



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTEM PRO OPTICKÉ MĚŘENÍ

OPTICAL MEASUREMENT SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN OPRAVIL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Jan Opravil

ID: 106685

Ročník: 2

Akademický rok: 2011/2012

NÁZEV TÉMATU:

System pro optické měření

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou bezkontaktní kontroly součástek pomocí kamer. Na základě získaných znalostí nasnímejte pro zvolenou aplikaci učící a testovací databáze snímků v různých modifikacích a s předpokládanými vadami. Navrhněte systém vhodný pro technickou praxi, který by při měření prováděl vyhledání a kontrolu oblastí/vlastností zadaných na vzorových datech. Implementujte kalibrační metody pro stanovení správného vzoru a pro možnost měření v reálných rozměrech. Implementujte metody měření a stanovení míry shody. Dosažené výsledky zhodnoťte.

DOPORUČENÁ LITERATURA:

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

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

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

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

Termín zadání: 6.2.2012

Termín odevzdání: 21.5.2012

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

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt:

Tato diplomová práce se zabývá tvorbou a testováním optického měřicího systému. Jsou zde uvedeny základní části počítačového vidění. Dále rozebrány jsou pak některé způsoby předzpracování obrazu a vyhledávání šablon. Vše je směřováno na konkrétní praktickou úlohu. Vybranými metodami detekce šablon jsou metoda korelace, klasická a hybridní Hausdorffova vzdálenost, Radial Sampling Space a Circular Sampling Space. Tyto metody jsou naprogramovány v jazyce C++ a srovnány s knihovní funkcí pro vyhledávání šablon.

Klíčová slova:

Metody vyhledání šablon, Hausdorffova vzdálenost, Hybridní Hausdorffova vzdálenost, Korelace, Circular Sampling Space, Radial Sampling Space, Optický tok, metody pro vytvoření optického měřicího systému.

Abstract:

This diploma thesis deals with the creation and testing of optical measurement system. There are basic parts of computer vision. Some ways of image preprocessing and templates matching are discussed. Everything is directed to a particular practical task. Selected methods for templates matching are the Correlation Method, the Classical and Hybrid Hausdorff Distance, Radial and Circular Sampling Space. These methods are programmed in C++ and they are compared with function for searching templates from specific library.

Keywords:

Template matching, Hausdorff Distance, Hybrid Hausdorff Distance, Correlation, Circular Sampling Space, Radial Sampling Space, Optical flow, Methods for creating optical measuring system.

Bibliografická citace:

OPRAVIL, J. *Systém pro optické měření*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 68 s. Vedoucí diplomové práce Ing. Miloslav Richter, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Systém pro optické měření jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 21. května 2012

.....

podpis autora

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Miloslavu Richterovi, Ph.D za účinnou metodickou, pedagogickou a odbornou pomoc při zpracování mé diplomové práce. Dále bych rád poděkoval své partnerce, rodině a blízkému okolí za cennou podporu.

V Brně dne: 21. května 2012

.....

podpis autora

Obsah

1. Úvod.....	9
2. Zpracování obrazu.....	11
2.1. Scéna.....	11
2.2. Snímání scény.....	12
2.3. Předzpracování.....	15
2.4. Segmentace.....	16
2.5. Popis objektu.....	16
3. Cíl práce.....	17
4. Předzpracování obrazu.....	18
4.1. Potlačení šumu.....	18
4.2. Prahování.....	18
4.3. Detekce hran.....	19
4.4. Detekce významných bodů.....	20
5. Metody pro vyhledání šablon.....	22
5.1. Vzájemná korelace	22
5.2. Korelace v DFT.....	23
5.3. Metoda H. Y. Kima a S. A. de Araújo	24
5.3.1. Převod obrazu do CiSS a RaSS prostoru.....	24
5.3.2. Analýza obrazu v CiSS prostoru.....	25
5.3.3. Analýza obrazu v RaSS prostoru.....	26
5.4. Hausdorffova vzdálenost.....	27
5.4.1. Výpočet Hausdorffovy vzdálenosti.....	28
5.4.2. Modifikace Hausdorffovy vzdálenosti.....	29
5.5. Hybridní Hausdorffova vzdálenost HHD.....	31
5.5.1. Výpočet vzdálenosti.....	31
5.5.2. Normalizace gradientu.....	32
5.6. Optický tok.....	32
5.6.1. Lucas and Kanade.....	34

5.6.2. Horn and Schunck.....	34
6. Vlastní práce.....	37
6.1. Implementace korelace v DFT.....	37
6.2. Implementace metody H. Y. Kima a S.A. De Araújo.....	38
6.2.1. Třída CiSS.....	38
6.2.2. Třída RaSS	39
6.3. Implementace Hausdorffovy vzdálenosti.....	40
6.3.1. Klasická Hausdorffova vzdálenost CHD.....	41
6.3.2. Modifikace HD.....	42
6.3.3. Hybridní Hausdorffova vzdálenost HHD.....	42
6.4. Optický tok.....	44
6.5. Kalibrace měření.....	46
6.6. Testovací program.....	48
6.7. Generátor snímků, automatické testování.....	49
7. Testování a srovnávání algoritmů.....	51
7.1. Testování na umělých snímcích.....	51
7.2. Testování posunu na reálných datech.....	52
7.3. Další testování.....	57
8. Praktické nasazení.....	61
9. Závěr.....	64
10. Literatura.....	66

1. Úvod

S kamerami se v dnešní době člověk setkává téměř všude. Většina notebooků a telefonů má kameru integrovanou přímo v těle přístroje, kamery pronikly i do domovních zvonků. Jejich instalace na budovách a veřejných prostranstvích je poměrně masová. Lidé si na kamery musí zvyknout, i když jejich přínos je mnohdy přinejmenším diskutabilní.

Zmíněné kamery slouží ve většině případů pouze k monitorování oblasti. V posledním desetiletí však zaznamenávají progresivní růst sofistikovanější systémy. Ty nacházejí uplatnění v dopravě, lékařství a v průmyslu, zkrátka všude tam, kde je zapotřebí automaticky identifikovat, měřit či kontrolovat různé objekty. V tomto případě se již bavíme o počítačovém vidění zahrnujícím celou oblast dané problematiky (snímání, zpracování, reprezentaci, ...). Tato práce se věnuje systému pro optické měření a jeho využití v technické praxi, tedy jen části této disciplíny.

Praktické nasazení optických systémů do průmyslu má několik zásadních výhod. Mezi ně patří například snadná modifikovatelnost. To znamená, že při změně výrobního programu na montážní lince stačí pouze přehrát software kamerového systému a výroba může pokračovat. Takto se ušetří náklady spojené s novou instalací čidel, která byla používána pro kontrolu dříve. Další výhodou je kontrola několika parametrů výrobku současně pomocí jediné kamery. Odpadají tedy náklady na další čidla a složitost výrobního procesu se patřičně sníží.

Kamerové systémy však mají i svá omezení. Při kontrole je potřeba určitým způsobem upravit proces tak, aby kamera mohla pořídit kvalitní snímek, to není vždy jednoduchá úloha. Dalšími požadavky je spolehlivost, rychlost a přesnost. To vše závisí nejen na použitém softwaru, ale také na ostatních komponentách a celkovém uspořádání celého měřicího systému.

Tvorba takového systému je značně složitá. Většina firem, která chce optické systémy používat, má několik možností. Pokud nemá potřebné technické zázemí pro jeho realizaci, může využít služeb specializovaných firem, které se touto problematikou zabývají (Neovision, Resim, Aviko, apod.). Tyto firmy zpravidla dodávají kompletní systém včetně kamer, softwarového vybavení a potřebného zaškolení. Kromě vyšší ceny je nevýhodnou také to, že je měřicí systém dodáván jako tzv. black box. Tedy nevíme, co přesně se v něm skrývá a jak funguje, neznáme tedy ani potenciální zdroje chyb. Výhodou je, že za provoz měřicího zařízení je zodpovědná profesionální firma, která

případné problémy rychle vyřeší. Pokud se firma rozhodne pro vývoj optického systému vlastními silami, pak po koupi HW má několik možností, jak měřící SW vytvořit. První možností je použití specializovaných SW pro dané účely. Tyto SW jsou např. Scorpion, Halcon nebo MontiVision atd. Tyto SW jsou samozřejmě zpoplatněny, jejich rysem je snaha o jednoduchost a rychlost vývoje, často není potřeba ani znalost nízkourovňových programovacích jazyků. Sestavování programu zde probíhá spojováním různých grafických bloků, které představují jednotlivé úkony počítačového vidění. Uživatel okamžitě vidí, jak jednotlivý blok mění obraz a může se tak snadno rozhodnout o jeho nastavení. Další výhodou tohoto SW je, že nevyžaduje podrobnou znalost jednotlivých metod počítačového vidění.

Další možností vývoje měřícího SW je použití knihoven funkcí pro počítačové vidění. Mezi tyto knihovny patří např. Matrox Imaging Library a nebo velmi známá populární knihovna OpenCV. Tyto knihovny jsou řádově levnější než specializovaný SW. Kromě jasné výhody programu na míru, má tento způsob vývoje měřícího SW několik nevýhod, jako jsou nutná znalost počítačového vidění a programování a spotřeba času v procesu vývoje.

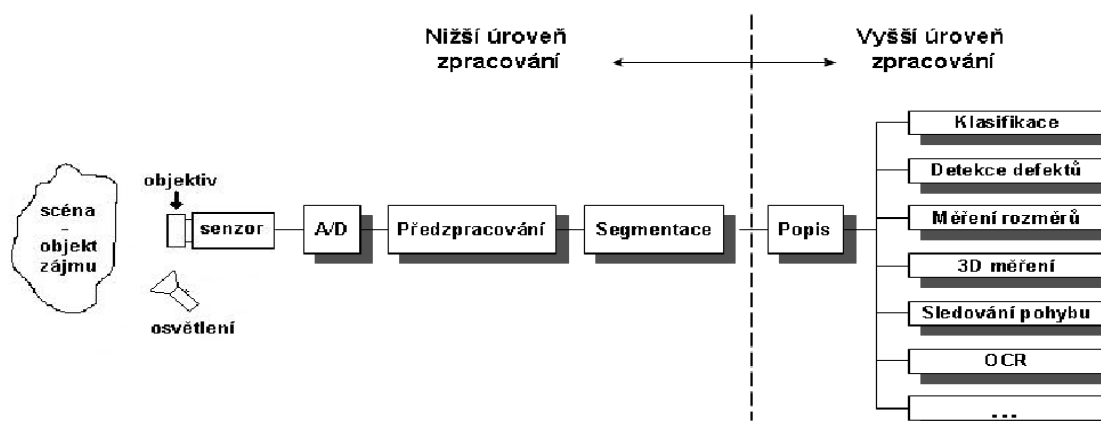
Poslední možností je vytvoření měřícího SW bez pomoci specializovaných nástrojů. Tato možnost je z hlediska vstupních investic nejlevnější, ale je vyžadována dobrá orientace v dané problematice. Další nevýhodou je, že vývoj může trvat velmi dlouhou dobu.

V této práci budou uvedeny základní pojmy a rozdělení počítačového vidění z hlediska praktického použití. Dále představím několik základních technik práce s obrazem a popíšu některé metody, které by mohly být použity jako základní kámen měřícího SW. Představené metody budou podrobně vysvětleny a následně naprogramovány bez využití podpůrných knihoven počítačového vidění. V další části práce budou metody otestovány a na základě těchto testů bude vybrána vhodná metoda pro nasazení do praxe.

2. Zpracování obrazu

Kamerové systémy začínají být v průmyslu velmi oblíbené. Jejich hlavními výhodami jsou flexibilita, rychlost, bezkontaktní měření, měření více parametrů jedním přístrojem. Tomu, aby měřicí kamerový systém měl všechny tyto výhody, musí předcházet kvalitní rozbor úlohy, výběr vhodného hardwaru, správné nasnímání a v neposlední řadě kvalitní vyhodnocení snímku. Splnění těchto podmínek není jednoduché.

Kompletní řetězec optického měřicího systému je na obrázku č. 1. V reálných aplikacích můžeme některé bloky ze schématu vynechat či sloučit v závislosti na konkrétní aplikaci. Běžně se ke zpracování pořízeného obrazu používají průmyslové počítače (IPC), pro rychlé aplikace pak DSP, které však nemohou dosáhnout komplexnosti a snadné modifikovatelnosti jako IPC, proto se používají v kombinaci.



Obr. č. 1: Řetězec zpracování obrazu.

Zdroj [1].

2.1. Scéna

Správné přípravě scény bychom měli věnovat dostatek času. Zde máme totiž největší množství informací a jakýmkoliv dalším zpracováním o ně nenávratně přicházíme. Scénu bychom měli nastavit vždy tak, aby se žádané informace uchovaly.

Důležitou součástí přípravy scény je její osvětlení. Pokud chceme potlačit pozadí, můžeme osvětlit pouze objekt zájmu nebo jeho část, musíme však předem znát jeho

přesnou polohu a rozměr. Většinou takovou informaci nemáme a proto využijeme jiných vlastností objektu (transparentnost, reflexe, luminescence materiálu), popřípadě, pokud máme možnost objekt nebo pozadí nějakým způsobem označit, označíme jej, aby žádané vlastnosti získal.

Světlo je elektromagnetické záření, které, podle, Einsteinovy teorie duality, má vlnový i částicový charakter. Většinou pracujeme s předměty, které jsou ve srovnání s vlnovou délkou světla podstatně větší, proto můžeme použít vlnový model světla, tedy klasickou fyziku (zákon odrazu, lomu, šíření světla, neovlivnění dvou vzájemných světelných paprsků ...).

Volbou vhodného zdroje světla můžeme ovlivnit některé jeho parametry. Nejdůležitější jsou intenzita, vlnová délka, polarizace a směrovost. Zdroj světla může být:

- bodový – zdůrazní objekt, drsnost povrchu, silné odlesky,
- difúzní,
- zadní – obrys,
- tangenciální – zdůrazní poruchy, trhliny na povrchu objektu.

Každý výše zmíněný zdroj má svoje výhody i nevýhody. V reálných aplikacích se tedy můžeme setkat s využitím více druhů osvětlení tak, že snímací kamera pořídí dva snímky s různým osvětlením. Také vhodná volba vlnové délky nám může usnadnit následné zpracování. Vlnovou délku můžeme volit buď širokospektrální nebo úzkospektrální. Úzkospektrálním zdrojem světla je například LED, laser apod. Tento typ osvětlení v ideálním případě vyzařuje monochromatické světlo dané barvy, toho s výhodou využíváme, pokud je objekt stejné nebo kontrastní barvy. To stejné můžeme uplatnit u pozadí. U výběru spektra musíme brát v úvahu objektiv a samotný snímací senzor. Běžně se využívá spektra ve viditelném záření, dále pak v oblasti IR. Ostatní oblasti spektra se využívají pro speciální případy (např. UV v kosmonautice, RTG v lékařství atd.). Každá úloha si žádá individuální nastavení tak, abychom co nejvíce potlačili pozadí a zároveň neztratili informace o objektu, popřípadě ho zvýraznili nad pozadí.

2.2. Snímání scény

Snímání scény je prvním blokem v celém řetězci zpracování (viz obr. č. 1). Spojitá obrazová funkce obsahuje největší počet informací. Ke snímání těchto informací používáme kameru. Drtivá většina kamer má před samotným světlocitlivým snímačem

objektiv, který soustředí světelné paprsky na snímač. Objektiv si v nejjednodušším příkladě můžeme představit jako ideální spojnou čočku. Po průchodu paprsku rovnoběžného s optickou osou čočky se paprsek láme a míří do bodu nazvaného ohnisko čočky. Vzdálenost tohoto bodu od středu čočky je ohnisková vzdálenost f . Ohnisková vzdálenost má přímý vliv na zvětšení (zmenšení) objektu v obrazové rovině. Dalšími důležitými parametry jsou obrazová vzdálenost a' a předmětová vzdálenost a . Vztah mezi těmito veličinami popisuje zjednodušená zobrazovací rovnice č. 1.

$$\frac{1}{a} + \frac{1}{a'} = \frac{1}{f} \quad \text{vzorec 1}$$

Příčné zvětšení čočky (Z) popisuje rovnice č. 2 a udává poměr velikosti objektu v předmětové rovině a v rovině obrazové. Příčné zvětšení je pro nás důležitým parametrem, pokud chceme snímat objekt s určitou přesností. Zvětšení je také úměrné vzdálenosti objektu (s) od čočky. Pokud objekt umístíme od oblasti $s \ll f$, pak výsledný snímek nebude ostrý. To znamená, že pro daný objektiv můžeme dosáhnout jen konečného zvětšení.

$$Z = -\frac{a'}{a} = -\frac{f}{a-f} = \frac{1}{\frac{s}{f}-1} \quad \text{vzorec 2}$$

Ostrost snímku hodnotíme pomocí rozptylového kroužku. Pokud do předmětové oblasti umístíme bod a , v obrazové oblasti se nám zobrazí jako kružnice o určitém poloměru. To znamená, že na tento bod nemáme zaostřeno. Velikost poloměru rozptylového kroužku, který budeme považovat ještě za ostrý, zvolíme s přihlédnutím k ostatním chybám snímače (vzorkování snímače atd.). [2]

Pro získání ostrého snímku musíme vzít v úvahu také hloubku ostrosti DoF (Depth of Field). DoF je vzdálenost, která určuje rozmezí mezi nejbližším a nejvzdálenějším bodem, který bude přijatelně ostrý i v obrazové rovině. DoF můžeme ovlivnit pomocí následujících parametrů:

- clonovým číslem (můžeme si představit, že čím menší clona, tím menší počet paprsků vlétne na snímač, tím menší kroužek vykreslí),
- ohnisková vzdálenost (čím je menší, tím je menší zvětšení a i velikost poloměru kroužku),
- vzdálenost objektu (čím dál je objekt, tím je menší zvětšení).

Pokud chceme snímat objekt s určitým detailem, musíme vypočítat zvětšení a tedy i vzdálenost objektu od objektivu. Poté na tuto vzdálenost zaostřit a snažit se o co největší DoF, s přihlédnutím k ostatním parametrům (jas).

U reálných čoček se projevují některé vady, oproti uvažované (ideální čočce). Zde jsou některé z nich:

- Astigmatism,
- Coma,
- Curvature of Field,
- Longitudinal Chromatic Aberration,
- Lateral Chromatic Aberration,
- Spherical Aberration,
- Distortion,
- Flare,
- Ghost Images,
- Diffraction.

Tyto vady se snaží výrobci objektivů potlačit. Proto jejich objektivy obsahují více druhů čoček, z nichž se každá snaží korigovat některou z vad. Dále jsou podle aplikace k objektivu přidávány různé filtry, které například propouštějí jen některé vlnové délky, nebo jen polarizované světlo. Pokud filtry vhodně použijeme, můžeme si usnadnit další zpracování obrazu. Musíme však vzít v úvahu, že při průchodu světla filtrem se sníží jeho intenzita (typickým příkladem je polarizační filtr, který může snížit intenzitu světla až o polovinu, někdy je tedy jeho použití spíše na škodu).

Světlo, které prošlo objektivem, dopadá na světlocitlivý čip, který ho snímá a digitalizuje. V současné době jsou na trhu dvě hlavní technologie snímání světla a to CCD čip a CMOS. Obě technologie mají své výhody a své zastánce. CCD čipy přeměňují foton na pár elektron-díra a tento náboj po dobu expozice integrují, proto mají nižší šum. V okolí světlocitlivých buněk nemá tento senzor žádnou elektroniku, vyčítání signálu se provádí postupným posouváním náboje z jednotlivých buněk.

CMOS senzory mají u každé buňky, která představuje fotoodpor PN nebo PIN přechod, jednoduchý obvod, který přímo měří signál na buňce. Jednotlivé pixely můžeme vyčítat v libovolném pořadí, cenou je však vyšší šum tohoto senzoru.

U obou druhů snímačů probíhá digitalizace. Tu můžeme rozdělit do dvou kroků. Prvním krokem je vzorkování. Vzorkovací frekvence odpovídá rozlišení senzoru. Druhým krokem je kvantizace, která probíhá v A/D převodníku.

Základní parametry senzorů jsou:

- velikost pixelu,
- citlivost,
- dynamický rozsah,
- shuttering – schopnost spuštění a vypnutí expozice v libovolný čas,
- rychlost,
- antiblooming – schopnost lokálního přexponování, které nemá vliv na zbytek obrazu,
- spolehlivost.

Zdroj: [3].

Čipy pro snímání barevného obrazu jsou tvořené barevnými filtry, které propouštějí na světlocitlivou buňku pouze danou barvu. V mnoha aplikacích se spokojíme s šedotónovou kamerou, která bude mít nižší datový tok a lepší citlivost (žádné barevné filtry). Také se zbavíme nepříjemné vlastnosti, kdy jeden pixel na obrázku je tvořen několika barevnými pixely na senzoru, které jsou od sebe posunuty, a tedy pixel na snímku odpovídá průměru jednotlivých barev kolem skutečného bodu. Tento jev eliminuje použití nových technologií výroby, které využívají toho, že světlo o různé vlnové délce proniká do různé hloubky křemíkových vrstev. Barevné světlocitlivé prvky jsou pak umístěny pod sebou.

2.3. Předzpracování

Jedná se o nejnižší úroveň zpracování obrazu, jehož cílem je zdůraznit či potlačit určité charakteristiky obrazu a připravit ho pro další zpracování. Dále pak redukovat množství dat. Je zřejmé, že pro každou úlohu zpracování obrazu je nutné použít odlišný způsob předzpracování.

Předzpracování se nejčastěji používá na:

- potlačení šumu;
- transformaci jasové úrovně;
- ostření a detekci hran;
- morfologii;
- transformaci barev;
- odstranění zkreslení;
- zvýšení kontrastu;

- obnovení obrazu při známé degradaci.

Pro většinu výše zmíněných úloh existují základní postupy, jak předzpracování provést. V některých případech se může tento blok zpracování obrazu přeskočit, obzvláště když jsme kvalitně nasnímali scénu.

2.4. Segmentace

Segmentace obrazu je proces, kterým rozdělíme obraz do oddělených množin. Každá množina odpovídá objektu nebo oblasti s určitými vlastnostmi. Pokud hovoříme o kompletní segmentaci, máme na mysli skupinu segmentovaných objektů, které mají přesnou shodu s požadovaným objektem. Častěji však segmentovaný objekt nesouhlasí, pak jde o částečnou segmentaci. K úspěšné částečné segmentaci potřebujeme předem znát některé údaje o objektech nebo pozadí (může se jednat o barvu, tvar, umístění, ...).

Hlavní problémy segmentace jsou nejednoznačnost obrazových dat, šum a nerovnoměrné osvětlení.

Základní rozdělení segmentačních metod je následující:

- segmentace prahováním;
- segmentace na základě detekce hran;
- segmentace založená na regionech.

Výsledek segmentace se liší nejen použitou metodou, ale také nastavením různých parametrů. Často se také kombinuje několik segmentačních metod tak, abychom se více přiblížili úplné segmentaci.

2.5. Popis objektu

Abychom mohli rozpoznat, co segmentované objekty ve scéně představují nebo je měřit, přiřadíme jim charakterizující parametry, kterým říkáme vektor příznaků. Každý vektor příznaků různých objektů by měl být odlišný tak, abychom podle něj mohli později klasifikovat objekty. Příznaky mohou být například velikost, obvod, průměrná barva, výstřednost atd. Dále můžeme zkombinovat některé příznaky a tím vytvořit nový. Nejdůležitější krok je správný výběr příznaků tak, aby byl výpočetně nenáročný, měl časovou stálost a dobrou klasifikační schopnost. Podle typu úlohy pak můžeme příznaky použít na vyhodnocení různých objektů.

3. Cíl práce

V práci se budu zabývat vytvořením algoritmů pro systém optického měření s návazností na jeho praktické využití. Pro jeho realizaci bude v práci uvedeno několik možných postupů založených na vyhledání šablon v obraze (kapitola č. 5.). Některé tyto metody vyžadují určité základní předzpracování obrazu, to je uvedeno v předchozí kapitole č. 4. . V kapitole číslo 6. budou uvedené metody pro vyhledání šablon implementovány v programovacím jazyce C++.

Pro jednodušší testování uvedených metod bude vytvořen program, jehož pomocí bude uživatel schopný nastavit jednotlivé parametry vyhledávání, předzpracování obrazu a také zobrazit výsledky hledání včetně časové náročnosti. S využitím tohoto programu budou následně v kapitole č. 7. všechny metody otestovány na reálných snímcích tak, aby výsledky testu vypovídaly o možnosti nasazení konkrétní metody do praxe. V poslední kapitole bude aplikována vybraná metoda na konkrétní praktické úloze.

Představené metody budou testovány na šedotónových obrazech. Při nutnosti zpracovávat barevný obraz, ho můžeme vhodnou transformací převést na obraz šedotónový, popřípadě rozšířit zmíněné metody příslušným způsobem tak, aby dokázaly zpracovat i barevný obraz.

4. Předzpracování obrazu

V následující kapitole bude uvedeno několik metod předzpracování obrazu. Uvedené metody předzpracování nejsou samozřejmě jedinou možností, existuje celá řada alternativních řešení stejného problému. V práci se však omezíme na ty nejpoužívanější.

4.1. Potlačení šumu

Každý reálně pořízený snímek obsahuje šum. Šum má náhodný charakter a nelze ho tedy úplně odstranit. Můžeme ho však do jisté míry potlačit. To provádíme na základě lokálních oblastí. Většinou použijeme konvoluci s určitou maskou (průměr, gauss), medián nebo rotující masku. Každá z metod má své klady a zápory. V této práci použijeme konvoluční masku, vyjadřující průměr (obrázek č. 2). Maximální velikost masky je dána nejmenším detailem v obraze, který chceme zachovat nerozmazaný. Další informace o potlačení šumu můžeme najít v [4].

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{A} \quad \text{B}$$

Obr. č. 2: Masky pro průměrování.

4.2. Prahování

Tento typ předzpracování se používá, když má pozadí obrazu na celé své ploše rozdílný jas než objekt. Pokud tuto podmínku splníme, segmentuje tato metoda pozadí od objektu. Její hlavní výhodou je výpočetní nenáročnost vyplývající pouze z porovnávání jasové úrovně s daným prahem T . Pokud je jas pixelu nižší než práh T , potom změním jas pixelu na 0, v opačném případě na 1. Výstupní obraz je tedy černobílý. Hodnotu prahu T můžeme nastavovat experimentálně nebo různými technikami (minimum histogramu apod.). Existují samozřejmě i sofistikovanější metody prahování, v naší práci nám však postačí tato.

Prahování můžeme modifikovat tak, že pokud je hodnota jasu nižší než zadaný prah T , pak je nová hodnota jasu rovna 0, jinak je hodnota jasu nezměněná (vzorec č. 3). V tomto případě je výstupní obraz opět šedotónový. Výhoda tohoto způsobu je v zachování jasové úrovně větší než práh T (a tedy i zachování většího množství informace).

$$f(i, j) = \begin{cases} 0 & \text{pro } f(i, j) \leq T \\ f(i, j) & \text{jinak} \end{cases} \quad \text{vzorec 3}$$

4.3. Detekce hran

Hrany v obraze vnímáme tam, kde dochází k výrazné změně jasu. Tedy tam, kde gradient jasové funkce dosahuje maximálních hodnot. Hranové operátory tedy pracují na principu vyhledávání lokálních minim, respektive maxim, první derivace obrazové funkce (Sobelův a Robertsův operátor), nebo na principu průchodu nulou druhé derivace (Laplaceův operátor) obrazové funkce. K těmto výpočtům můžeme použít konvoluční masku uvedenou ve vzorci č. 4.

$$\begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \text{ - Sobelův operátor.} \quad \text{vzorec 4}$$

$$\begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix} \text{ - Laplaceův operátor.}$$

Vidíme, že Sobelův operátor je závislý na směru gradientu. Kompletní masku pro všechny směry bychom získali rotací uvedené masky o 45° . Každý z operátorů reaguje na hrany rozdílně. Například Laplaceův operátor více reaguje na šum, než operátory aproximující první derivaci. Výběr vhodného gradientního operátoru závisí na charakteru vstupního obrazu a na konkrétní úloze. Tak jako u metod potlačení šumu, existuje mnohem více hranových detektorů. Za zmínku stojí Cannyho hranový detektor, který se obecně považuje za optimální nástroj. Ten v sobě kombinuje několik dalších metod předzpracování obrazu (protože hranové detektory jsou citlivé na vyšší frekvence, které představují hrany, ale také šum) tak, aby výsledné detekované hrany

korespondovaly s opravdovými hranami v obraze. Další informace o hranových detektorech můžeme nalézt v [5].

4.4. Detekce významných bodů

Pokud mluvíme o významných bodech, máme na mysli takové body, ve kterých dochází k velké jasové změně ve všech směrech, tedy v místě setkání hran, které představují hranice, či rohy objektů. Opět máme několik možností výpočtu např. Moravec, Forsten, Susan, Harriss. Z článku [6], který se zabývá funkcí a srovnáním různých detektorů významných bodů, jsem pro tuto práci vybral Harrisův operátor, který má dobré detekční hodnocení a přitom není na implementaci složitý.

V algoritmu se používají některé již uvedené techniky (potlačení šumu – ze stejného důvodu jako hranové detektory, prahování, vyhledání gradientu, vyhledání lokálních maxim – z důvodu omezení významných bodů v určité oblasti). Algoritmus však využívá i výpočtu vlastních čísel z matice M (vzorec č. 5). Pomocí vlastních čísel lze rozhodnout o jaký typ bodu se jedná, pokud jsou obě čísla malá, jedná se patrně o bod, který se příliš neliší od okolí. Pokud je hodnota jednoho vlastního čísla vysoká a druhého malá, jedná se patrně o hranu objektu. Pokud jsou hodnoty obou vlastních čísel vysoké, jedná se o rohový bod, tedy o námi hledaný bod.

Výpočet vlastních čísel matice je výpočetně náročná operace, proto se aproximuje pomocí vzorce č. 6. Celá funkce algoritmu se skládá z následujících bodů:

1. Pro každý pixel (x,y) ve vstupním obraze spočítej autokorelační matici M podle vzorce č. 5.

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$A = \left(\frac{\partial f(x,y)}{\partial x} \right)^2 \oplus w, B = \left(\frac{\partial f(x,y)}{\partial y} \right)^2 \oplus w$$

vzorec 5

\oplus konvoluce, w Gaussovo okno

2. Pro každý pixel vypočítej $C(x,y)$ podle vzorce č. 6, což představuje detekované body.

$$C(x, y) = \det(M) - k(\text{trace}(M))^2$$

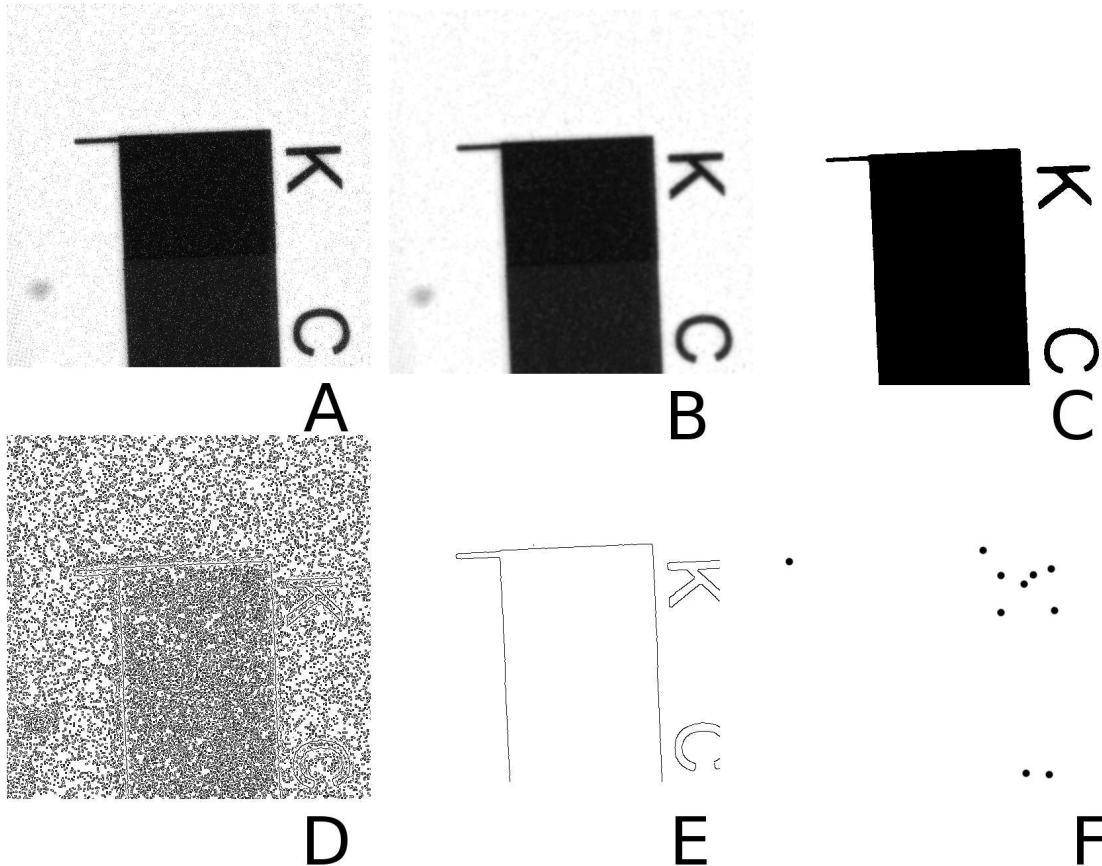
$$\det(M) = \lambda_1 \lambda_2 - AB - C^2$$

$$\text{trace}(M) = \lambda_1 + \lambda_2 = A + B$$

$k = \text{konstant}$

vzorec 6

3. Vyprahuj mapu $C(x,y)$ prahem T .
4. Vyhledej lokální maxima v $C(x,y)$ a označ je jako významné body.



Obr. č. 3: Předzpracování obrazu: A - původní snímek, B - potlačení šumu, C - prahování, D - hranování, E - hranování s potlačením šumu, F - významné body.

5. Metody pro vyhledání šablon

Typ měřených vlastností objektu určuje měřicí software, tedy použité metody. V mnoha případech je potřeba přesné určení polohy daného objektu v obraze. Tímto se bude zabývat následující kapitola. Představené metody jsou podrobně popsány tak, aby je bylo možné implementovat bez podpůrných knihoven. Některé z uvedených metod nejsou součástí knihoven počítačového vidění a nejsou tedy standardně používány pro vyhledávání šablon. Autoři těchto metod však poukazují na jejich dobré vlastnosti, bohužel ve většině případů bez uvedení parametrů potřebných ke zhodnocení. Je tedy na místě se přesvědčit, zda tyto, zatím neprosazené, metody mají potenciál stát se součástí měřicích systémů.

První metodou je metoda korelace. Tato metoda patří k velmi známým a používaným metodám. Ostatní zde uvedené metody nejsou tak běžné a proto je tato práce bude srovnávat vzhledem k metodě korelace.

5.1. Vzájemná korelace

Pro vzájemné popsání dvou signálů $f(t)$ a $g(t)$ můžeme použít korelační funkci (vzorec č. 7). Tento vzorec je velmi podobný vzorci pro konvoluci, s drobnou úpravou lze vzorec kombinovat. Maximum funkce $c(t)$ určuje posunutí signálu $g(t)$ oproti $f(t)$ do místa, kde jsou si signály nejvíce podobné.

$$c(t) = \int_{-\infty}^{\infty} f^*(\Gamma) * g(t + \Gamma) d\Gamma \quad \text{vzorec 7}$$

Počítačové vidění zpracovává dvourozměrné diskrétní signály, které představují obraz, proto můžeme vzorec č. 7 rozšířit o jednu dimenzi a diskretizovat. Korelaci dvou diskrétních konečných signálů $m(x,y)$ a $n(x,y)$ můžeme popsat vzorcem č. 8, ve kterém můžeme meze sumy nahradit konstantami (protože součin nulových koeficientů obrazu není třeba sčítat) M a N , které představují rozměry obrazu.

$$r[x, y] = \sum_{j=-0}^{N-1} \sum_{i=-0}^{M-1} m[i, j] * n[x+i, y+j] \quad \text{vzorec 8}$$

Zjištění korelace podle výše zmíněného vzorce je výpočetně náročné, proto se téměř nepoužívá. Pokud potřebujeme provést korelaci dvou signálů, provedeme to pomocí korelace ve frekvenční oblasti (kapitola 5.2.).

5.2. Korelace v DFT

Korelace v prostorové doméně je výkonově náročná, proto se signál převádí do frekvenční domény (v počítačovém vidění mluvíme o prostorové frekvenci) pomocí Fourierovy transformace (FT, ve vzorcích označena jako \mathfrak{F}). Zde využijeme vlastnosti posunutí vzoru v čase (time shifting). Tato vlastnost říká, že pokud posuneme signál $f(t)$, který má obraz ve FT $F(w)$ v časové doméně o r , pro jeho obraz v FT bude platit vzorec č. 9. Posunutím funkce $f(t)$ v čase se tedy změní fáze v DFT. Posunutou funkci můžeme označit $g(t)$ s obrazem $G(w)$. Pokud by byla funkce $f(t)$ dvourozměrná a diskrétní, vztah se náležitě rozšíří.

$$G(w) = \mathfrak{F} \{ f(t \pm r) \} = e^{\mp jwr} F(w) \quad \text{vzorec 9}$$

Pro korelaci $K(u, v)$ v DFT dvou stejných časově posunutých diskrétních dvourozměrných signálů, které představují snímek, můžeme odvodit následující vztah (vzorec č. 10).

$$K(u, v) = \frac{G^*(u, v) F(u, v)}{|G^*(u, v) F(u, v)|} = \frac{F^*(u, v) e^{-jw\left(\frac{u\Delta x}{M} + \frac{v\Delta y}{N}\right)} F(u, v)}{\left| F^*(u, v) e^{-jw\left(\frac{u\Delta x}{M} + \frac{v\Delta y}{N}\right)} F(u, v) \right|} = \quad \text{vzorec 10}$$

$$\frac{F^*(u, v) e^{-jw\left(\frac{u\Delta x}{M} + \frac{v\Delta y}{N}\right)} F(u, v)}{|F^*(u, v) F(u, v)|} = e^{-jw\left(\frac{u\Delta x}{M} + \frac{v\Delta y}{N}\right)}$$

V rovnici č. 10 jsme označili posuny v jednotlivých osách jako Δ . M, N je velikost obrazu a $*$ označuje komplexně sdružené číslo. Komplexně sdružené číslo jsme použili díky vztahu konvoluce a korelace, vzorec č. 11.

$$f * g = f^*(-t) \times g$$

$$\mathfrak{F}\{f * g\} = (\mathfrak{F}\{f\})^* \cdot \mathfrak{F}\{g\}$$
vzorec 11

Korelace dvou funkcí $f(x)$ a $g(x)$ přechází na násobení jejich obrazů ve frekvenční oblasti tak, jak je naznačeno ve vzorci č. 12. Nevýhodou tohoto postupu je nutnost převedení signálů do frekvenční oblasti a zpět. Pro minimalizaci této nevýhody můžeme pro převod použít rychlé algoritmy vyvinuté Cooleyem a Tukeyem v roce 1965 (označované jako Discrete Fast Fourier Transform DFFT). Tyto algoritmy najdeme v projektu *FFTW* na adrese www.fftw.org, kde je i jejich popis.

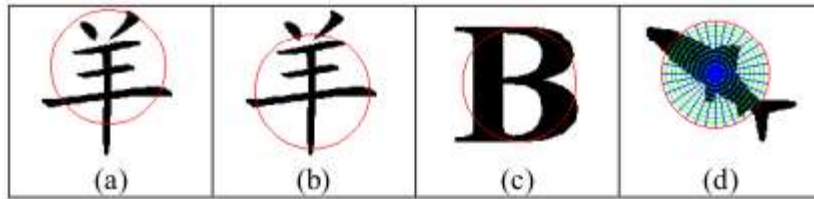
$$k(x, y) = \mathfrak{F}^{-1} \left[\frac{\mathfrak{F}(f(x, y)) \cdot \mathfrak{F}(g(x, y))}{|\mathfrak{F}(f(x, y)) \cdot \mathfrak{F}(g(x, y))|} \right]$$
vzorec 12

5.3. Metoda H. Y. Kima a S. A. de Araújo

Metodu k rozpoznání šablony invariantní vůči rotaci a posunutí představili v roce 2007 Hea Yong Kim a Sidnei Alves de Araújo z Polytechnické univerzity San Paulo v Brazílii ve svém článku [7]. V této práci se zaměříme pouze na metodu, která bude zpracovávat černobílý obraz, tím se sníží výpočetní nároky. Představená metoda pracuje s šablonou Q a s obrazem I , ve kterém se šablona hledá. Metoda je založena na kruhovém vzorkování tzv. *CiSS* (Circular Sampling Space) a radiálním vzorkování *RaSS* (Radial Sampling Space) vstupního obrazu. Použitím této metody obohatíme vstupní 2D obraz o další rozměr. Vzniklý 3D prostor vytvoří projekci k analýze obrazu I .

5.3.1. Převod obrazu do CiSS a RaSS prostoru

Pro jednoduchost budeme předpokládat: $H(I) = \{0, 1\}$, $H(Q) = \{0, 1\}$, kde H je obor hodnot dané funkce. Šablona Q je definována v doméně D jako funkce $Q: D \rightarrow \{0, 1\}$. Doména D představuje kružnici o zvoleném poloměru. Šablona může mít různý tvar, může být různě velká a může obsahovat „díry“ tak, jak je ukázáno na obr. č. 4 (a-c). Šablonu představuje pouze část obrazu uvnitř kruhové domény D . Poloměr i střed D může být změněn libovolně. Tato šablona může být umístěna kdekoliv v obraze I .



Obr.č. 4: Ukázka šablony(a-b), d-zeleně CiSS modře RaSS.

Zdroj: [7].

Pro analyzovaný 2D obraz $I:R^2 \Rightarrow R$ je funkce zobrazení CiSS $C_I:R^2 \times R^+ \Rightarrow R$ definována podle vzorce č. 13. Ve vzorci si $C_I(x,y,r)$ můžeme představit jako průměr všech jasových hodnot v obrazu I ve vzdálenosti r od středu (x,y) .

$$C_I(x, y, r) = \int_0^{2\pi} I(x + r \cos \theta, y + r \sin \theta) d\theta \quad \text{vzorec 13}$$

Podobně pro převedení obrazu I do RaSS s délkou l , je funkce zobrazení $R_I^l:R^2 \times R^+ \Rightarrow R$ definována podle vzorce č. 14. Opět si můžeme představit, že výraz $R_I^l(x,y,\alpha)$ ve vzorci představuje průměrnou hodnotu pixelů v obraze I , které leží na radiální úsečce o velikosti l a úhlu α od středu (x,y) .

$$R_I^l(x, y, \alpha) = \int_0^l A(x + t \cos \alpha, y + t \sin \alpha) dt \quad \text{vzorec 14}$$

V obou případech můžeme s výhodou použít jakékoliv algoritmy pro vykreslení kružnice či úsečky (např. Bresenhamův algoritmus (úsečka), Midpoint algoritmus (kružnice), blíže popsáno v [8]).

5.3.2. Analýza obrazu v CiSS prostoru

Nejprve vypočteme $C_I(x,y,r)$ pro všechny pixely $(x,y) \in I$ podle rovnice č. 13 a pro zvolené poloměry kružnic $r_i = \{r_0, r_1, \dots, r_{N-1}\}$. Výsledkem bude 3D pole, ve kterém bude každému bodu (x,y) přiřazen vektor K_I o N prvcích. Dále vypočítáme pro šablonu podobným způsobem $C_Q(x',y',r)$, nyní však pouze pro jediný bod (x',y') , který představuje střed šablony. Opět můžeme zvolit poloměry kružnic r_Q (je vhodné volit $r_Q = r_i$) a dostáváme jediný vektor K_Q . Tento vektor představuje sadu příznaků šablony, které jsou invariantní vůči otočení. Nyní vypočítáme podle vzorce č. 15, sumu absolutní

odchylky mezi vektory K_Q a K_I pro všechny body $C_Q(x,y)$. Pokud je v bodě (x,y) diference $CD_{I,Q}(x,y)$ menší než námi určený práh \mathcal{Z} , potom v tomto bodu leží střed hledané šablony. Zmíněný postup však může i přes vhodně zvolený práh obsahovat chyby detekce šablony v místech, kde se nevyskytuje. Abychom tuto chybu co nejvíce eliminovali, zvolíme větší počet kružnic.

$$CD_{I,Q}(x,y) = \frac{1}{N} \sum_{n=0}^{N-1} |C_I(x,y,r_n) - C_Q(x',y',r_n)| \quad \text{vzorec 15}$$

5.3.3. Analýza obrazu v RaSS prostoru

Příznaky vypočítané pomocí RaSS (vzorec č. 14) nejsou přirozeně invariantní vůči rotaci tak, jak tomu bylo v CiSS. Abychom docílili nezávislosti na rotaci, musíme provést další výpočty. Nejprve však převedeme jak obraz I , tak šablonu Q podobným způsobem popsaným v kapitole 5.3.2. , s tím rozdílem, že použijeme vzorec č. 14. Musíme si tedy zvolit délku l úseček. Tu většinou volíme stejnou jako poloměr domény D . Dále pak úhel $\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_{M-1}\}$. Vypočítáme tedy $R^l_I(x,y,\alpha)$ a $R^l_Q(x',y',\alpha)$. Diferenci určíme pomocí vzorce č. 16, ve kterém *cshift_j* znamená kruhový posun argumentu vektoru šablony. To zajistí invarianci vůči otočení. Tímto postupem dosáhneme přesnějšího zaměření šablony než u CiSS, nevýhodou je vyšší časová náročnost.

$$RD_{I,Q}(x,y) = \frac{1}{M} \underset{j=0}{\text{MIN}}^{M-1} \left[\sum_{m=0}^{M-1} |R^l_I(x,y,\alpha_m) - \text{cshift}_j[R^l_Q(x',y',\alpha_m)]| \right] \quad \text{vzorec 16}$$

Výše byly popsány dvě metody, jejich výhody a nevýhody. Aby se dosáhlo co nejvyšší účinnosti algoritmů je podle [7] vhodné použít obě techniky ve dvou fázích. V první fázi se využije prostoru CiSS, který je rychlý, avšak často chybně označí oblast jako šablonu. Nestává se však, že by mylně neoznačil místo se šablonou. V druhé fázi budeme pracovat pouze s omezenou množinou z první fáze. V prostoru RaSS tedy zkoumáme pouze některé oblasti vstupního obrazu, proto se vyšší časová náročnost eliminuje. Tímto postupem získáme rychlou a přesnou metodu, která je poměrně silná vůči šumu.

Tuto metodu lze použít i na šedotónový obraz a lze ji rozšířit o invarianci vůči zvětšení, postup je popsán v [7].

5.4. Hausdorffova vzdálenost

Hausdorffova (německý matematik, 1968) vzdálenost (HD), někdy označovaná jako Pompeius-Hausdorffova vzdálenost, je matematický popis vzdálenosti dvou konečných podmnožin A, B z množiny D . V našem případě budou podmnožiny představovat body obrazu. Množina $A = \{a_0, a_1, \dots, a_{N_A-1}\}$ je množina bodů z obecného obrazu I a množina $B = \{b_0, b_1, \dots, b_{N_B-1}\}$ je množina bodů z obrazu šablony Q . Hausdorffova vzdálenost je nelineární operátor, je orientovaná a neplatí tedy komutativnost. Výpočet Hausdorffovy vzdálenosti A od B označíme $h(A, B)$ a vypočítáme podle vzorce č. 17. V tomto vzorci označuje $\| \cdot \|$ vzdálenost, pro její výpočet můžeme použít libovolnou metriku (např. Eukleidovskou, Manhattan, Chessbord (vzorec č. 18), ...). Pro výpočet nejkratší vzdálenosti, ve vzorci označena jako d_B , můžeme využít Voronoiův diagram (obrázek č. 5). Pomocí tohoto diagramu okamžitě najdeme nejbližší bod.

$$h(A, B) = \max_{a \in A} d_B(a) \quad \text{vzorec 17}$$

$$d_B = \min_{b \in B} \|a - b\|$$

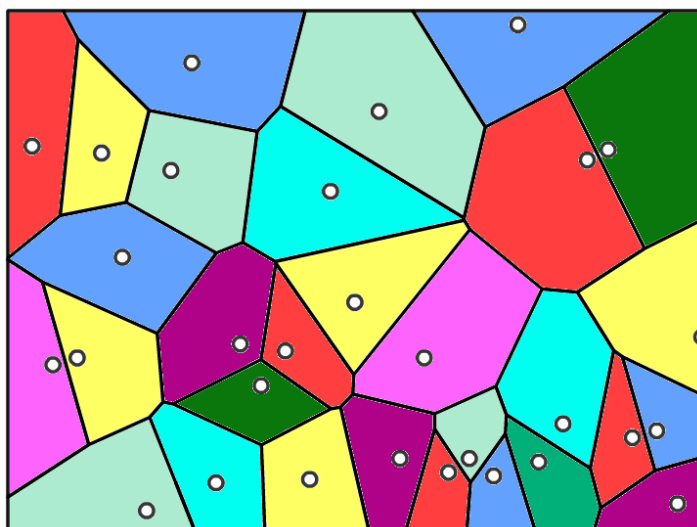
$$\|a - b\|_{chess} = \max(|a_x - b_x|, |a_y - b_y|)$$

$$\|a - b\|_{city} = |a_x - b_x| + |a_y - b_y| \quad \text{vzorec 18}$$

$$\|a - b\|_{euclid} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

Hausdorffova vzdálenost $h(A, B)$ vyjadřuje maximální hodnotu z nejmenších vzdáleností množiny A od množiny B . Pro tuto vzdálenost, jak již bylo řečeno, neplatí komutativnost argumentů: $h(A, B) \neq h(B, A)$, pro oboustrannou Hausdorffovu vzdálenost $H(A, B)$ můžeme napsat vzorec č. 19.

$$H(A, B) = \max(h(A, B), h(B, A)) = \max\left(\max_{a \in A} \min_{b \in B} \|a - b\|, \max_{b \in B} \min_{a \in A} \|b - a\|\right) \quad \text{vzorec 19}$$



Obr. č. 5: Voronoiův diagram náhodně vygenerovaných bodů.

Zdroj:[14].

5.4.1. Výpočet Hausdorffovy vzdálenosti

V základním tvaru, který je uveden výše, nemůžeme Hausdorffovu vzdálenost použít pro hledání šablony v obraze. Proto ji musíme vhodně rozšířit tak, jak je uvedeno ve vzorci č. 20.

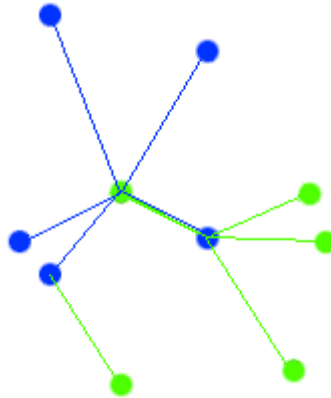
$$M_T(A, B) = \min_t H(A, B \oplus t) \quad \text{vzorec 20}$$

$$B \oplus t = \{b + t | b \in B\}$$

Ve výše zmíněném vzorci symbol \oplus označuje standardní Minkowského sumu a t je strukturální element. Pokud t obsahuje pouze jeden prvek $t = \{t_0\}$ lze dokázat, že Minkowského součet je posunutím dané množiny. Výsledek tohoto předpokladu lze vidět ve vzorci č. 20. Je patrné, že stačí posun pouze jedné z množin A, B . Obrázek č. 6 nám názorně ukazuje množiny A, B včetně vyznačených odpovídajících minimálních vzdáleností $h(A, B)$ a $h(B, A)$. Pokud na tomto obrázku srovnáme výsledné vzdálenosti H a M_T , bude jistě platit: $H(A, B) \gg M_T(A, B)$. To proto, že existuje takové posunutí t , které posune množinu B tím způsobem, že se všechny dílčí vzdálenosti zkrátí. Vzorec č. 20 můžeme pomocí vzorců č. 17 a 19 upravit. Výsledný vzorec bude vzorec č. 21.

$$\begin{aligned}
M_T(A, B) &= \min_t H(A, B \oplus t) = \\
&= \min_t \max \left(\max_{a \in A} \min_{b \in B} \|a - (b+t)\|, \max_{b \in B} \min_{a \in A} \|(b+t) - a\| \right) \\
M_T(A, B) &= \min_t \max(d_B(a-t), d_A(b+t))
\end{aligned}$$

vzorec 21



Obr. č. 6: Hausdorffova vzdálenost množin A (zelená) a B (modrá),
 $h(A, B)$ modře, $h(B, A)$ zeleně.

5.4.2. Modifikace Hausdorffovy vzdálenosti

Při klasickém výpočtu (vzorec č. 17) Hausdorffovy vzdálenosti (HD) je výsledek značně závislý na šumu. Vlivem šumu v jedné z množin A, B může být hodnota HD výrazně odlišná. Je tedy vhodné použít nějakou modifikaci díky které by se vliv šumu snížil. V následujícím textu je uvedeno několik základních modifikací HD. Tyto modifikace byly popsány v [9] a v [10].

1. Modifikace podle Huttenlocher et al.

Místo vyhledání maximální hodnoty d_A vybereme pouze K^{th} prvek z této množiny, kterou musíme nejprve seřadit. Označme N_a jako počet prvků v množině A . q bude počet prvků v množině B . K^{th} prvek však musí být z intervalu $1 \leq K^{th} \leq N_a$. K^{th} prvek určíme podle vztahu $K^{th} = \lfloor f * q \rfloor$, kde platí $0 \leq f \leq 1$. Jak uvádí [10], nejlepších výsledků se dosahuje při volbě $f=0.6$. Pokud bychom zvolili $f=0.5$, pak se jedná o výběr mediánu. Pokud bude platit $K^{th} = N_a$, pak vzorec č. 22 přejde na klasickou HD a bude roven vzorci č. 17. Pro určení vzdálenosti $h_K(B, A)$ platí výpočet obdobně.

$$h_K(A, B) = K_{a \in A}^{th} d_B(a) \quad \text{vzorec 22}$$

2. Modifikace podle Dubuisson a Jain

Tato modifikace je založena na průměrné vzdálenosti. Modifikovaná vzdálenost h_{MHD} se vypočítá podle vzorce č. 23. Výhodou této modifikace je, že není potřeba volby žádného parametru, bohužel ve srovnání s h_K nedosahuje tak dobrých výsledků. To je způsobeno právě podstatou tohoto výpočtu a to průměrováním všech vzdáleností.

$$h_{MHD}(A, B) = \frac{1}{N_A} \sum_{a \in A} d_B(a) \quad \text{vzorec 23}$$

3. Metoda M-estimation

Vzdálenost vypočítaná touto metodou nepředstavuje vzdálenost použité normy d_B . Výsledek je nahrazen průměrem váhové funkce $p(x)$. Výpočet vzdálenosti h_M podle této metody je uveden ve vzorci č. 24.

$$h_M(A, B) = \frac{1}{N_A} \sum_{a \in A} p(d_B(a)) \quad \text{vzorec 24}$$

$$p(x) = \begin{cases} |x|, & |x| \leq \tau \\ \tau, & |x| > \tau \end{cases}$$

Pro výpočet funkce $p(x)$ byla zvolena podle experimentů uvedených v [10], výše zmíněná funkce, ve které τ je prahovací konstanta eliminující vzdálenosti představující velkou chybu. Pro výpočet h_M tedy potřebujeme vhodně určit jeden parametr τ . Pokud bychom zvolili práh $\tau = \infty$, potom by tato metoda byla ekvivalentní metodě pro výpočet h_{MHD} . Prahovou hodnotu je vhodné zvolit v rozmezí $3 < \tau < 5$.

4. Metoda LTS

Tato metoda je definována jako lineární kombinace předchozích statistik. Výpočet h_{LTS} je určen vzorcem č. 25. Ve vzorci H představuje $H = h * N_A$ a $d_B(x)_{(i)}$ i-tou vzdálenost seřazené množiny vzdáleností d_B . Pro výpočet této vzdálenosti potřebujeme opět určit jeden parametr a to h . Tento parametr volíme z intervalu $0 \leq h \leq 1$. Jeho velikost je částečně určena mírou degradace obrazu. Pokud $h = 1$, potom bude výpočet ekvivalentní výpočtu h_{MHD} . Tato metoda dává podle [10] dobré výsledky i když je obraz degradován.

$$h_{LTS}(A, B) = \frac{1}{H} \sum_{i=1}^H d_B(A)_{(i)}$$

vzorec 25

5.5. Hybridní Hausdorffova vzdálenost

HHD

Metodu představili autoři Chyuan-Huei, Thomas Yang, Shang-Hong Lai a Long-Wen Chang v roce 2006, jako robustní vyhledání šablony v obraze [11].

5.5.1. Výpočet vzdálenosti

Tato metoda vychází z klasické Hausdorffovy vzdálenosti. Hlavní rozdíl je ve výpočtu vzdáleností dvou bodů. U klasické HD jsme pro výpočet vzdálenosti (označení $\| \cdot \|$ ve vzorci č. 17) dvou bodů použili některou z metrik (vzorec č. 18). Výsledná vzdálenost závisela pouze na vzájemných polohách těchto bodů. U Hybridní Hausdorffovy vzdálenosti HHD se výpočet vzdálenosti $\| \cdot \|$ mění tak, že již nezáleží pouze na vzájemných polohách bodů, ale také na rozdílnosti jejich jasové složky. Čím více se budou body lišit v jasové složce, tím větší bude jejich výsledná vzdálenost. Pomocí zavedených vah w_1 a w_2 můžeme volit, jakou mírou bude výsledná vzdálenost ovlivněna jasem a samotnou polohou bodů. Nový výpočet vzdáleností je popsán vzorcem č. 26 (změny oproti klasickému výpočtu jsou vyznačeny zeleně), ve vzorci označuje I obraz šablony a F referenční obraz.

$$\begin{aligned} \|a, b, I, F\|_{Hchess} &= \max(w_1|a_x - b_x|, w_1|a_y - b_y|, w_2|I_{(a)} - F_{(b)}|) \\ \|a, b, I, F\|_{Hcity} &= w_1(|a_x - b_x| + |a_y - b_y|) + w_2(|I_{(a)} - F_{(b)}|) \\ \|a, b, I, F\|_{Heucleid} &= \sqrt{w_1((a_x - b_x)^2 + (a_y - b_y)^2) + w_2(I_{(a)} - F_{(b)})^2} \end{aligned}$$

vzorec 26

Použití nového výpočtu vzdáleností můžeme samozřejmě využít i v kombinaci s modifikacemi HD uvedenými v kapitole č. 5.4. . Tím dostáváme značný počet kombinací (výběr modifikace, metriky, výpočtu vzdálenosti).

5.5.2. Normalizace gradientu

Uvedený výpočet vzdálenosti je závislý i na jasových složkách bodů, proto bychom měli zajistit minimalizaci vlivu nerovnoměrného osvětlení, které by mohlo způsobovat velkou chybu ve výpočtu. K potlačení nerovnoměrného osvětlení využívá HHD normalizaci gradientu vzorec č. 27.

$$E(a, b, I, F) = \left| \frac{\nabla I(a)}{\max_{(m,n) \in w_a} |\nabla I(m,n)| + c} * \frac{\nabla F(b)}{\max_{(m,n) \in w_b} |\nabla F(m,n)| + c} \right| \quad \text{vzorec 27}$$

Ve vzorci označuje $\nabla I(a)$ a $\nabla F(a)$ gradient obrazu I a F v bodech a a b . Jmenovatel v rovnici obsahuje maximální hodnotu gradientu v lokálním okně W , dále pak malou kladnou konstantu c , která zajišťuje, aby nedošlo k dělení nulou. Celý zlomek představuje normalizovaný gradient. Znak $*$ označuje ve vzorci skalární součin.

Kombinace normalizace gradientu a výpočtu vzdálenosti způsobí opětovnou úpravu vzorce, která je naznačená ve vzorci č. 28 (zeleně jsou vyznačeny změny oproti vzorci č. 27).

$$\begin{aligned} \|a, b, I, F\|_{HNchess} &= \max(w_1 |a_x - b_x|, w_1 |a_y - b_y|, w_2 |1 - E_{(a,b,I,F)}|) \\ \|a, b, I, F\|_{HNcity} &= w_1 (|a_x - b_x| + |a_y - b_y|) + w_2 |1 - E_{(a,b,I,F)}| \\ \|a, b, I, F\|_{HN euclid} &= \sqrt{w_1 ((a_x - b_x)^2 + (a_y - b_y)^2) + w_2 (|1 - E_{(a,b,I,F)}|)^2} \end{aligned} \quad \text{vzorec 28}$$

5.6. Optický tok

Metody založené na optickém toku se využívají zejména při analýze sekvence snímků, či videa. Optický tok vyjadřuje lokální pohyb objektů v obraze. Touto technikou můžeme zkoumat pohyb ve dvou nebo ve třech osách. V této práci se zaměříme pouze na pohyb v osách kolmých k pohledu kamery. Pro výpočet optického toku využijeme jednoduché metody založenou na diferencích.

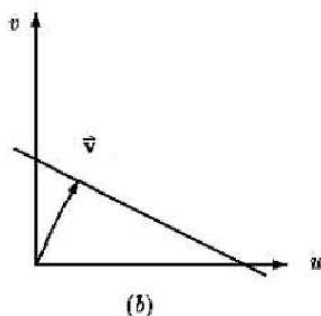
Pro zjednodušení budeme předpokládat, že kamera a pozadí objektu jsou v čase neměnné. Jediný pohyb na scéně koná zkoumaný objekt. Díky tomuto předpokladu můžeme pohyb objektu vyjádřit změnou jeho pozice v čase pomocí vzorce č. 29.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad \text{vzorec 29}$$

Výpočet optického toku provádíme na základě diference jasové a časové složky. Pokud tyto diference budou malé, můžeme vzorec č. 29 rozvinout pomocí Taylorovy řady a dostaneme vzorec č. 30, ve kterém ζ značí vyšší řády derivace, které můžeme pro náš výpočet zanedbat. Postupnými úpravami tohoto vzorce získáme finální podobu vhodnou pro výpočet optického toku \vec{v} .

$$\begin{aligned}
 I(x, y, t) &= I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + \zeta \\
 \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t &= 0 \\
 \frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} &= 0 \\
 \frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} &= 0 \\
 I_x v_x + I_y v_y + I_t &= 0 \\
 \nabla I \cdot \vec{v} + I_t &= 0
 \end{aligned} \quad \text{vzorec 30}$$

Výše uvedená rovnice se nazývá Gradientní omezení. Je to skalární rovnice, která obsahuje dvě neznámé. Pomocí ní můžeme určit pouze normálovou komponentu rychlosti v_n . Tento problém se v [12] popisuje pomocí obrázku č. 7. Kdekoliv na vyznačené přímce mohou ležet správné hodnoty rychlosti v_x a v_y . Na obrázku je také vyznačena normálová komponenta rychlosti představující minimální amplitudu rychlosti.



Obr. č. 7: Problém výpočtu správné rychlosti optického toku.

Hledaná rychlost leží teoreticky kdekoli na uvedené přímce, avšak pouze některé z bodů přímky mají v reálném prostředí smysl (objekt se během pohybu nedeformuje apod.). Známými algoritmy pro výpočet optického toku jsou Lucas and Kanade a Horn and Schunck.

5.6.1. Lucas and Kanade

Tento algoritmus je založen na lokální minimalizaci kvadrátu odchylek gradientu. Pro každý pixel zavádíme chybový člen, který byl odvozen podle vzorce č. 30. Celý chybový člen vidíme ve vzorci č. 31. V něm je zaveden váhovací člen W , který většinou volíme jako dvourozměrnou Gaussovou funkci. Tento člen ovlivňuje váhu daného pixelu v okolí Ω . Čím dále jsme od aktuálního pixelu, tím méně nám bude ovlivňovat celkovou chybu.

$$E(x, y) = \sum_{x, y \in \Omega} W^2(x, y) [\nabla I(x, y, t) \vec{v} + I_t(x, y, t)]^2 \quad \text{vzorec 31}$$

Pro vyhledání minima chybové rovnice položíme její derivaci podle jednotlivých složek rychlosti (\vec{v}) rovnu nule. Po drobných úpravách dostaneme vzorec č. 32, kde jsou rovnice zapsány maticově. Z této rovnice již můžeme vypočítat žádaný optický tok.

$$\hat{A} \vec{v} = \hat{b}$$

$$\begin{pmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_x I_y & \sum W^2 I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{pmatrix} \quad \text{vzorec 32}$$

5.6.2. Horn and Schunck

Tato metoda vychází z minimalizace chybové funkce na globální úrovni tak, jak ukazuje vzorec č. 33. Tato chybová funkce se skládá ze dvou dílčích chybových členů. První z nich je požadavek na gradientní omezení, který byl popsán pomocí vzorce č. 30. Druhý chybový člen vychází z požadavku na homogenní změnu optického toku v okolí. Pokud je změna rychlosti optického toku u, v plynulá, pak jsou druhé parciální derivace

rovny nule. Váhový koeficient λ určuje požadavek na homogenalitu optického toku. Ve vzorci \bar{u}, \bar{v} označuje vážený průměr okolí pixelu.

$$E = (\nabla I \vec{v} + I_t)^2 + \lambda^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$$

$$E = (\nabla I \vec{v} + I_t)^2 + \lambda^2 ((\nabla^2 u)^2 + (\nabla^2 v)^2)$$

kde: $\nabla^2 u = (\bar{u} - u)$; $\nabla^2 v = (\bar{v} - v)$

vzorec 33

Pro získání minima chybové funkce položíme její derivaci podle jednotlivých složek rychlosti u, v , rovnu nule. Tím získáme dvě rovnice o dvou neznámých, které můžeme maticově zapsat pomocí vzorce č. 34.

$$\begin{pmatrix} \lambda^2 + I_x^2 & I_x I_y \\ I_x I_y & \lambda^2 + I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \lambda^2 \bar{u} - I_x I_t \\ \lambda^2 \bar{v} - I_y I_t \end{pmatrix}$$

vzorec 34

Tento vzorec však počítá optický tok pro konkrétní pixel v závislosti na jeho okolí. Pro všechny pixely (x_0, \dots, x_n) obrazu dostaneme soustavu lineárních rovnic. V maticovém zápisu jsou naznačeny ve vzorci č. 35.

$$A \begin{pmatrix} u_0 \\ v_0 \\ \vdots \\ u_n \\ v_n \end{pmatrix} = \lambda^2 \begin{pmatrix} \bar{u}_0 \\ \bar{v}_0 \\ \vdots \\ \bar{u}_n \\ \bar{v}_n \end{pmatrix} + \begin{pmatrix} -I_x(x_0) I_t(x_0) \\ -I_y(x_0) I_t(x_0) \\ \vdots \\ -I_x(x_n) I_t(x_n) \\ -I_y(x_n) I_t(x_n) \end{pmatrix}$$

$$A = \begin{pmatrix} \lambda^2 + I_x^2(x_0) & I_x(x_0) I_y(x_0) & & & & \\ I_x(x_0) I_y(x_0) & \lambda^2 + I_y^2(x_0) & & & & \\ & & \ddots & & & \\ & & & \lambda^2 + I_x^2(x_n) & I_x(x_n) I_y(x_n) & \\ & & & I_x(x_n) I_y(x_n) & \lambda^2 + I_y^2(x_n) & \end{pmatrix}$$

vzorec 35

Řešení výše uvedené soustavy rovnice běžným způsobem by bylo velmi náročné. Za určitých podmínek (uvedeny v [13]) můžeme použít iterační Jacobiho metodu. Tím dostaneme konečný tvar pro výpočet rychlostí, vzorec č. 36. V každé iteraci se pro každý pixel vypočtou nové rychlosti u, v optického toku.

$$u^k = u^{-(k-1)} - \frac{I_x (I_x u^{-(k-1)} + I_y v^{-(k-1)} + I_t)}{\lambda^2 + I_x^2 + I_y^2}$$

$$v^k = v^{-(k-1)} - \frac{I_y (I_x u^{-(k-1)} + I_y v^{-(k-1)} + I_t)}{\lambda^2 + I_x^2 + I_y^2}$$

vzorec 36

6. Vlastní práce

Postupy uvedené v metodice (kapitola 5.) jsou implementovány v jazyce C++. Pro každou metodu byla vytvořena třída. Aby bylo dosaženo jednoty v reprezentaci obrazových dat a manipulaci s nimi, byla vytvořena šablona třídy, která funguje jako vstup i výstup ostatních tříd. Tato podpůrná třída již využívá v některých svých funkcích různé knihovny (například pro načtení obrazu ze souboru se využívá knihovna OpenCV). Takovéto rozdělení má tu výhodu, že pokud budeme chtít využít jinou knihovnu nebo vyčítat data přímo z bufferu kamery, stačí přepsat danou funkci a tím jednoduše odstranit závislost na použité knihovně. Dále byly vytvořeny další statické funkce, které se starají o některé metody předzpracování obrazu (prahování, vyhledání hran, normalizace gradientu, vyhledání významných bodů apod.). Ze všech těchto vytvořených tříd byla vytvořena statická knihovna.

Všechny uvedené algoritmy pracují s šedotónovým obrazem. Pokud bychom potřebovali zpracovat barevný obraz, můžeme dané algoritmy příslušným způsobem upravit nebo barevný obraz převést na šedotónový (např. výpočtem průměrného jasu).

6.1. Implementace korelace v DFT

První implementovanou metodou je metoda korelace, která je prováděna ve frekvenční oblasti kvůli rychlejšímu zpracování. K převodu snímků do frekvenční oblasti byla použita knihovna *FFTW2*, psaná v jazyce C. Z implementované metody byla vytvořena třída v jazyce C++.

Tato metoda není invariantní vůči rotaci. Pokud bychom však tuto vlastnost vyžadovali, můžeme algoritmus upravit následujícím způsobem. Z referenčního snímku vytvoříme nový referenční snímek, tvořený spojením několika rotovaných referenčních snímků (počet rotací závisí na požadované přesnosti). V tomto novém referenčním snímku budeme hledat šablonu. Z pozice maxima ve snímku vzniklém výpočtem korelace šablony a referenčního snímku získáme údaj o rotaci i translaci šablony. Tento postup byl v pyramidovém provedení implementován do programu Matlab.

6.2. Implementace metody H. Y. Kima a S.A. De Araújo

Implementace této metody je rozdělena do dvou tříd. První z nich reprezentuje radiální vzorkování (RaSS) a druhá kruhové vzorkování (CiSS).

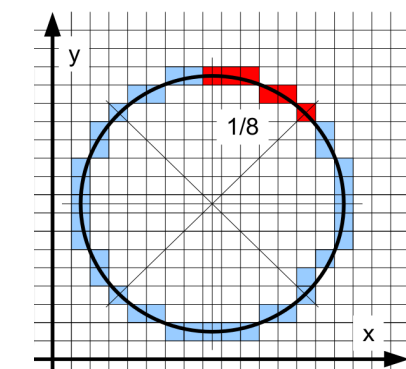
6.2.1. Třída CiSS

Základem metody je rozšíření obrazu o další rozměr. Vznikne tedy 3D prostor, ve kterém je každému bodu vstupního obrazu přiřazen vektor. Ve vlastní třídě je tento prostor reprezentován dvourozměrným polem soukromých tříd *CI*. Třída *CI* obsahuje vektor o prvcích získaných rovnicí č. 13. Třída má také přetíženy operátory tak, abychom mohli jednoduše provádět potřebné výpočty.

Další důležitou funkcí třídy CiSS je výpočet integrálu (rovnice č. 13) bodů na dané kružnici. V našem případě vzorec přejde do diskrétního tvaru, tedy sumy. Právě výpočet bodů ležících na kružnici je časově kritickým úsekem. Rovnici kružnice popisuje vzorec č. 37, kde body x a y jsou body ležící na kružnici se středem s a poloměrem r .

$$(x - s_x)^2 + (y - s_y)^2 = r^2 \quad \text{vzorec 37}$$

Pro urychlení výpočtu rasterizovaných bodů na kružnici využijeme její symetričnost (obrázek č. 8). Kružnice je 8x symetrická, proto nám stačí výpočet pouze 1/8 bodů ležících na kružnici. Ostatní body získáme záměnou souřadnic.



Obr. č. 8: Rasterizovaná kružnice.

Zdroj: [8].

Použitím vzorce č. 37 počítáme aritmetiku s plovoucí desetinnou čárkou a využíváme goniometrické funkce, proto není algoritmus nijak efektivní. Pro zvýšení efektivity můžeme použít algoritmus Midpoint, který používá pouze celočíselnou aritmetiku a pouze operace sčítání a porovnání. To činí algoritmus velmi efektivním. Algoritmus je založen na vykreslování kružnice se středem v počátku $s=[0,0]$. Princip spočívá ve vykreslování pixelů od bodu $[0,R]$ se vzrůstajícím indexem x (v kladném směru osy x) tak dlouho, dokud není $x=y$. Po vykreslení každého bodu spočítáme prediktor P , jeho znaménko indikuje posunutí souřadnice y o jeden pixel v záporném směru osy y . Detaily k algoritmu lze najít v [8].

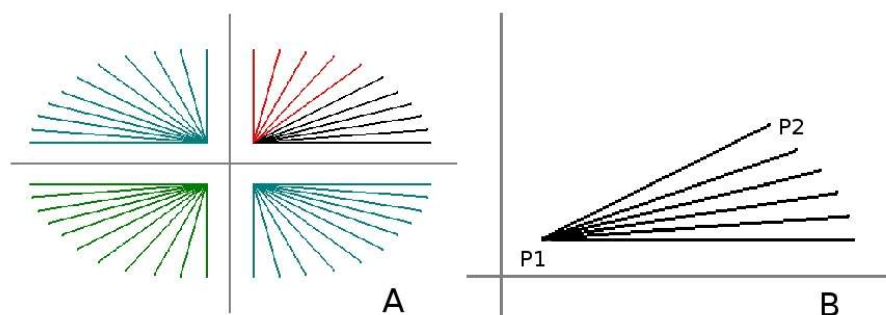
Změnu parametrů metody (počet kružnic, jejich hustota atd.) můžeme ovlivnit pouze u vytváření třídy. Je to z toho důvodu, že při volání konstruktoru třídy se přepočítává vstupní referenční obraz do prostoru CISS. Počet kružnic a jejich hustotu volíme tak, aby největší kružnice měla průměr velikosti šablony. Zvyšováním počtu kružnic zpřesňujeme, ale také prodlužujeme výpočet.

Po vytvoření třídy můžeme zavolat metodu *Find*, která provede vyhledání šablony v referenčním obraze. Nejprve převede vstupní obraz do prostoru CiSS (referenční obraz převádět nemusí, ten je již převeden) a následně provede analýzu obou obrazů. U převodu šablony do prostoru CiSS nám stačí vypočítat vektor CI pouze pro jeden bod šablony (střed). Výsledky jsou uloženy ve vektoru bodů.

6.2.2. Třída RaSS

Metoda analýzy pomocí RaSS je podobná jako v případě CiSS analýzy, tedy i implementace třídy bude podobná. Soukromá třída, reprezentující vektor jednoho bodu, je pojmenována *RI*. Opět i třída *RI* obsahuje přetížené operátory pro výpočty. Za zmínku zde stojí nutnost doplnění diference o posun vektorů tak, jak je uvedeno ve vzorci č. 16. Posun je proveden rotací vektoru.

Ve třídě CiSS byl důležitý rychlý výpočet bodů na kružnici. Zde je důležitý rychlý výpočet bodů na úsečce, která je definovaná dvěma koncovými body $[P1, P2]$. Libovolnou obecnou úsečku (obrázek č. 9A) můžeme získat pomocí úsečky ležící v prvním kvadrantu prohozením souřadnic, os apod..



Obr. č. 9: Úsečka A-obecná úsečka, B-úsečka v prvním kvadrantu.

Pokud bychom počítali body na úsečce pomocí vzorce úsečky, museli bychom použít aritmetiku s plovoucí čárkou a násobení. Abychom se tomuto problému vyhnuli, je v práci použit Bresenhamův algoritmus, který používá pouze celočíselnou aritmetiku a operace sčítání a porovnání. Algoritmus vykresluje úsečku od počátečního bodu $P1$ do bodu $P2$ tak, že postupně zvyšuje index x . V každém kroku se dále přepočítá prediktor a na základě jeho znaménka se rozhodne o posunutí souřadnice y v kladném směru osy y .

Samotná třída RaSS má podobnou strukturu jako třída CiSS. Funkce *Find* je u třídy přetížena tak, abychom mohli omezit prohledávání obrazu pouze na zkoumané oblasti a tím snížit rychlostní nevýhodu této metody tak, jak bylo popsáno v teoretickém rozboru. Postup omezení prohledávání obrazu budeme označovat RC. U této třídy si můžeme zvolit délku úseček a jejich počet. Je vhodné volit délku úsečky ve velikosti poloviny šablony. Jejich počet pak udává, po jaké hodnotě bude měněna jejich směrnice. Počet a velikost přímek ovlivňuje přesnost a čas výpočtu. Výsledky jsou uloženy ve vektoru bodů.

6.3. Implementace Hausdorffovy vzdálenosti

Výpočet jednotlivých $H(A,B)$ je poměrně časově náročný. Proto za předpokladu, že pro posunutí t se bude $M_T(A,B)$ blížit k nule, můžeme použít jednosměrnou vzdálenost $h(A,B)$. Ve všech realizovaných třídách je výpočet HD prováděn pomocí $h(A,B)$, ve kterém A je šablona a B je obraz.

6.3.1. Klasická Hausdorffova vzdálenost CHD

Metoda je založena na vyhledání nejmenší $h(A,B)$ ze všech pozic šablon tak, jak ukazuje vzorec č. 21. K samotnému výpočtu $h(A,B)$ můžeme použít modifikovaný Voroniův diagram. Ten bude představovat předpočítanou mapu minimálních vzdáleností V . Mapu V si můžeme představit jako šedotónový obraz, ve kterém jasová složka představuje minimální vzdálenost k nejbližšímu bodu. Výpočet $h(A,B)$ (vzorec č. 17) se nám značně zrychlí, protože nemusíme hledat minimální vzdálenost d_B . Tu zjistíme pomocí mapy V . Velkou výhodou je to, že předpočítaná mapa se nemění v rámci posouvání šablony, stačí nám tedy pouze jednou vypočítat mapu vzdáleností referenčního snímku B . Další výhodou je, že výpočet mapy není závislý na množině A a lze tedy vytvořit mapu i v době, kdy ještě A neznáme.

Výpočet vzdálenostní mapy

Vytvoříme si obrázek, který bude představovat mapu V o rozměrech stejných, jako vstupní obraz. Všechny pixely mapy nastavíme na maximální hodnotu vzdálenosti (např. 255).

Samotný výpočet minimálních vzdáleností můžeme provádět v následujících bodech:

- Procházíme vstupní obraz bod po bodu, dokud nenalezneme bod T (splňuje předem definovanou podmínku např. jeho jas je 0).
- Na pixel mapy s indexem bodu T zapíšeme vzdálenost 0 (pixel je totožný s bodem).
- Od bodu T v mapě postupně procházíme všechny sousední body a počítáme jejich vzdálenost – podle vybrané metriky – od bodu T . Pokud je vypočítaná vzdálenost menší než uložená vzdálenost v daném bodě, nahradíme ji novou vzdáleností. Takto postupujeme až do maximálně povolené vzdálenosti. Pokud v některém směru vyplňování bude gradient uložených vzdáleností klesat, daný směr můžeme přeskočit.
- Opakujeme od prvního bodu.

Tento postup je poměrně rychlý, pokud je ve vstupním obraze málo bodů (počet bodů můžeme zredukovat např. hranováním). Další způsob, který je vhodný spíše pro velký počet bodů, popisuje následující postup:

1. Procházíme vstupní obraz a pro každý bod S obrazu zopakujeme následující body:

2. Pokud bod S je bod objektu (např. jeho jas je 0), potom do mapy V zapiš 0 a pokračuj bodem 1.
3. Postupně prohledej (do vzdálenosti max – na začátku nastavena na hodnotu např. 255) čtvercové okolí (obvod čtverce) bodu S , dokud nenalezneš bod.
4. Urči nové maximum: vzdálenost nalezeného bodu od bodu S , zaokrouhlovaně nahoru a ulož minimum.
5. Pokračuj v prohledávání čtvercového okolí bodu S do nového maxima, pokud naleznáš bod, srovnej a ulož minimum.
6. Po dosažení maximální vzdálenosti při prohledávání okolí bodu S je uložena vzdálenost nejbližšího bodu k bodu S .

Třída Hausdorff

Při vytváření objektu použijeme referenční snímek. Konstruktor z něho vytvoří vzdálenostní mapu (jedním z výše uvedených způsobů). Poté můžeme jednoduše zavolat metodu *find* s parametrem hledané šablony. Výsledky poloh minimálních vzdáleností jsou seřazeně uloženy ve vektoru bodů.

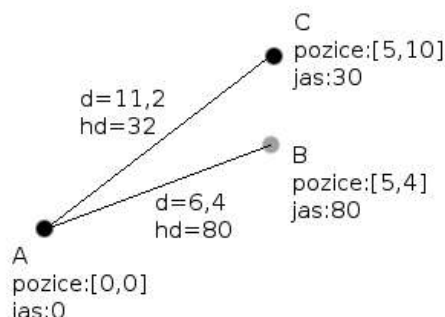
6.3.2. Modifikace HD

V teoretickém rozboru bylo představeno několik základních modifikací HD. Ve většině případů se jedná o záměnu výběru maxima ve vzorci č. 17. Pro tyto modifikace není tedy vytvořena zvláštní třída, modifikace jsou zahrnuty ve třídě Hausdorff. Výběr použité modifikace se provádí přímo v programu a po její změně je potřeba celý program znovu sestavit.

6.3.3. Hybridní Hausdorffova vzdálenost HHD

Protože výpočet vzdálenosti dvou bodů u HHD je závislý i na jasových úrovních bodů, nemůžeme přímo použít vzdálenostní mapu tak, jak v předchozích případech. Ani rozšíření mapy o jasový údaj nejbližšího bodu by nám v této situaci nepomohlo. Důvod je následující: pokud chceme najít nejbližší hybridní vzdálenost od bodu A , podle vzdálenostní mapy bychom našli geometricky nejbližší bod a jeho jas a vypočetli hybridní vzdálenost. Tento přístup je však špatný, protože se může stát, že geometricky nejbližší bod bude mít rozdílnou jasovou úroveň, než bod A a bude mít tedy větší výslednou hybridní vzdálenost, než jiný bod, který leží geometricky dále, ale jeho jas je podobný jasů bodu A . My se však snažíme najít minimální hybridní vzdálenost. Tento

případ ilustruje obrázek č. 10. Podle klasického výpočtu vzdáleností by byl bodu A nejbližší bod B, při použití hybridní vzdálenosti bude nejbližší bod C.



Obr. č. 10: Demonstrace výpočtu klasické HD a hybridní HD.

Vzdálenostní mapu, která nám značně urychlovala výpočet minimální vzdálenosti, u této metody použít nemůžeme. Minimální vzdálenost budeme muset pro každý bod hledat.

Při testování se ukázalo, že neustálé vyhledávání minimálních vzdáleností je velmi zdlouhavé. Jako vhodný postup se ukázalo uložení všech bodů a prohledávání pouze tohoto pole. Další zrychlení provedeme snížením počtu prvků v tomto poli. K tomu byly vybrány dvě metody, které měly za cíl obraz předzpracovat tak, aby obsahoval minima bodů. Jednalo se o hranové detektory (Sobel, Canny) a detekci významných bodů (Harrisův operátor). Všechny algoritmy kromě Cannyho hranového detektoru byly implementovány v rámci této práce. Cannyho hranový detektor byl použit z knihovny OpenCV.

Hybridní Hausdorffova vzdálenost také normalizuje gradient obrazu. K tomuto účelu byla vytvořena metoda, která pro výpočet gradientu používá jednoduchý gradientní operátor. Masku operátoru je ve vzorci č. 38. K výpočtu gradientu stačí provést konvoluci masky a obrazu.

$$\nabla I(i, j) = [-1 \ 0 \ 1] \quad \text{vzorec 38}$$

Pomocí uvedené masky spočítáme gradient v ose y, to však není příliš výhodné. Lepším řešením je využít jiné masky, která počítá gradient v obou osách zároveň, například Sobel (vzorec č. 4), Robinson apod.

Vzhledem ke kvalitě snímků, přípravy scény, atd. umožňuje vytvořená třída tento krok vynechat a počítat Hybridní Hausdorffovu vzdálenost pomocí vzorce č. 26. Která

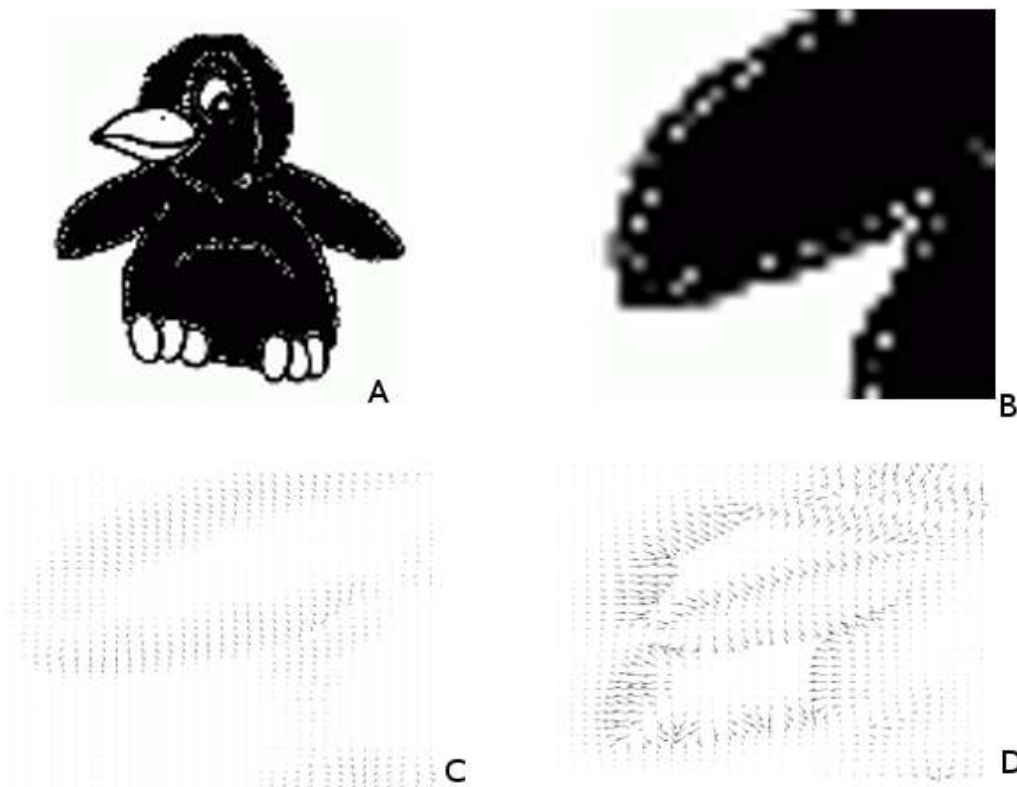
metrika se k výpočtu použije, je na volbě uživatele. Nejčastěji se však používá Eukleidovská metrika.

Pokud máme v obraze malý počet bodů (např. vlivem předzpracování), je nevýhodné prohledávat okolí každého bodu. Lepším způsobem je všechny body obrazu uložit (registrovat) do vektoru (včetně jasové složky) a hledat minimální vzdálenost vypočítáním všech vzdáleností bodů uložených ve vektoru. Postup je implementován i do klasické HD.

6.4. Optický tok

K rychlému ověření vhodnosti této metody byla implementace provedena nejprve v programu Matlab. V případě kladného zhodnocení by metoda byla implementována do C++. Z metodiky můžeme pro výpočet optického toku použít dvě metody (viz. kapitola 5.6.). Postupně byly naprogramovány a následně ozkoušeny obě metody. Výsledek obou metod je vektorové pole představující rychlost každého pixelu. Pro určení posunu šablony je vhodné vypočítat průměr z rychlostního pole po odstranění nulových rychlostí.

Testování probíhalo na uměle vytvořených snímcích jako např. obr. č. 11, na kterém můžeme vidět šablonu a detail rychlostního pole pro posuny 5 a 15 px vypočteného pomocí metody Horn and Schunck.



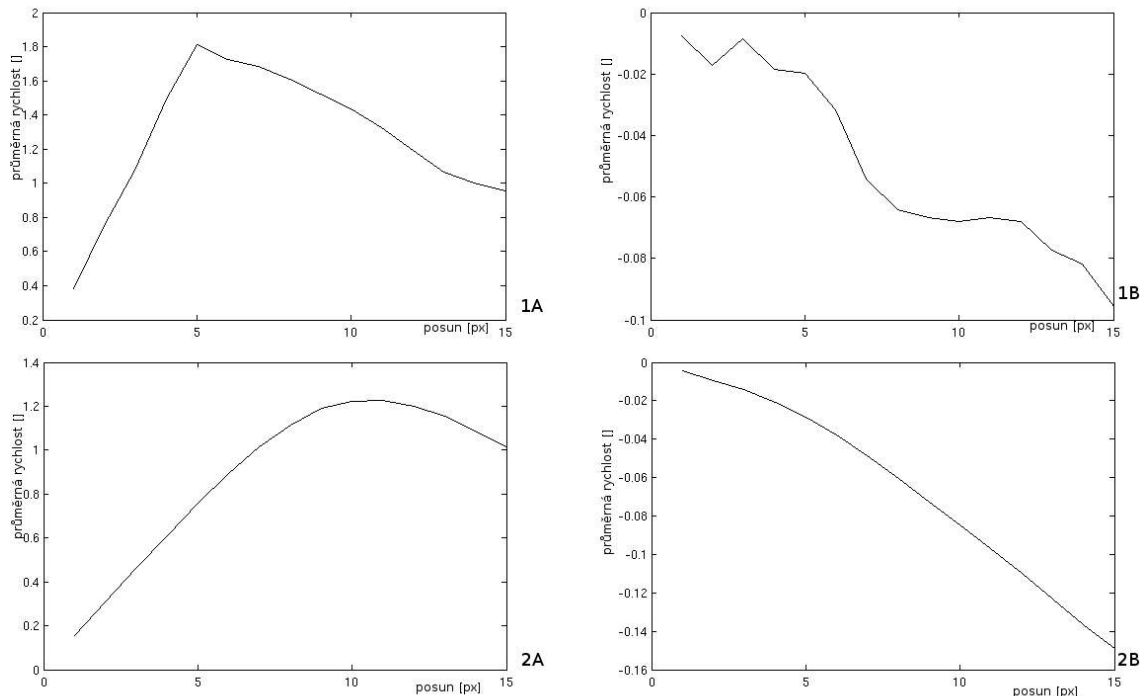
Obr. č. 11: Testovací snímek optického toku.

A - šablona; B - detail; C, D - detail optického toku pro posun C - 5 px, D - 15 px.

Chování optického toku pro výpočet posunu šablony je nejlépe vidět na grafu závislosti posunu šablony a vypočítanému posunu. Pro přehlednost je šablona posunována pouze v jedné ose. Z grafů si můžeme všimnout, že je detekován malý pohyb i v ose, která zůstává neposunutá. Dále lze vidět, že použitelnost této metody je omezená maximálním posunem. Toto chování vyplývá z teorie, kdy jsme stanovili omezující podmínku homogenality gradientu. Proto je metoda převážně určena pro zpracování sekvence snímků a tedy je tato vlastnost žádoucí. Zvětšení maximálního možného posunu lze tedy docílit rozmazáním obrazu (obr. č. 12). Tím můžeme dosáhnout i dvojnásobně velké vzdálenosti kdy k sobě odpovídající body ještě „přilnou“. Překročení maximálního posunu lze vidět i na obrázku č. 11D, kde vektory rychlosti mají špatný směr i velikost.

Druhá metoda výpočtu (Lucas and Kanade), byla testována za stejných podmínek. Výsledky však byly ještě méně uspokojivé. Metody založené na optickém toku by se tedy daly využít v kombinaci s jinou metodou vyhledání šablony, kterou by se odhadla pozice šablony a pomocí optického toku vyhledala její přesná poloha. Toto

řešení se (s ohledem na výpočetní náročnost) nezdá být pro praxi ničím výhodné. Proto algoritmy založené na optickém toku nebyly dále testovány a implementovány v jazyku C++.



Obr. č. 12: Průběhy optického toku v závislosti na posunu.
 1 - rozostření Gauss ($\sigma = 1$, maska 5x5),
 2 rozostření Gauss ($\sigma = 2.5$, maska 15x15), A - osa X, B - osa Y.

6.5. Kalibrace měření

Pokud chceme využít počítačové zpracování v praxi pro nějakou aplikaci (např. měření vzdálenosti), musíme se snažit pořídit kvalitní snímky tak, abychom zjednodušili proces předzpracování obrazu, který je v některých případech velmi složitý a náročný na výpočty. Samotný výběr a nastavení kamery tedy tvoří klíčovou roli. V našem případě se jedná o kvalitní průmyslovou kameru od firmy Basler s objektivem firmy Cinegon.

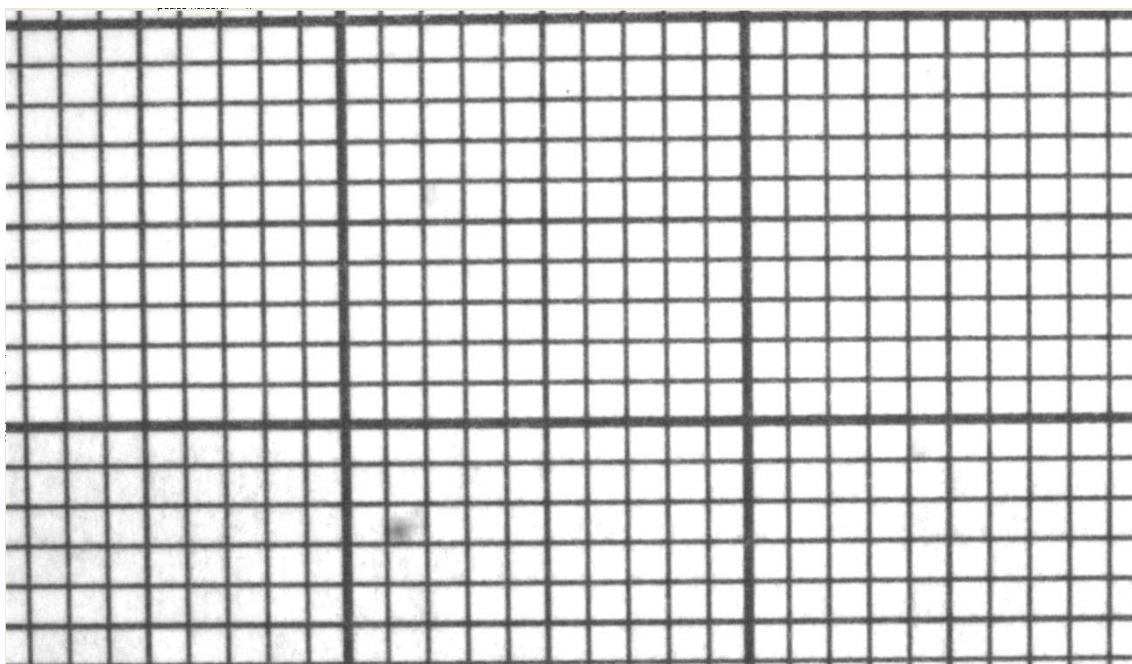
Nastavení polohy kamery ovlivňuje výslednou přesnost rozpoznání šablony při přepočtu na milimetry. Jako příklad uvedeme výpočet pro reálný objektiv Cinegon 1.8/16 f16,4 mm F1.8-22, kdy máme za cíl dosáhnout přesnosti 50 μm . Jako snímač použijeme kameru Basler acA1300-30gm C 1296 x 966 30 Monochrome CCD 1/3.

Pokud zanedbáme chyby objektivu a snímače, můžeme použít zjednodušené vztahy popsané v kapitole 2.2.

Výpočet, parametry a nastavení kamery:

- požadovaná přesnost = 50um,
- ohnisko = 16,4mm,
- rozlišení snímače = 1296x966,
- velikost 1 pixelu snímače = 3,75um,
- viditelné okno s požadovanou přesností $[x,y] = \{50 * 1296; 50 * 966\} = \{64,8; 48,3\}[\text{mm}]$
- velikost snímacího chipu $[x,y] = \{3,7 * 1296; 3,7 * 966\} = \{4,797; 3,57\}[\text{mm}]$,
- pro tenkou čočku přibližně platí vzorec č. 2,
- potřebné zvětšení v ose x vypočítáme: $Z_x = \frac{4,797}{64,8} = 0,074[-]$,
- potřebné zvětšení v ose y vypočítáme: $Z_y = \frac{3,57}{48,3} = 0,0739[-]$,
- pro další výpočet vybereme zvětšení, které splní požadovanou přesnost v obou osách $Z = \min \{Z_x, Z_y\}$,
- z toho vzdálenost objektu od středu čočky

$$s = \frac{f}{Z} + f = \frac{16,4}{0,074} + 16,4 = 238[\text{mm}] \quad .$$



Obr. č. 13: Snímek milimetrového papíru pořízený ze vzdálenosti 82 mm.

Z výše popsaného postupu vychází, že je kamerová sestava schopna dosáhnout dané přesnosti na pixel při vzdálenosti $s=238$ mm. Dále se můžeme přesvědčit, že se stejnou sestavou může být přesnost daleko vyšší. Nyní byl pořízen snímek milimetrového papíru ve vzdálenosti $s_m=82$ mm od objektivu (obrázek č. 13).

Změřená vzdálenost 27 mm odpovídá 1230 px (změřeno programem Gimp).

Z těchto údajů lehce zjistíme, že na 1 pixel připadá $\frac{27}{1230}=21,9 \mu m$.

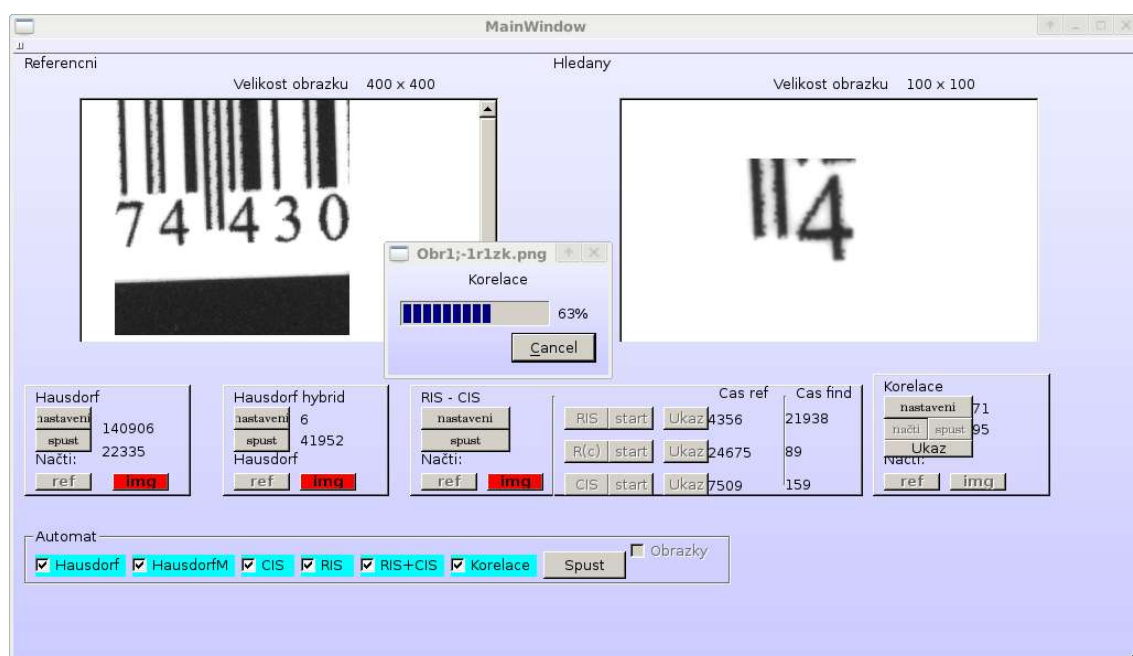
Pro ověření můžeme postupovat opačně. Pokud chceme přesnost $22 \mu m$, zvětšení v ose y musí být $Z = \frac{4,797}{966 * 0,022} = 0,23[-]$ a tedy vzdálenost

$s = \frac{16,4}{0,2257} + 16,4 = 86 \text{ mm}$. Vidíme, že naše výpočty se liší oproti skutečnosti velmi málo (odlišnost je zejména způsobená zaokrouhlením, nesprávným odečtením hodnot vzdálenosti kamery, počtu pixelů atd...).

6.6. Testovací program

Pro jednoduché a přehledné testování výše zmíněných algoritmů byl vytvořen grafický program. Ten byl psán v programovacím jazyce C++ s využitím grafické

multiplatformní knihovny QT4. Program je rozdělen do několika grafických tříd. Jednou ze tříd je třída pro načítání a zobrazení obrázků ze souboru. Podporovány jsou různé formáty (jpg, png, gif atd.). Uživatel tak jednoduše a přehledně vybere uložený obraz, který chce použít k testování. Další třídy v sobě zahrnují implementace výše uvedených tříd. U každé třídy lze speciálně nastavit typ předzpracování a další parametry. Dále program zobrazuje čas k vytvoření referenčního snímku (např. výpočet vzdálenostní mapy u Hausdorffovy vzdálenosti) a čas samotného vyhledání šablony. Dále je možné zobrazit vyhledanou šablonu na referenčním obraze a tím se vizuálně přesvědčit o správnosti výsledků. V tomto programu může uživatel volit různé metody předzpracování a jejich kombinace, výsledek předzpracování se okamžitě zobrazí, což velmi urychluje práci s jeho nastavením. Testovací program je znázorněn na obrázku č. 14.



Obr. č. 14: Testovací grafický program.

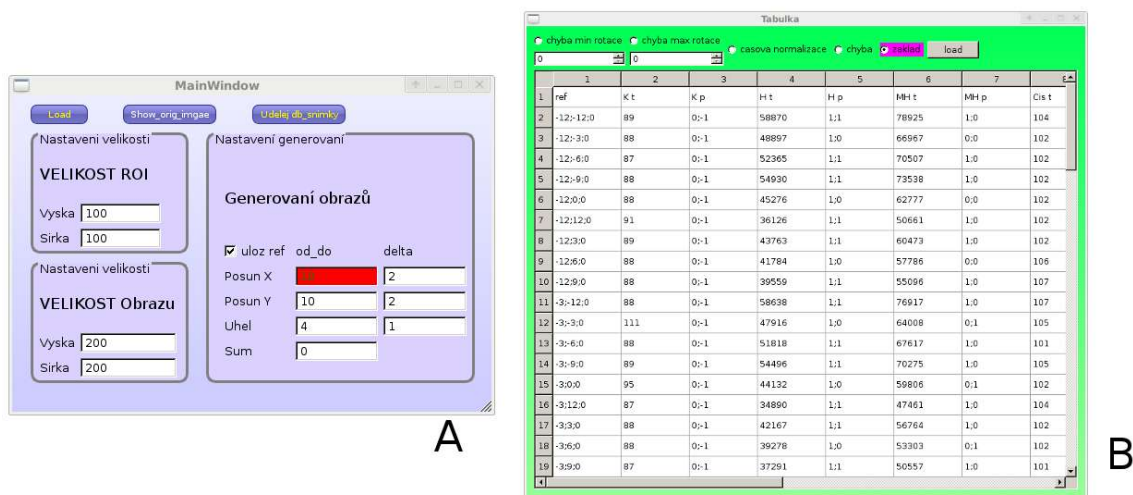
6.7. Generátor snímků, automatické testování

K ověření funkčnosti a vlastností jednotlivých metod je potřeba provést testy na velkém počtu snímků. Pro tento účel byl v rámci této práce vytvořen program, který po

načtení snímků a zadání parametrů vytvoří sadu testovacích snímků. Program umožňuje k vygenerovaným snímkům přidat šum, rotaci, translaci. Vygenerované testovací snímky jsou pojmenovány tak, aby bylo jasné jakým způsobem snímek vznikl. To bude užitečné při ověřování správnosti výpočtů metod.

Po vygenerování sady snímků ji můžeme začít testovat. Nejprve musíme nastavit vhodné předzpracování obrazu. Až jsme s ním spokojeni, začneme se samotným testováním. Můžeme postupně spouštět jednotlivé metody na každý testovací snímek a zapisovat si hodnoty. Tento způsob je však pro testování většího počtu snímků nepohodlný, proto byla do programu přidána třída, která se o testování stará. Do souboru ukládá jak výslednou pozici, tak i potřebný čas na její výpočet. Při spuštění automatického testování se použije aktuální předzpracování snímků. Je tedy důležité program před spuštěním správně nastavit a přesvědčit se, že je pro daný obraz nastavení vhodné. Po spuštění automatu vybereme zdrojový adresář, z kterého se budou načítat testovací snímky a zvolíme, které metody použít.

Po skončení automatického testování bude vytvořen soubor s výsledky. Pro jednodušší orientaci a jejich reprezentaci byl vytvořen program, který zobrazí požadované údaje, například rychlost výpočtu, chybu výpočtu v jednotlivých osách, celkovou chybu výpočtu (s rotací, bez rotace) a podobně. Výsledky jsou zobrazeny v přehledné tabulce, která umožňuje jejich přenesení do jiných dokumentů. Oba programy jsou zobrazeny na obr. č. 15.



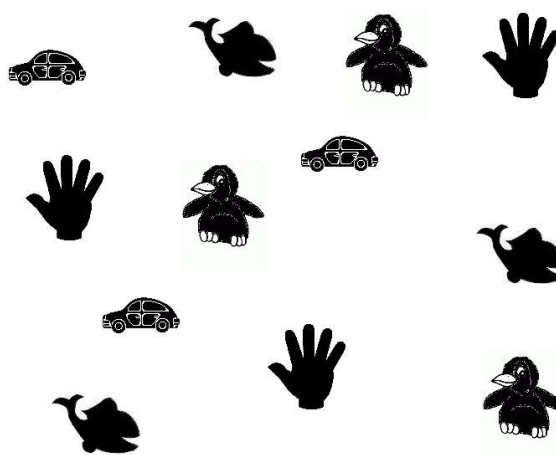
Obr. č. 15: A - Generátor snímků, B - Výsledková tabulka.

7. Testování a srovnávání algoritmů

Testování bylo prováděno pomocí speciálně vytvořeného programu, který umožňuje jednoduché nahrávání snímků, nastavení parametrů vyhledávání a předzpracování a také zobrazení výsledků. Testy byly prováděny na několika snímcích, nejprve na umělých (vytvořené v PC) a později na reálných snímcích pořízených výše uvedenou kamerou.

7.1. Testování na umělých snímcích

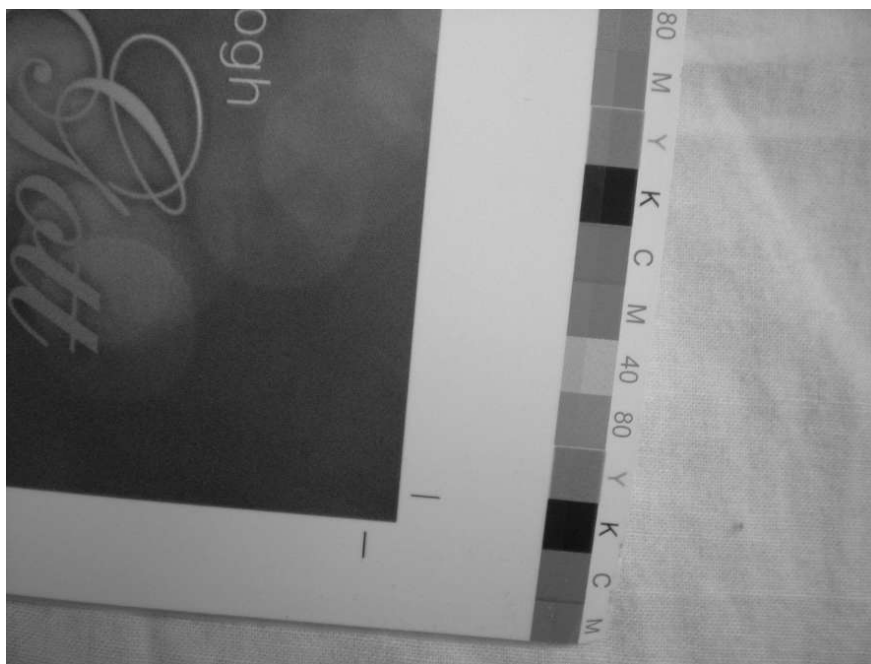
První testování bylo provedeno na umělých snímcích. Jeho cílem bylo ověření funkčnosti naprogramovaných algoritmů a odladění testovacího programu. Na obrázku č. 16 vidíme jeden z testovacích snímků o velikosti 800x600 pixelů. Jako šablony byly použity jednotlivé objekty na obrázku. Nalezení pozice šablony se u všech metod provádí pomocí hledání extrému. Metody vrací seřazený vektor bodů, které reprezentují pravděpodobnost výskytu šablon. Pokud je v obraze uvedena šablona vícekrát, budou její obrazy figurovat na prvních pozicích vektoru výsledků.



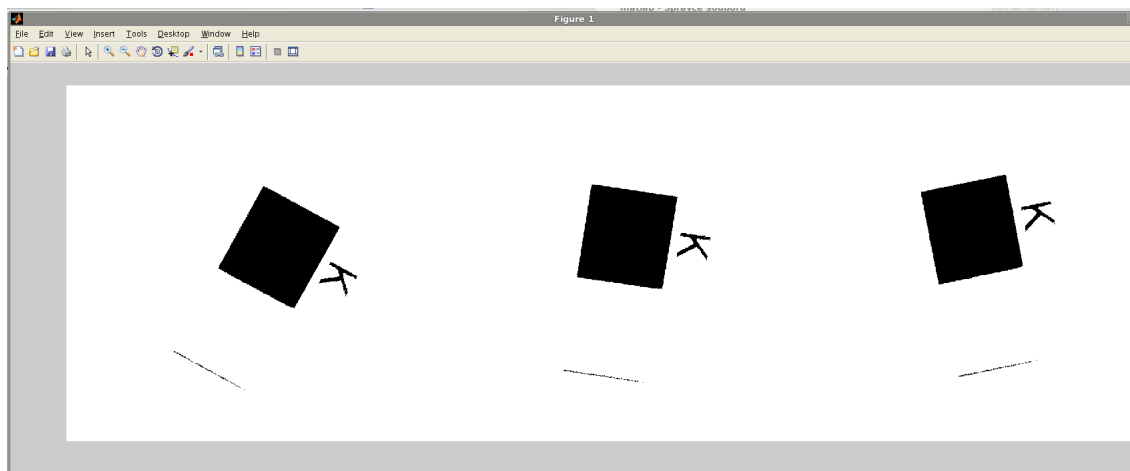
Obr. č. 16: Testovací snímek.

7.2. Testování posunu na reálných datech

První testování bylo provedeno na metodě korelace v programu Matlab. Velikost testovacích šablon byla 400 pixelů. Testované natočení bylo v intervalu $\langle -10^\circ; 10^\circ \rangle$. Snímek, na kterém byl algoritmus testován, je na obrázku č. 17. Obraz byl předzpracován pomocí prahování, práh byl stanoven podle minima histogramu. Z takto předzpracovaného obrazu byla vybrána oblast o velikosti 400 px a z této oblasti byl vytvořen referenční snímek. Na obrázku č. 18 je zobrazena první část tohoto snímku, zbývající části vzniknou postupným vkládáním dalších rotací.

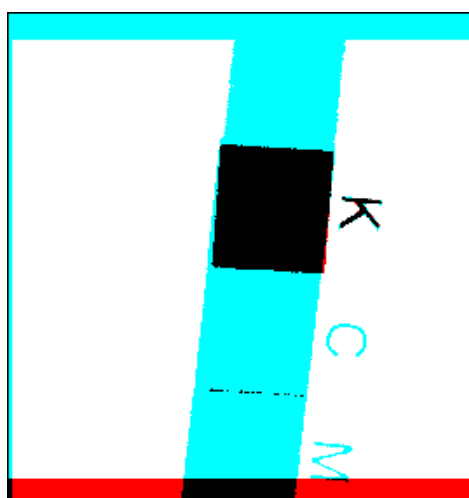


Obr. č. 17: Vyfocený reálný snímek kamerou Basler.



Obr. č. 18: Referenční snímek s třemi rotacemi.

Pomocí maxima z korelace lze určit úhel natočení a pozici šablony. Výsledek je znázorněn na obrázku č. 19, na něm vidíme překrývající se obraz v jedné z RGB vrstev se šablonou v další RGB vrstvě. Drobné chyby jsou způsobeny tím, že šablona a obraz pocházejí ze dvou různých snímků.



Obr. č. 19: Výsledek určení polohy šablony pomocí korelace.

Pro další testování byla metoda korelace použita bez doplnění o určení rotace tak, aby její výsledky byly srovnatelné s ostatními metodami, které nejsou vůči rotaci invariantní. Testování proběhlo na 13 různých reálných snímcích. Ty byly vybrány tak, aby obsahly co největší rozmanitost druhů šablon, týkající se daného praktického úkolu. Vybrána byla například šablona tiskařské značky, dále značka pro kalibraci barev, ale i obecné vzory jako například nápis, grafika apod. Ukázka šablon je na obrázku č. 20.

(V šablonách se objevil i snímek s texturou, který některým metodám dělal problémy.)
Všechny šablony jsou uloženy v digitální příloze.



Obr. č. 20: Ukázka některých referenčních šablon.

V tabulkách č. 1 a č. 2 jsou uvedeny výsledky testů bez rotace, testovaný posun byl 10 % - 30 % velikosti šablony v obou směrech obou os. Tento test obsahoval celkem 1070 šablon vytvořených ze zmíněných 13 referenčních snímků.

Číslo snímku	Průměrný čas vyhledání šablony [s]					
	Korelace	HD	HHD	CiSS	RaSS	RC
1	0,16	20,55	14,21	0,16	39,02	0,17
2	0,02	14,36	19,54	0,01	0,60	0,02
3	0,13	6,46	8,02	0,20	36,00	0,24
4	0,19	35,00	50,65	0,28	182,70	0,37
5	0,10	41,39	199,86	0,10	121,73	0,15
6	0,05	13,55	5,27	0,05	91,33	0,13
7	0,09	30,89	39,76	0,11	81,51	0,19
8	0,09	43,51	57,96	0,10	29,33	0,17
9	0,16	12,35	18,40	0,23	58,08	0,33
10	0,18	39,43	317,64	0,22	56,83	0,31
11	0,06	18,60	14,16	0,06	10,49	0,11
12	0,09	23,43	0,00	0,13	25,49	0,29
13	0,14	37,54	0,00	0,19	57,54	0,48
Průměrný čas [s]	0,11	25,93	57,34	0,14	60,82	0,23

Tab. č. 1: Výsledky průměrné časové náročnosti jednotlivých metod.

Tabulka č. 1 zobrazuje průměrný čas jednotlivých metod. Časově nejhůře dopadly metody RaSS a HHD. Naopak nejrychlejší jsou metody korelace a CiSS.

Zpoždění metody RaSS je způsobeno samotnou její podstatou. Čím více přímk v programu nastavíme, tím déle trvá prohledávání obrazu, neboť při porovnávání přímk je musíme příslušným způsobem rotovat.

Časovou náročnost HHD a potažmo i HD způsobilo nevhodné nastavení předzpracování snímků. Protože obě metody pracují na stejném základu, bylo při prvních testech nastavení předzpracování podobné. To se však později ukázalo jako nevhodné. U HHD se musí ve vstupním obraze hledat významné body a obraz dále vyprahovat pomocí vzorce č. 3. Minimální vzdálenost bodů se potom počítá pomocí registrace (vysvětleno v kapitole 6.3.3.). U klasické HD je však vhodnější jiný postup. Ve vstupním obraze nejprve hledáme hrany a poté pomocí vzdálenostní mapy určíme minimální vzdálenosti. Takto nastavené metody podávají výrazně lepší výsledky (zvláště co se týče časové náročnosti). Tento postup předzpracování byl použit v dalších testech.

Poměrně dobrých výsledků v tomto testu dosáhly metody CiSS a kombinace RC. Tyto metody mají oproti metodám založeným na HD výhodu jednoduchosti nastavení. Volíme si počet kružnic (přímk) a jejich průměr (délku). Tyto parametry ovlivňují rychlost a úspěšnost algoritmů.

Číslo snímku	Počet šablon	Počet chybně nalezených šablon					
		Korelace	HD	HHD	CiSS	RaSS	RC
1	23	0	0	0	0	0	0
2	121	0	0	0	3	45	1
3	121	0	0	0	0	0	0
4	81	0	52	48	0	0	0
5	81	0	0	0	0	0	0
6	81	0	0	0	0	0	0
7	81	0	0	0	0	0	0
8	81	0	0	0	0	0	0
9	169	4	161	165	0	1	0
10	43	0	0	17	0	0	0
11	93	0	36	30	1	5	0
12	7	0	1	4	0	0	0
13	88	0	9	83	0	0	0
Úspěšnost nalezení [%]		99,82	94,73	92,82	99,73	96,68	99,94

Tab. č. 2: Výsledky testování rozpoznání šablon při jejich posuvu.

Tabulka č. 2 zobrazuje procento úspěšnosti jednotlivých metod. Téměř sta procent dosahuje metoda RC. Ta by měla dosahovat stejných výsledků jako metoda RaSS, pokud bychom nastavili stejné parametry, to však z časových důvodů nebylo možné. U metody RaSS bylo zadáno 36 přímek o velikosti 30 pixelů, u metody RC bylo nastaveno 180 přímek s maximální možnou velikostí (většinou 40 pixelů). Samozřejmě bylo také vyzkoušeno hledání šablon metodou RaSS se stejnými parametry jako RC, ale výsledek jednoho výpočtu trval v řádech desítek minut. Proto při testech bylo používáno již zmíněné nastavení, které umožňovalo metodu otestovat, tím však utrpěla na své úspěšnosti.

Nejnižší úspěšnost má HHD. Podle teorie by však měla tato metoda mít minimálně stejné výsledky jako HD. To by jistě platilo, pokud bychom použili stejné metody předzpracování pro obě metody v průběhu všech testů. Při testování se však ukázalo vhodnější nastavení HD (využití hran a vzdálenostní mapy), proto má tato metoda lepší výsledky. U metody HHD vzdálenostní mapu použít nelze, rychlost tedy můžeme ovlivnit pouze počtem bodů v obraze. Velkou redukcí bodů nám zajistí vyhledání významných bodů v obraze. Tato technika však jistým způsobem degraduje jasovou informaci, na které je HHD založena. Vliv jasové složky je tedy minimalizován a metoda dává podobné výsledky jako klasická HD. Vyhledávání šablony pomocí významných bodů se však v některých případech negativně projevilo na výsledcích.

7.3. Další testování

Další testovací sada snímků byla tvořená translací (až 15 % z velikosti šablony) a malou rotací ($0^\circ - 4^\circ$). V této sadě bylo celkem 3030 různých pozic šablon. Výsledky jsou uvedeny v tabulce č. 3. Z tabulky byly odstraněny hrubé chyby, způsobené nevhodným nastavením. Podle předpokladů byla nejuspěšnější metoda RC, která vychází z kombinací metod CiSS a RaSS, které měly úspěšnost rovněž přes 90 %. Poměrně dobré přesnosti dosáhla metoda HD.

Číslo snímku	Počet šablon	Počet chybně nalezených šablon					
		Korelace	HD	HHD	CiSS	RaSS	RC
1	93	4	35	-	-	-	-
2	968	145	341	326	63	422	7
3	484	-	178	-	2	52	0
4	81	54	17	43	-	-	-
5	36	1	10	27	0	0	0
6	36	13	11	0	0	0	0
7	36	24	3	28	0	0	0
8	36	17	1	17	0	0	0
9	81	0	31	-	2	0	0
10	36	4	0	15	0	0	0
11	186	103	58	48	92	22	10
12	161	-	67	89	33	37	0
13	796	168	40	-	0	17	0
Úspěšnost nalezení [%]		70,33	78,62	58,53	92,79	91,70	99,45

Tab. č. 3: Výsledky rozpoznávání šablon s posuvem a rotací.

V následující tabulce (tabulka č. 4) si můžeme prohlédnout rychlost vyhledání šablon metodou HD a HHD po správném nastavení metod předzpracování a srovnat ji s rychlostí v tabulce č. 1. Vidíme, že nyní je rychlost HHD vyšší. To je z důvodu využití vhodných významných bodů v šabloně tak, aby neutrpěla přesnost (tedy snaha o to, aby největší počet významných bodů byl právě v šabloně).

Číslo snímku	Průměrný čas vyhledání šablony [s]	
	HD	HHD
1	16,44	21,13
2	21,97	27,78
3	20,69	1,21
4	12,94	15,07
5	20,24	2,15
6	12,13	0,63
7	20,21	11,73
8	13,12	17,93
9	18,53	5,98
10	13,19	43,74
11	19,60	3,20
12	20,09	1,25
13	19,64	13,20
Průměrný čas [s]	17,60	12,69

Tab. č. 4: Průměrný čas vyhledání šablony metod HD a HHD po správném nastavení parametrů.

Další test prověřil odolnost metod vůči rotaci ($0^\circ - 60^\circ$) a posunu šablony (o 60 % z velikosti šablony) na 244 snímcích, výsledky jsou zobrazeny v tabulce č. 5. Je vidět, že nejlepšími výsledky dosahuje metoda CiSS, dále pak metoda RC. To se však dalo očekávat, protože tyto metody jsou invariantní vůči rotaci.

Číslo snímku	Počet šablon	Počet chybně nalezených šablon					
		Korelace	HD	HHD	CiSS	RaSS	RC
1	61	40	38	53	40	34	40
2	61	40	56	55	20	20	20
3	61	58	41	60	0	29	4
4	61	37	26	38	0	17	0
Úspěšnost nalezení [%]		28,28	34,02	15,57	75,41	59,02	73,77

Tab. č. 5: Výsledky vyhledání šablon při rotaci ($0^\circ - 60^\circ$) a posun šablony (60% z velikosti šablony).

V dalším testu budou metody testovány na odolnost vůči šumu. Výsledky můžeme vidět v tabulce č. 6. Testy byly prováděny na několika šablonách, které byly zašuměny náhodným šumem z 5 % a v druhém případě z 15 %. Vidíme, že většina metod si s tímto šumem poradila.

Číslo snímku	Umělý šum [%]	Počet šablon	Počet chybně nalezených šablon					
			Korelace	HD	HHD	CiSS	RaSS	RC
1	5	31	0	9	-	0	0	0
2	5	31	0	0	2	0	0	0
3	5	31	0	-	-	0	1	0
4	5	31	0	0	12	0	0	0
1	15	31	0	-	19	0	0	0
2	15	31	0	0	20	0	0	0
3	15	31	1	12	-	16	28	4
4	15	31	0	0	16	0	0	0
Úspěšnost nalezení [%]			99,60	88,71	55,48	93,55	88,31	98,39

Tab. č. 6: Výsledky testování odolnosti metod proti šumu.

V následující tabulce (tabulka č. 7) jsou zobrazeny časy jednotlivých metod potřebné k načtení referenčního snímku. Nejdelší čas potřebuje HD, která v této době vypočítává vzdálenostní mapu. Nejnížší čas potřebuje HHD, která vstupní obraz pouze načte a neprovádí žádné výpočty.

Číslo snímku	Průměrný čas pro načtení referenčního snímku [ms]					
	Korelace	HD	HHD	CiSS	RiSS	RC
1	71	26871	12	13177	17209	37350
2	6	12451	2	1018	2590	36271
3	48	7175	8	13891	12695	31806
4	25	117735	20	44077	26353	43221
5	64	114089	2	21338	20762	38452
6	18	120023	3	3401	5572	39245
7	33	46382	6	7757	7694	41245
8	36	92478	6	7108	4753	31870
9	34	43310	12	7934	5934	35242
10	98	165828	8	13491	8175	39459
11	65	197630	1	8210	3788	36541
12	39	112647	17	7710	4523	37985
13	119	507316	10	11996	5308	41245
Průměrný čas [ms]	50,46	120302,69	8,23	12392,92	9642,77	37687,08

Tab. č. 7: Průměrný čas načtení snímku.

Z provedených testů můžeme usoudit, že nejvhodnějším kandidátem pro praktické použití, pokud nevyžadujeme invarianci vůči rotaci, je metoda korelace. Tato metoda je poměrně jednoduchá na implementaci, nevyžaduje téměř žádné nastavení a podává dobré výsledky s velmi malými časovými nároky. Dalším vhodným kandidátem je metoda RC, která je invariantní vůči rotaci. Tato metoda má tu nevýhodu, že je potřeba přepočítat referenčního obrazu (do CiSS respektive RaSS prostoru), to však můžeme provést pouze jednou. Tato metoda sice již byla optimalizována na rychlejší výpočty (použití Bresenhamova algoritmu apod.), ale při případném použití bychom jistě našli větší prostor pro urychlení než u korelace. Další výhodou RC metody je lehká

představa průběhu vyhledávání a tedy i jednodušší určení vhodné šablony a jejího předzpracování.

Metoda HHD je poměrně složitější, proto i její programování má jisté rezervy. Její výhody se v testu neprojevily a pro praktické nasazení, kterým se zabývá tato práce, není příliš vhodná. Lepší výsledky vykazuje klasická metoda HD. Dalšího urychlení výpočtů by se dosáhlo využitím paralelního programování, to však nebylo použito u žádné z testovaných metod.

Do digitální přílohy jsou zařazeny všechny dílčí výsledky všech testů, včetně referenčních snímků a použitých šablon.

V posledním kroku byla pro srovnání otestována metoda pro vyhledávání šablon z knihovny OpenCV. Jednalo se o funkci *cvMatchTemplate*. Testování proběhlo obdobně jako v kapitole 7.2. , ale s menším počtem snímků. Tato knihovní funkce může provést vyhledání šablony několika metodami, v testu byla použita metoda označená makrem CV_TM_CCORR. V tomto testu byl průměrný čas vyhledávání 72 ms a úspěšnost nalezení šablony 100%. V druhém testu byly použity šablony s rotací (0° - 4°). Z tabulky výsledků (tabulka č. 8) vidíme, že tato funkce je rychlejší než metoda korelace implementovaná v této práci. V úspěšnosti je srovnatelná s metedou CiSS nebo RC. Tyto metody jsou však téměř dvakrát pomalejší.

Číslo šablony	Rychlost [ms]	Počet šablon	Počet chybně nalezených šablon
1	66,5	36	0
2	66,9	36	0
3	66,3	81	4
4	66,9	36	8
5	66,1	36	0
Průměrný čas	66,5	Úspěšnost nalezení [%]	94,07

Tab. č. 8: Výsledek testu funkce *cvMatchTemplate* z knihovny OpenCV.

8. Praktické nasazení

Nasazení některé z metod v praxi má mnohé využití. Tato práce se zabývá využitím metod pro lakovací stroj plakátů velkých rozměrů. Lakování probíhá na již potištěný plakát. Podavač papírů pomocí přísavek odebere papír ze zásobníku a vloží jej do stroje. Stroj papír zarovná podle okrajů a nanese lak na daná místa na papíře pro zdůraznění vizuálního efektu některé natisknuté části. Aby se dosáhlo správného efektu, musí být lak nanesen přesně na dané místo. Doposud se na stroji používalo zarovnávání pomocí okrajů plakátu. To však není dostatečně přesné, protože nátisk plakátu může být různě posunut. Na plakátech však bývají různé tiskařské kalibrační značky, které jsou vždy v přesné pozici s nátiskem. Těchto značek tedy využijeme pro přesné pozicování plakátu. Využijeme k tomu výše zmíněnou kameru s objektivem. Pro určení posunu je zapotřebí dvou nezávislých kamer. Zpracování proběhne v řídicím PC, které má na starosti řízení celého stroje, včetně zajištění operátorského rozhraní. PC je postaveno na platformě Windows a řídicí program je programován v jazyce C#.

Z výsledků testů jednotlivých metod byla pro tento úkol vybrána metoda korelace. Rotaci v tomto případě uvažovat nemusíme. Plakát má maximální velikost 1,5 m, maximální odchylka značky může tvořit maximálně 4 mm. Vypočítaná rotace, o kterou se v nejhorším případě značka otočí, je velmi malá a tedy ji zanedbáme, korelace se s tímto zaokrouhlením snadno vypořádá.

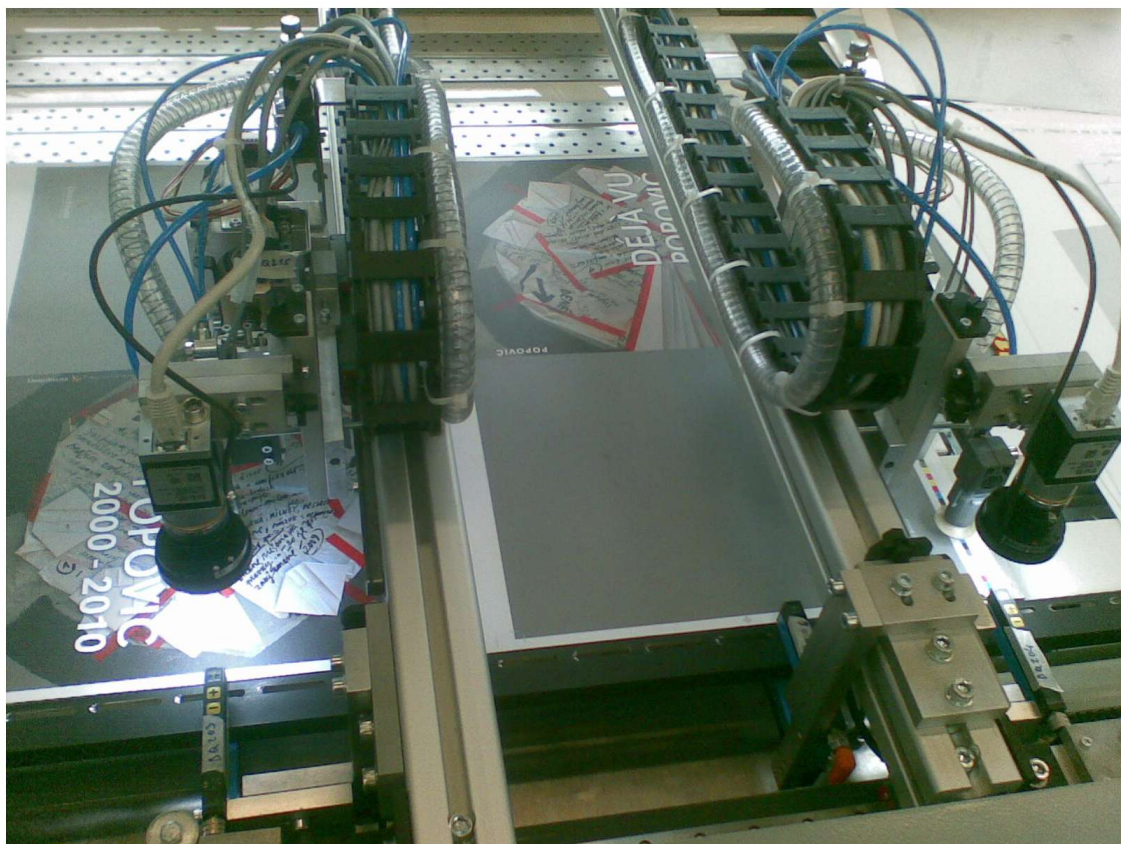
Jedním z požadavků bylo, aby výpočet byl proveden do 200 ms s přesností 30 um. Pro nastavení kamer byl použit výpočet uvedený v kapitole č. 6.5. . Metoda korelace byla přepsána do jazyka C#. Pro převod obrazu do FT bylo použito místo knihovny FFTW2 knihovny Aforge (konkrétně AForge.Math). Vstupní obraz se načítá přímo z bufferu kamery, využívá se k tomu rozhraní GigE. Kamera má také možnost externího trigování (například z PLC), které by pomohlo snížit celkový čas potřebný ke kompletnímu zpracování, toho však zatím nebylo využito. Z požadované přesnosti a rychlosti se koreluje obraz o velikost 512 px. Je tedy zbytečné přenášet z kamery větší obraz, kamera se tedy nastaví na danou velikost. Výpočet korelace a určení posunu z obou kamer probíhá nezávisle a paralelně. Celkový výpočet pro obě kamery včetně sejmutí obrazu trvá 180 ms (s odchylkou +/- 10 ms v závislosti na aktuální nastavené délce expozice a na systémových prostředcích řídicího PC). Celkový čas pro jednu kameru je řádově o 10 ms kratší, díky tomu můžeme jasně vidět, že výpočet i komunikace (dvě síťové karty) běží opravdu paralelně.

Program je dále doplněn o grafické rozhraní. Operátor při najíždění nové zakázky upraví pozici kamer tak, aby snímaly potřebnou značku. Aktuální obraz (stream) je zobrazován v plném rozlišení kamery. Po nastavení kamer na správnou pozici operátor zvolí ROI (oblast zájmu, region of interest), ve kterém se značka nachází. Program tento snímek uloží a připraví ho pro korelaci (převod s využitím DFT). Program dále nastaví pomocí API na kameře ROI a přepne ji do snímkovacího módu. Tento postup se provede pro obě kamery. Nyní je vše připraveno a po najetí nového plakátu řídicí program spustí vyhledání pozice, tím proběhne sejmutí ROI oblasti, následuje korelace a výpočet posunu. Řídicí program po obdržení výsledků z kamer posune plakát na správné místo.

Další funkcí programu je po výpočtu posunu plakátu ověřit správnost výsledků. Tento krok je důležitý, protože se mezi plakáty může objevit jiný, jinak otočený, nebo špatně natištěný plakát. Dalším důvodem pro kontrolu je fakt, že pokud posun přesáhne 1/2 obrazu, korelace selže. Selhání může být zapříčiněno např. šumem či nečistotou. V takovém případě je posun plakátu mylný a je potřeba zastavit tisk. Jak bylo řečeno, tato kontrola je prováděna v době, kdy dochází k pozicování plakátu a zahájení tisku. V případě získání podezření na špatný výsledek, je tisk okamžitě zastaven a je zobrazena výzva operátorovi k ověření. Operátor rozhodne o pokračování tisku.

Ověření správnosti výsledku je prováděno jednoduše pomocí sumy rozdílového snímku, získaného mezi posunutou šablonou a referenčním snímkem. Získaná hodnota je normalizována podělením počtu pixelů, které se podílely na sumaci rozdílového snímku.

Takto vytvořený měřicí optický systém (obrázek č. 21) byl testován několik dnů ve zkušebním provozu. V průběhu testování se objevilo několik nedostatků. První z nich bylo osvětlení. To bylo řešeno pomocí LED diod umístěných kolem objektivu. Toto světlo bylo velmi jasné a kvalitně nasvěcovalo scénu. Při tisku lesklých plakátů se však světlo odráželo zpět do kamery, což způsobovalo odlesky. Tento problém byl částečně vyřešen použitím . Druhý problém byl při posuvu plakátu. Plakát je na stroji uchycen pomocí vakuového stolu. Kamery snímají polohu plakátu vždy na koncích plakátu. Při tisku velkoformátových plakátů, které jsou nižší gramáže, dochází v jeho rozích (kde se často snímá obraz) k nepřilnutí plakátu na vakuový stůl. Pořízený snímek je pak velmi nekvalitní vlivem ohnutí rohu a také jeho vibrováním ke stolu. Tento problém je řešen posunem kamer a snímáním v místech, kde k tomuto jevu nedochází.



Obr. č. 21: Optický měřicí systém.

9. Závěr

Práce se zabývá problematikou bezkontaktní kontroly součástek pomocí kamer. V první části byl představen základní řetězec zpracování obrazu a popis jeho jednotlivých částí. V další části jsou detailněji popsány jednotlivé postupy předzpracování obrazu, jedná se o metody potlačení šumu, hranový detektor, prahování a detektor významných bodů.

Stěžejní oblastí práce je návrh měřicího systému vhodného pro technickou praxi. Pro jeho realizaci byl vybrán konkrétní praktický příklad měření posunu plakátu od referenční polohy na tiskařském stroji. Pro řešení tohoto úkolu byly v další kapitole představeny metody pro vyhledání šablon v obraze. Nejprve byly teoreticky popsány, v další části pak postupně naprogramovány a to bez použití knihoven počítačového vidění v programovacím jazyce C++. Jednalo se o techniky méně používané, jako například metoda H.Y. Kima a S.A. De Araújo, která se ukázala jako zajímavá alternativa k běžně používané metodě korelace.

Další představené metody byly metody založené na Hausdorffově vzdálenosti. Hybridní Hausdorffova vzdálenost, která vnesla do výpočtu vzdáleností i jasovou složku, mohla být považována za metodu vhodnou pro praktické nasazení. Následně se ukázalo, že díky výpočetním nárokům by byla vhodnější metoda klasické Hausdorffovy vzdálenosti, u které můžeme použít vzdálenostní mapu a tím výpočet urychlit.

Poslední metodou, kterou se tato práce zabývala, byla metoda založená na optickém toku. Tato metoda se používá pro analýzu sekvencí snímků. Proto by byla vhodná pro vyhledání šablon (detekce pohybu šablony) s malým posunem, při větším posunu šablony je metoda takřka nepoužitelná.

Další část práce se zabývá testováním těchto metod. Pro tento účel byl naprogramován GUI testovací program, pomocí něhož lze přehledně testovat snímky, nastavovat parametry předzpracování a zobrazovat výsledky. K testování byla vytvořena testovací databáze. Referenční snímky jsou pořízeny průmyslovou kamerou Basler.

Testování bylo nejprve provedeno na uměle vytvořených šablonách, to sloužilo k ověření správnosti algoritmů a doladění funkčnosti programu. K dalšímu testování byla vytvořena testovací databáze. Testy se snažili odhalit výhody a nevýhody metod a jejich robustnost. V testech bylo zjištěno, že potřeba kvalitního předzpracování obrazu je hlavně u metod založených na Hausdorffově vzdálenosti. Předzpracování obrazu u nich velmi ovlivňuje přesnost a rychlost vyhledávání šablon.

Největší problémy při testování dělala Hybridní Hausdorffova vzdálenost. Před použitím této metody je potřeba pečlivě vybrat šablonu a typ předzpracování. Dosažená úspěšnost nalezení šablony byla v provedených testech 92,8 %. O něco lépe si v testech vedla klasická Hausdorffova vzdálenost s úspěšností 94,7 %. Této metodě stačí, aby šablona obsahovala hrany, které hranový detektor zvýrazní. Tato metoda si vedla lépe i v zašuměném obraze, zde byla její úspěšnost 88,7 %, HHD měla úspěšnost 55,4 %.

Velmi dobře si v testech vedly metody CiSS a RaSS. Metoda RaSS byla v testech velmi přesná, ale kvůli časové náročnosti byl ve většině případech nastaven malý počet přímek, což se projevilo negativně a metoda dosáhla pouze 96,7% úspěšnosti, zatímco metoda CiSS měla úspěšnost 99,7 %. Pokud však metodu RaSS použijeme na body označené metodou CiSS (v testech označováno RC), pak i při nastavení většího počtu přímek získáme rychlou a přesnou metodu (99,94%). Výhoda těchto metod je, že nevyžadují speciální nastavení ani předzpracování. Tuto vlastnost má i korelace. V provedených testech to byla nejrychlejší metoda (průměrně 110 ms) a její účinnost dosahovala 99,82 %. Dvojnásobný čas (230 ms) potřebovala pro nalezení šablony kombinace metod CiSS a RaSS.

Posledním testem bylo porovnání funkce `cvMatchTemplate` z knihovny OpenCV, která slouží k vyhledávání šablon. Této funkci trvalo nalezení šablony 67 ms. Byla také přesnější při hledání šablon s posunem a rotací (94 %), než metoda korelace uvedená v této práci (70 %). Z uvedených metod měla v tomto testu lepší výsledky pouze metoda RC (99,45 %).

V poslední části se práce zabývá případnou implementací metody korelace na praktický stroj. Jedná se o tiskařský lakovací stroj, ve kterém byla pomocí jedné metody určována odchylka plakátu od referenční polohy. K snímání plakátu byly použity dvě průmyslové kamery, které byly umístěny podle výpočtů. Vhodnou metodou se jeví metoda korelace, je velmi rychlá a přesná. Vytvořený optický systém s metodou korelace splnil technické zadání, kde byl kladen důraz na rychlost, přesnost a jednoduchost. Tento měřicí optický systém byl několik dnů testován ve zkušebním provozu. Ač nebyla představenému systému vytknuta žádná vážnější vada, byl na stroj instalován jiný měřicí systém.

10. Literatura

- [1]: Ilona Kalová, Počítačové vidění - Předzpracování obrazu, prezentace, 2011.
- [2]: Jeff Donald, Chris Hurd, The Ultimate Depth-of-Field Skinnyan article by Jeff Donald, 2002.
- [3]: Dave Litwiller, CCD vs. CMOS: Facts and Fiction, 2001.
- [4]: Hlaváč V., Šonka M., Počítačové vidění, 1992.
- [5]: Žára J., Beneš B., Sochor J., Felkel P, Moderní počítačová grafika, 2004.
- [6]: D.Parks, J.P. Gravel, Corner Detection, .
- [7]: Kim, H.Y. and de Araújo, S.A., Rotation, scale and translation-invariant segmentation-freeshape recognition, 2007.
- [8]: P. Kršek, M. Španěl, Základy počítačové grafiky: Rasterizace objektů ve 2D, 2011.
- [9]: Huttenlocher, D.P., Klanderman, G.A., Rucklidge, Comparing images using Hausdorff distance, 1993.
- [10]: Dong-Gyu Sim, Oh-Kyu Kwon, and Rae-Hong Park, Object Matching Algorithms Using Robust Hausdorff Distance Measures, 1999.
- [11]: Chyuan-Huei Thomas Yang, Shang-Hong Lai, Long-Wen Chang, Hybrid image matching combining Hausdorff distance with normalized gradient matching, 2006.
- [12]: J.L.Barron and N.A.Thacker, Computing 2D and 3D Optical Flow., 2005.
- [13]: Horn B.K.P and Schunck B.G, Determining optical flow, 1981.
- [14]: <http://relivethefuture.com/code/VoronoiTest.swf> [online]

Seznam obrázků

Obr. č. 1: Řetězec zpracování obrazu.	
Zdroj [1].....	11
Obr. č. 2: Masky pro průměrování.....	18
Obr. č. 3: Předzpracování obrazu: A - původní snímek, B - potlačení šumu, C - prahování, D - hranování, E - hranování s potlačením šumu, F - významné body... ..	21
Obr.č. 4: Ukázka šablony(a-b), d-zeleně CiSS modře RaSS.	
Zdroj: [7].....	25
Obr. č. 5: Voronoiův diagram náhodně vygenerovaných bodů.	
Zdroj:[14].	
.....	28
Obr. č. 6: Hausdorffova vzdálenost množin A (zelená) a B (modrá), h(A,B) modře, h(B,A) zeleně.....	29
Obr. č. 7: Problém výpočtu správné rychlosti optického toku.....	33
Obr. č. 8: Rasterizovaná kružnice.	
Zdroj: [8].....	38
Obr. č. 9: Úsečka A-obecná úsečka, B-úsečka v prvním kvadrantu.....	40
Obr. č. 10: Demonstrace výpočtu klasické HD a hybridní HD.....	43
Obr. č. 11: Testovací snímek optického toku. A - šablona; B - detail; C, D - detail optického toku pro posun C - 5 px, D - 15 px.....	45
Obr. č. 12: Průběhy optického toku v závislosti na posunu. 1 - rozostření Gauss ($\sigma = 1$, maska 5x5), 2 rozostření Gauss ($\sigma = 2.5$, maska 15x15), A - osa X, B - osa Y.....	46
Obr. č. 13: Snímek milimetrového papíru pořízený ze vzdálenosti 82 mm.....	48
Obr. č. 14: Testovací grafický program.....	49
Obr. č. 15: A - Generátor snímků, B - Výsledková tabulka.....	50
Obr. č. 16: Testovací snímek.....	51
Obr. č. 17: Vyfocený reálný snímek kamerou Basler.....	52
Obr. č. 18: Referenční snímek s třemi rotacemi.....	53

Obr. č. 19: Výsledek určení polohy šablony pomocí korelace.....	53
Obr. č. 20: Ukázka některých referenčních šablon.....	54
Obr. č. 21: Optický měřicí systém.....	63