů válce. Paraboly jsou sice schopné se vyrovnat s případnou nepřesností nalezených parametrů válce, ale přímky jsou na správnosti nalezených parametrů závislé o mnoho více. Celková chybovost byla značná a výsledek 42%, je velmi špatný.

Také se prokázala velká závislost na konkrétním vzoru markeru. Některé poskytovali dostatek hranových informací a algoritmus fungoval dobře i pro velmi málo viditelných polí, pro některé naopak selhal i pro větší matici polí markeru.

Výsledky jsou v tabulce 6.3.

Šedotónové Uniform Marker Field									
ID	Úspěšnost			Přímky			Prstence		
	S	D	%	S	D	%	S	D	%
1	4	1	25	20	14	70	20	11	55
2	4	1	25	24	18	75	20	15	75
3	4	1	25	20	15	75	24	19	79
4	4	1	25	24	16	67	24	24	100
5	4	2	50	24	21	87	28	28	100
6	4	2	50	28	18	64	28	28	100
7	4	4	100	34	28	82	60	60	100
Souhrn	28	12	43	174	130	74	204	185	90,69

Tabulka 6.3: Tabulka testování vlivu počtu viditelných polí markeru na detekci pro vzor Šedotónové UMF. Legenda: S – Skutečný správný počet, D – Počet detekovaný algoritmem

6.3.3 Testování vlivu počtu viditelných polí na detekci Speciálních šedotónových UMF

V tomto testu, více než v kterémkoli jiném, se ukázaly výhody Speciálního Šedotónového UMF oproti klasickému Šedotónovému UMF. Více hranové informace pro detekci elips zajišťuje nesrovnatelně lepší výsledky úspěšnosti detekci.

Výslednými hodnotami se řadí těsně za výsledky ideální šachovnice. Výsledky jsou v tabulce 6.4.

Speciální Šedotónové Uniform Marker Field									
ID	Úspěšnost			Přímky			Prstence		
	S	D	%	S	D	%	S	D	%
1	4	4	100	20	20	100	20	20	100
2	4	3	75	24	22	92	20	20	100
3	4	4	100	20	20	100	24	24	100
4	4	4	100	24	24	100	24	24	100
5	4	4	100	24	24	100	28	28	100
6	4	4	100	28	28	100	28	28	100
7	4	4	100	34	28	82	60	60	100
Souhrn	28	27	96	174	166	95	204	204	100

Tabulka 6.4: Tabulka testování vlivu počtu viditelných polí markeru na detekci pro vzor Speciální Šedotónové UMF. Legenda: S – Skutečný správný počet, D – Počet detekovaný algoritmem

6.4 Testování vlivu okolní scény

Tento test má za úkol ukázat, jak moc je algoritmus robustní vůči vlivu pozadí scény. Je zaměřen na reálné použití markeru, proto na tuto vlastnost byl testován pouze Speciální

Šedotónový UMF. Je reálně použitelný (na rozdíl od vzoru šachovnice) se v předchozích testech podával nejlepší výsledky.

Výslednými hodnotami se řadí těsně za výsledky ideální šachovnice. Výsledky jsou v tabulce 6.5 a ukázky detekce ve scéně jsou na obrázcích D.4.

Vzor	Úspěšnost [%]	Přímký [%]	Prstence [%]
Spec. Šedotónové UMF	60	68	78

Tabulka 6.5: Tabulka s výsledky testování vlivu okolní scény. Úspěšnost je určena celkovým počtem testovacích dat a dat, kde byl správně nalezena alespoň jedna podoblast UMF. Přímký a prstence jsou určeny počtem viditelných a korektně nalezených přímek nebo prstenců.

Toto testování dopadla podle předpokladu nejhůře. Vliv okolní scény na úspěšnost algoritmu se ukázal jako poměrně značný. Celková úspěšnost pro testované Speciální Šedotónové UMF klesla až na hodnotu 60 %.

V několika případech nastalo, že algoritmus sice na počátku dobře určil dva hlavní směry, ale hned vzápětí špatně zaměnil směr přímek na plášti a směr elips (prstenců). Výsledek byl tedy zcela nekorektní.

Problémy nastávaly také ve shlukování, kde algoritmus nedokáže odhalit falešně detekovaná šumová data a tato data poté degradují výsledek algoritmu.

6.5 Testování rychlosti detektoru

Toto testování má cíl zjistit přibližnou rychlost algoritmu. Cílem práce je vytvořit a implementovat funkční algoritmus, který primárně ukáže možnosti detekování UMF na válci. Rychlost algoritmu tedy není primárním účelem a algoritmus ani není nijak zvlášť optimalizován. Přesto je informace o rychlosti algoritmu užitečná a to především pro zlepšování algoritmu v budoucnu.

Testy byly prováděny bez jakýchkoli nezbytných výstupů, tak aby se minimalizoval vliv režie a měřil se pouze čas samotného algoritmu. Testovací stroj bylo PC typu Desktop s CPU Intel i5 se čtyřmi jádry o frekvenci 3.10 GHz a paměti 8GB DDR3. Testovacím softwarem byl OS Arch Linux s jádrem 3.8.11 a knihovna OpenCV ve verzi 2.4.5. Výsledky jsou vypsány v tabulce 6.6.

Vzor	Průměr [ms]	Odchylka [ms]	Maximum [ms]	Minimum [ms]
Šachovnice	100	69	210	30
UMF	28	213	70	10
Šedotónové UMF	52	36	120	10
Speciální Šedotónové UMF	70	45	140	20

Tabulka 6.6: Tabulka rychlosti algoritmů pro různá vstupní data. Průměr, maximum a minimum jsou časy zpracování pro jeden snímek.

Real-time aplikace musí zvládat v reálném čase zpracovat alespoň 25 snímků za sekundu (to znamená maximálně 50 ms na jeden snímek). Průměrné časy reálně použitelných šedo-

tónových markerů se tomuto času poměrně úspěšně přibližují. Testování však probíhalo na poměrně výkonném stroji a například na mobilních zařízeních by výsledky byly pravděpodobně řádově horší. Detektor v aktuální implementaci tedy za real-time aplikaci považovat nelze, avšak je zde velký prostor pro optimalizaci a vylepšení algoritmu může být náplní další práce.

Z výsledků je také patrné, že rychlost běhu algoritmu není statická, ale naopak se poměrně značně liší. Rychlost zpracování snímku je závislá na počtu viditelných hran. Tento závěr je možné odvodit z průměrné rychlosti. Nejdelší dobu zpracování má prostá šachovnice, která obsahuje nejvíce hranové informace. Následuje speciální šedotónový UMF, který je uzpůsoben tak, aby obsahoval více hranové informace pro prstence než běžný šedotónový UMF. Následuje klasický šedotónový UMF a v průměru nejrychlejší je klasický černobílý UMF, který nabízí nejméně hranové informace.

Z předchozích testů detektoru však vyplývá, že čím více máme hranové informace, tím lépe a přesněji jsme schopni marker detekovat. Tyto dvě věci jdou tedy proti sobě a nalezení optimálního vyvážení se nabízí jako možné vylepšení algoritmu v rámci další práce.

6.6 Zhodnocení testů

Testování prokázalo, že pro správnou detekci je nutné dostatečné množství hranové informace. Potvrdila se tedy hypotéza zmíněná na úvodu této kapitoly. Klasický černobílý UMF dopadl už během úvodního testu jednoznačně nejhůře, takže v dalších testech už nebyl více použit pro testování. Šachovnicový vzor naopak skončil vždy nejlépe. Toto svědčí o správné funkčnosti algoritmu.

Také se prokázalo, že Speciální Šedotónový UMF je skutečně schopen znatelně vylepšit detekci oproti klasickému Šedotónovému UMF. Toto se nejvíce projevilo v testu vlivu počtu viditelných čtverců markeru, kde se více hranové informace pro detekci elips využilo nejvíce.

Testování vlivu úhlu pohledu prokázalo znatelnou závislost. Krajních úhlech pohledu kolísá jak přesnost, tak úspěšnost detekce. Toto se nejvíce projevuje u klasických Šedotónových UMF. Přesto je algoritmus schopen úspěšné detekce alespoň části markeru i v extrémnějších podmínkách.

Test na počet viditelných částí UMF podobně jako test vlivu úhlu pohledu. Největší problém s korektní detekcí měl algoritmus opět s klasickým Šedotónovým UMF, který v těchto extrémních podmínkách neposkytoval dostatek hranové informace.

Přesto obsahuje-li obraz dostatečnou hranovou informaci je algoritmus schopný velmi přesné detekce a nezáleží příliš na počtu viditelných oblastí markeru. Dokonce přímky na plášti se zpravidla detekují přesněji, kvůli větším vzdálenostem v obraze. Opačný problém nastává, je-li válec s markerem ve větší vzdálenosti a jednotlivé čtverce markeru jsou velmi malé. V tomto případě algoritmus často chybí kvůli špatnému shlukování podobných hran.

Největší problémy algoritmus prokázal při detekci v reálné scéně, která data zanáší množstvím šumových a falešných dat. Algoritmus nedokázal dostatečně filtrovat tato špatná data, která poté degradují výsledek. Zvýšení robustnosti v této oblasti může být také námětem budoucí práce.

Během testování byli zjištěny dva zásadnější problémy. Algoritmus hledání prstenců je postaven tak, že je pro daný snímek téměř 100 % úspěšný nebo naopak úplně selže. Téměř neexistují případy, kdy by selhal jen částečně. Tento problém vyplývá především z kroků shlukování a dopočítávání chybějících parabol. Možným řešením by bylo vylepšení kritéria shlukování a vylepšení algoritmu doplnění parabol například o odhad přesnosti.

Druhý problém se týká detekce přímek na plášti válce. Zde se problém týká časté nepřesnosti detekce hraničních přímek při závěrečné fázi – shlukování. algoritmus často není schopen odhadnout kdy se jedná dva různé shluky a kdy o jeden, protože vzdálenost mezi shluky na krajích válce je velmi malá. řešením by bylo zlepšení kritéria a případně i zesílený výchozí předpoklad minimálního počtu viditelných přímek na plášti.

Test rychlosti ukázal, že při optimalizaci, by algoritmus pravděpodobně mohl běžet na běžném PC i v reálném čase. Tato informace je velmi povzbudivá a dává příslib do dalšího vývoje algoritmu.

Testování ukázalo že je možné detekovat UMF na válci pomocí představeného algoritmu s úspěšností přesahující 90 % (platí pro Speciální Šedotónové UMF) a to s dobrou přesností, která přesahuje 90 % při detekci prstenců a přímek na plášti. Přesto algoritmus vykazuje některé špatné vlastnosti. Aby se mohl algoritmus reálně používat je potřeba zpravení těchto vlastností.

Kapitola 7

Závěr

V této kapitole je zhodnocena celá práce. Její přínos, výsledky a závěry. Také jsou zde uvedeny možnosti budoucího rozšíření a možné další směry vývoje detektoru.

7.1 Možnosti budoucí práce

Nejdůležitějším rozšířením algoritmu v budoucnu je určitě jeho zakomponování do stávajícího detektoru UMF, tak aby výstupem nebyla jen matice nalezených bodů, ale i detekovaná konkrétní podoblast UMF.

Tato práce měla prokázat především možnost detekce UMF na válci, algoritmus tedy není příliš optimalizován a v aktuální podobě není zcela vhodný pro ostré nasazení. RefaktORIZACE kódu a jeho optimalizace společně s lepším otestováním nejvhodnějších parametrů jednotlivých částí algoritmu, by přispěla ke znatelnému zrychlení kódu. Vzhledem k výsledkům měření rychlosti v tabulce 6.6 je velmi pravděpodobné, že by se algoritmus dokázal zrychlit i na úroveň real-time aplikace.

Dalšími možnostmi ke zlepšení je určitě oprava nebo alespoň minimalizace největších slabín algoritmu. Především se jedná o vylepšení, nebo úplné přepracování, algoritmu shlukování přímek, který se projevil jako největší zdroj chyb při testování. Druhou nejvíce chybující částí je krok dopočítání elips, který v některých komplikovaných případech dává zcela nekorektní výsledky. Možným řešením je přepracování rovnice pro dopočítávání parabol a lepší využití znalostí parametru válce.

7.2 Celkové zhodnocení práce

Hlavním cílem této práce je ukázat možnosti detekce UMF a vůbec obecných markerů založených na pravoúhlé mřížce vzorů na válci. Poznamenejme, že autor práce nenalezl žádnou práci, která by se zabývala problematikou detekce markeru na válci nebo jiném neplanárním povrchu. Tento cíl se povedlo splnit a testování prokázalo, že detektor je schopen pomocí představené metody korektně najít potřebné parametry válce i na neideálních testovacích datech. Nalezené parametry válce ve finále vedou k nalezení jednotlivých polí markeru.

Testování dále prokázalo, že klasický černobílý UMF není pro detekci na válci vhodný. Naopak Speciální Šedotónová varianta UMF se ukázala jako o mnoho vhodnější a jejich výsledky lze jistě považovat za velmi slibné.

Zásadním krokem celého algoritmu je myšlenka hledání horizontu a na něm společného průsečíku všech parabol, které definují jednotlivé prstence. Pravě tato myšlenka umožňuje

velmi přesnou detekci prstenců na válci a je jádrem celé práce. Dalším zásadním krokem je myšlenka detekování prstenců na válci jako částí elips a výpočtu parametrů válce pomocí těchto elips.

Poznatky a základní myšlenky navrženého algoritmu v této práci mohou v budoucnu posloužit jako solidní základ pro implementaci reálně použitelného detektoru markeru na válci.

Literatura

- [1] Azuma, R.: A Survey of Augmented Reality. *Presence*, ročník 6, č. 4, 1997: s. 355–385.
- [2] Bartsch, H.-J.: *Matematické vzorce*. SNTL, 1987.
- [3] Burns, J.; Laboratories, H.-P.; Mitchell, C.: *Coding Schemes for Two-dimensional Position Sensing*. HP Laboratories technical report, Hewlett-Packard Laboratories, Technical Publications Department, 1992.
URL <http://books.google.cz/books?id=gpDaGwAACAAJ>
- [4] Cai, W.; Yu, Q.; Wang, H.: A fast contour-based approach to circle and ellipse detection. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, ročník 5, 2004, s. 4686–4690 Vol.5, doi:10.1109/WCICA.2004.1342408.
- [5] Fajmon, B.; Růžičková, I.: *Matematika 3*. Fakulta elektrotechniky a komunikačních technologií VUT v Brně.
- [6] Fiala, M.: ARTag, a Fiducial Marker System Using Digital Techniques. In *CVPR*, 2005, s. II: 590–596.
URL <http://dx.doi.org/10.1109/CVPR.2005.74>
- [7] Fitzgibbon, M., A. W. and Pilu; Fisher, R. B.: Direct least-squares fitting of ellipses. ročník 21, č. 5, 1999: s. 476–480.
- [8] Gander, W.; G, W.; Golub, G. H.; aj.: Least Squares Fitting of Circles and Ellipses. 1994.
- [9] Guil, N.; Zapata, E. L.: Lower order circle and ellipse hough transform. *J. Pattern Recognition*, ročník 30, 1997: s. 1729–1744.
- [10] Kato, H.; Billinghurst, M.: Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99*, Washington, DC, USA: IEEE Computer Society, 1999, ISBN 0-7695-0359-4, s. 85–94.
URL <http://dl.acm.org/citation.cfm?id=857202.858134>
- [11] Prof. Charles Woodward, e. a.: ALVAR [online].
<http://virtual.vtt.fi/virtual/proj2/multimedia/index.html>, [cit. 2013-01-04].
- [12] Sutherland, I. E.: The Ultimate Display. In *Proceedings of the IFIP Congress*, 1965, s. 506–508.

- [13] Szentandrási, I.; Zachariáš, M.; Havel, J.; aj.: Uniform Marker Fields: Camera Localization By Orientable De Bruijn Tori. In *Proceedings of the 2012 11th IEEE International Symposium on Mixed and Augmented Reality*, Institute of Electrical and Electronics Engineers, 2012, str. 2.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=10122
- [14] Szentandrási, I.; Zachariáš, M.; Havel, J.; aj.: Five Shades of Gray for Fast and Reliable Camera Pose Estimation. 2013.
- [15] Wang, C.; Sklar, D.; Johnson, D.: Forward error-correction coding. *Crosslink*, ročník 3, č. 1, 2001: s. 26–29.
- [16] Ward, J. H.: Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, ročník 58, č. 301, 1963: s. 236–244.
URL <http://www.jstor.org/stable/2282967>
- [17] Zhou, F.; Duh, H. B.-L.; Billingham, M.: Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *ISMAR*, IEEE, 2008, s. 193–202.
URL <http://dx.doi.org/10.1109/ISMAR.2008.4637362>
- [18] Zuliani, M.: RANSAC for Dummies. Technická zpráva, 2008.

Příloha A

Obsah CD

Příložené CD obsahuje v kořenovém adresáři níže popsanou adresářovou strukturu.

testing-data Obsahuje testovací data (obrázky)

chessboard-all Data pro celkové testování vzoru šachovnice

umf-all Data pro celkové testování klasického černobílého UMF

grayscale-umf-all Data pro celkové testování Šedotónového UMF

special-grayscale-umf-all Data pro celkové testování Speciálního Šedotónového UMF

chessboard-angle Data pro testování vlivu úhlu pohledu na detekci šachovnice

grayscale-umf-angle Data pro testování vlivu úhlu pohledu na detekci Šedotónového UMF

special-grayscale-umf-angle Data pro testování vlivu úhlu pohledu na detekci Speciálního Šedotónového UMF

chessboard-distance Data pro testování vlivu počtu viditelných čtverců šachovnice

grayscale-umf-distance Data pro testování počtu viditelných čtverců na detekci Šedotónového UMF

special-grayscale-umf-distance Data pro testování počtu viditelných čtverců na detekci Speciálního Šedotónového UMF

special-grayscale-umf-scene Data pro testování Speciálního Šedotónového UMF ve scéně

results Obsahuje ukázkové výsledné obrázky, pro každý obrázek je pro názornost uloženo všech 15 mezikroků

chessboard-all Výsledky celkového testování vzoru šachovnice

umf-all Výsledky celkového testování klasického černobílého UMF

grayscale-umf-all Výsledky celkového testování Šedotónového UMF

special-grayscale-umf-all Výsledky celkového testování Speciálního Šedotónového UMF

chessboard-angle Výsledky vlivu úhlu pohledu na detekci šachovnice

grayscale-umf-angle Výsledky vlivu úhlu pohledu na detekci Šedotónového UMF

- special-grayscale-umf-angle** Výsledky vlivu úhlu pohledu na detekci Speciálního Šedotónového UMF
- chessboard-distance** Výsledky vlivu počtu viditelných čtverců šachovnice
- grayscale-umf-distance** Výsledky vlivu počtu viditelných čtverců na detekci Šedotónového UMF
- special-grayscale-umf-distance** Výsledky vlivu počtu viditelných čtverců na detekci Speciálního Šedotónového UMF
- special-grayscale-umf-scene** Výsledky testování Speciálního Šedotónového UMF ve scéně
- src** Obsahuje zdrojové kódy programu, včetně souboru README, konfiguračního souboru programu a souborů pro program CMake
- umf-cylinder-detector** Zdrojové kódy detektoru UMF na válci
- original-umf** Zdrojové soubory původního UMF detektoru
- eigen2** Knihovna Eigen2
- bin** Přeložený program a konfigurační soubor. Bohužel překlad na referenčních školních počítačích není možný (viz. **B**). Binární soubory tedy nemusí fungovat. Doporučuji program před použitím přeložit.
- tests** Obsahuje soubor pro tabulkový procesor (formát ODS programu Libre Office) z celkovými výsledky testování
- text** Text diplomové práce ve formátu PDF a zdrojové soubory programu \LaTeX
- presentation** Text prezentace práce ve formátu PDF a zdrojové soubory programu \LaTeX
- program-doc** Programová dokumentace generovaná programem doxygen ve formátu HTML, nezahrnuje zdrojové kódy původního detektor UMF
- poster** Plakát k práci
- video** Prezentační video

Příloha B

Manuál

Tato příloha obsahuje informace o programu, jeho instalaci a spuštění.

B.1 Základní informace o programu

Program je napsán v jazyce C++ podle standartu C++11. Platformní nezávislost překladu zajišťuje program CMake¹, který generuje soubor Makefile. Program využívá knihovny OpenCV² a Eigen2³, které jsou nutné pro překlad programu. Knihovna Eigen2 je přímo součástí instalačního balíku, není nutné jej tedy instalovat externě.

Překlad programu byl testován na Linuxové distribuci Arch⁴. Instalace na jiných systémech než UNIX-like nebyla testována, ale program by měl být bez problémů přenosný mezi různými platformami.

V případě problému se prosím obraťte na autora práce.

Poznámka Překlad na referenčních studentských serverech není možný. Servery eva i merlin používají starou verzi překladače gcc nepodporující standart C++11.

B.2 Potřebný software

- Překladač jazyka C++ podporující formát C++11. Nejlépe aktuální verze překladače gcc⁵. Testována na verzi gcc 4.8.
- Program CMake pro vygenerování souboru Makefile. Testováno na verzi CMake 2.8.
- Knihovna OpenCV. Testováno na verzi OpenCV 2.4.5

¹Cross Platform Make: www.cmake.org

²OpenSource Computer Vision Library: opencv.org

³C++ template library for linear algebra: eigen.tuxfamily.org

⁴Arch Linux distribution: www.archlinux.org

⁵GCC, the GNU Compiler Collection: gcc.gnu.org

B.3 Překlad

Překlad na UNIX-like systémech se provede následující sekvencí příkazů:

```
cd <kořenový adresář projektu>
mkdir build
cd build
cmake ..
make
```

Na systémech Windows a jiných se postup instalace může lišit podle nastavení prostředí. V případě použití MinGW by měla instalace proběhnout stejným způsobem.

B.4 Spuštění

Program se po přeložení výše zmíněným postupem nachází ve složce build.

Syntaxe spuštění programu z aktuální složky přes příkazovou řádku:

```
./umfCylinderDetectorTest [-s] [-w] [-a] [-t (1-9)] ..
-c config_file -d directory -o output_file
```

Jednotlivé parametry mají tento význam:

-s	zapíná zobrazování výsledků detekce v oknech
-w	zapíná zápis výsledných obrazů do souborů
-t (1 – 9)	definuje tloušťku čar při kreslení
-a	zapíná kompletní debug výstupy (zápis i zobrazení obrazů)
-c config_file	definuje cestu ke konfiguračnímu souboru
-d directory	definuje cestu k adresáři s testovacími daty
-o output_file	soubor s výstupními daty – souřadnice nalezených středů polí markeru

Příklad spuštění programu ze složky build:

```
./umfCylinderDetectorTest -s -c ../default_configuration.yml ..
-d ../data/chessboard -o output.txt
```

Poznámka Konkrétní výsledky detekce mohou záviset na aktuálním nastavení parametrů v konfiguračním souboru. Příkladem může být rozdílné nastavení prahu detekce hran pro šedotónové a černobílé UMF.

Příloha C

Konfigurační soubor

konfigurační soubor obsahuje některé hodnoty, kterými je možné ovlivnit chování jednotlivých kroků algoritmu. Konfigurační soubor je ve formátu YAML¹ a všechny níže popsané parametry je nutné v souboru uvést. Vzorový konfigurační soubor s vhodnými hodnotami se nachází v kořenovém adresáři projektu. Následuje popis jednotlivých nastavitelných parametrů.

adaptiveThreshold Práh adaptivní prahování

scanlineStep Vzdálenost skenovacích přímek

bufferSize Velikost bufferu pro adaptivní prahování

searchStep Velikost kroku pro hledání dalšího prahového bodu

searchRadius Rozsah hledání hranového bodu

searchThreshold Práh rozlišující hranu

inliersRatioEllipse Předpoklad poměru inliers vůči všem datům v algoritmu RANSAC při hledání elips

modelDistanceThreshold Práh kritéria vzdálenosti středu elipsy pro algoritmus RANSAC pro elipsy

modelPyramideDistanceThreshold Práh pyramidového kritéria pro algoritmus RANSAC pro elipsy

modelAngleThreshold Práh kritéria úhlu pro algoritmus RANSAC pro elipsy

inliersRatioParabola Předpoklad poměru inliers vůči všem datům v algoritmu RANSAC při hledání parabol

modelDistanceThresholdParameters Práh vzdálenosti od parametrické přímky pro algoritmus RANSAC pro paraboly

modelDistanceThresholdHorizon Práh vzdálenosti průniku na horizontu pro algoritmus RANSAC pro paraboly

¹YAML: www.yaml.org

distanceSupplementThreshold Práh při doplňování parabol zda jsou paraboly shodné

correctnessSupplementThreshold Práh při doplňování parabol výstupní korekce

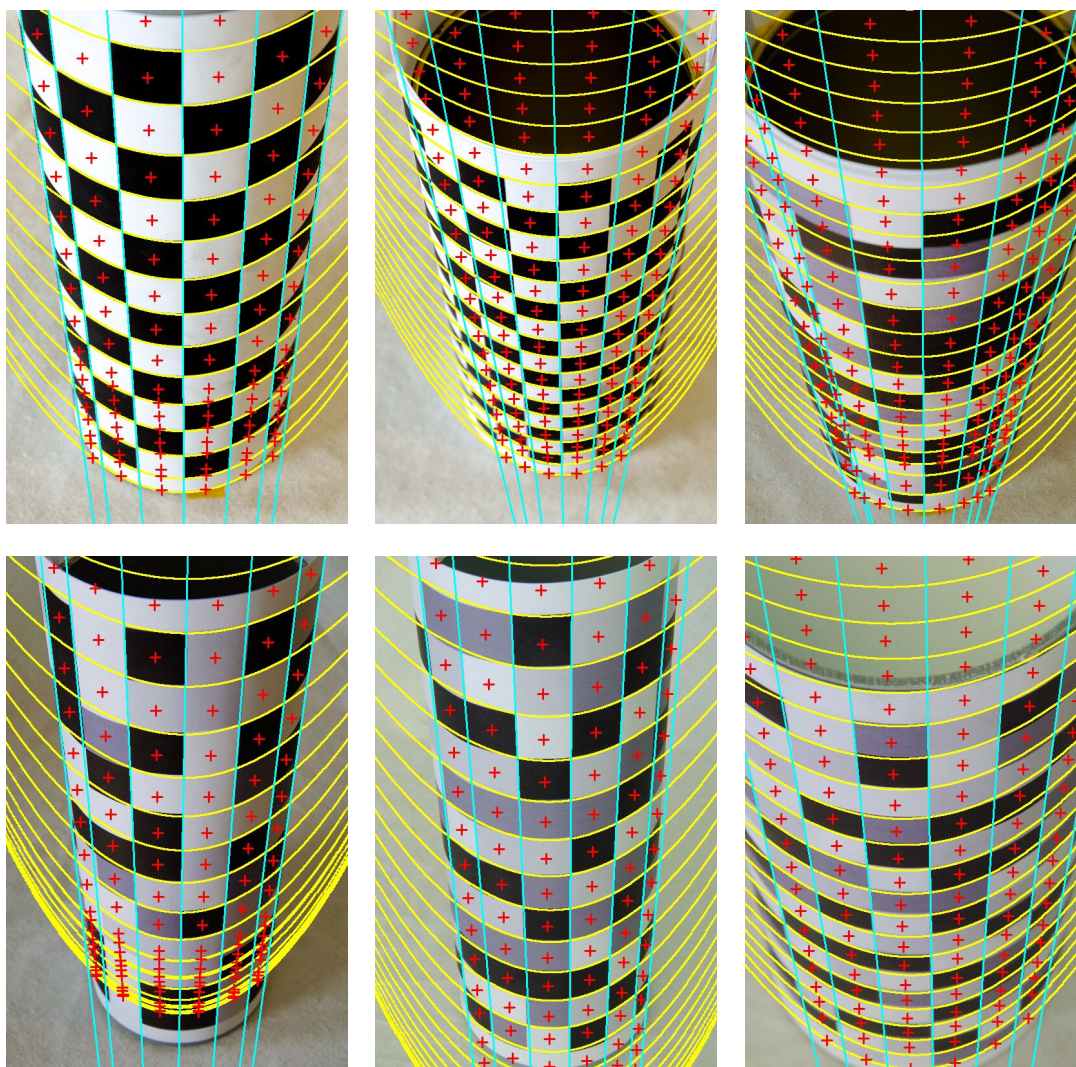
linesDeviationLimit Práh standartní odchylky pro přímky při rozhodování směru elips a
přímek

linesMeanLimit Práh průměrné vzdálenosti pro přímky při rozhodování směru elips a
přímek

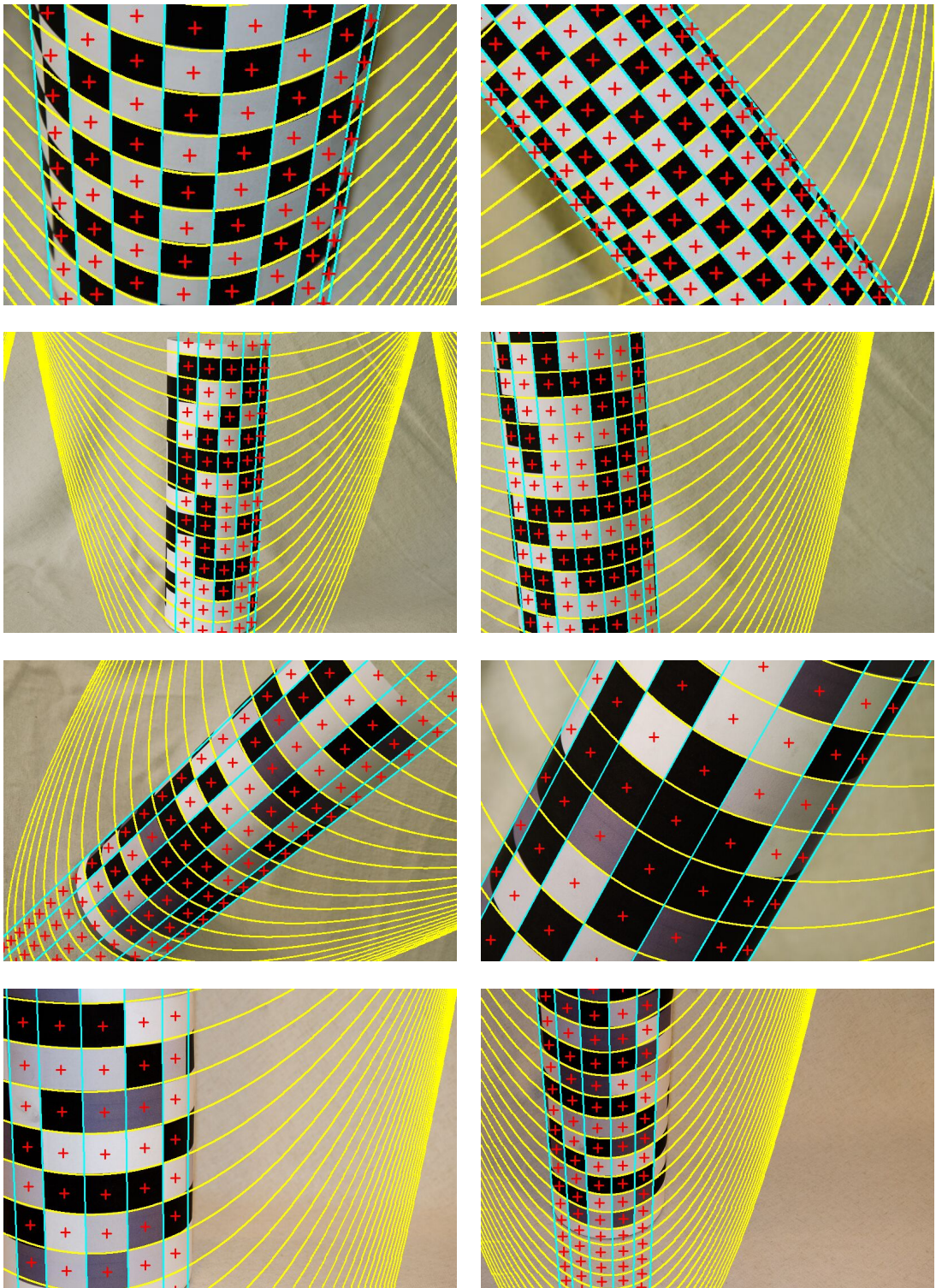
Příloha D

Ukázky testovacích dat

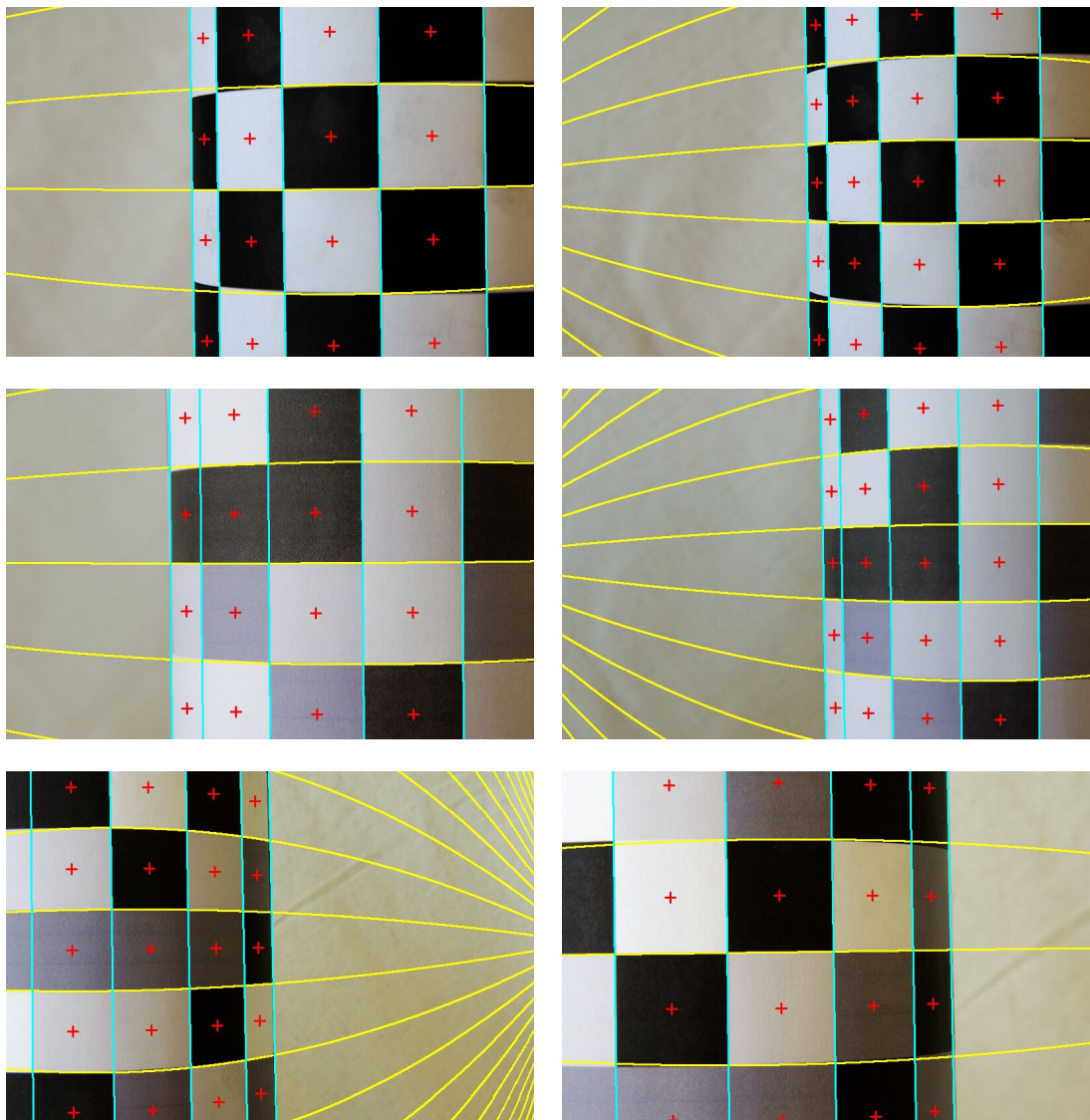
V této příloze najdete ukázky výstupů z testování algoritmu.



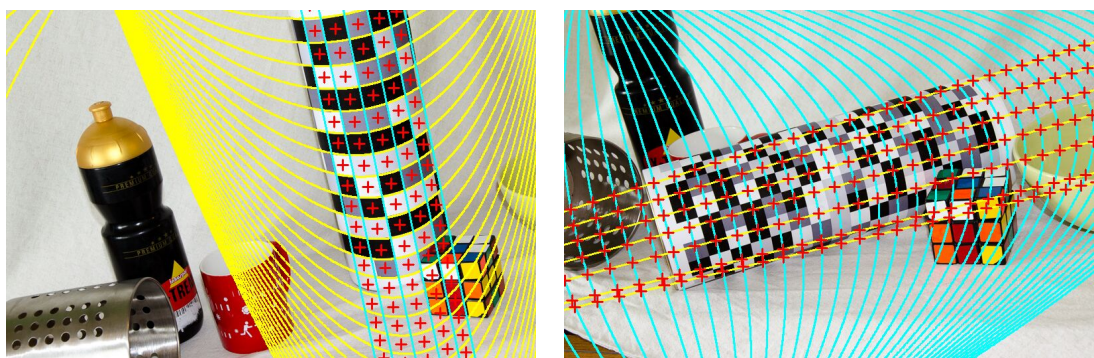
Obrázek D.1: Ukázky výstupů z testování vlivu úhlu pohledu.



Obrázek D.2: Ukázky výstupů z celkového testování.



Obrázek D.3: Ukázky výstupů z testování vlivu počtu markerů.



Obrázek D.4: Ukázky testování Speciálního Šedotónového UMF ve scéně.

Příloha E

Plakát

UNIFORM MARKER FIELD NA VÁLCI

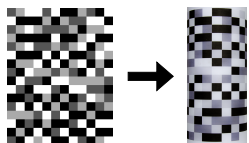


Ústav Počítačové
Grafiky a Multimédií

Autor: Radim Kříž

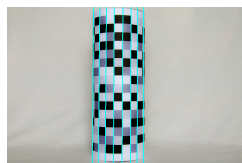
Vedoucí: doc. Adam Herout

Práce představuje metodu umožňující detekovat vzory Uniform Marker Field a jeho šedotónové varianty na plášti válce. Algoritmus vychází s existujícího planárního detektoru.



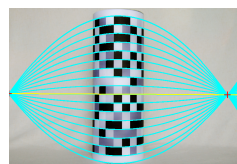
Nejprve jsou detekovány skupiny hranových bodů vytvářené markerem na plášti válce, se kterými se pak dále pracuje.

Jsou rozlišeny dva hlavní směry hran. Ve směru, který je tvořen přímkami, se detekují přímkami na plášti válce.

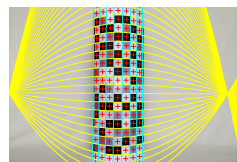
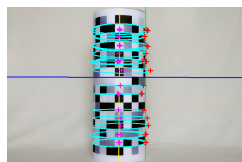


Z ostatních hran tvořených částmi elips se detekují elipsy reprezentující prstence tvořené markerem. Z nalezených parametrů lze zjistit vlastnosti válce.

Kvůli zpřesnění detekce v extrémnějších případech se elipsy nahrazují parabolami. Z parametrů je nalezen horizont a společné průsečíky parabol na něm.



Z nalezených parabol a přímek je definována výsledná mřížka určující středy polí markeru.



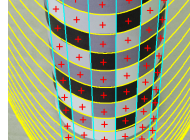
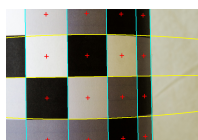
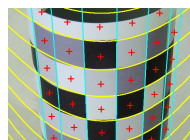
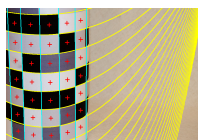
SHRNUTÍ

K realizaci jednotlivých kroků detekčního algoritmu jsou mimo jiné využity metody RANSAC, Hierarchické shlukování a Metoda nejmenších čtverců.

V tabulce jsou výsledky testování pro různé typy markerů. Šachovnicový vzor odpovídá ideálnímu vzoru a Speciální Šedotónové UMF se prokázal jako nejvhodnější reálný vzor pro detekci na válci.

Vzor	Úspěšnost [%]	Přímky [%]	Prstence [%]
Šachovnice	100	83	99
Černobílý UMF	40	55	56
Šedotónový UMF	85	85	89
Spec. šed. UMF	96	92	98

VÝSLEDKY



Radim Kříž, xkriz03@stud.fit.vutbr.cz, Diplomová Práce, 2013

Obrázek E.1: Plakát k projektu.