VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

DETEKCE ZMĚN PROSTŘEDÍ ZPRACOVÁNÍM OBRAZU Z KAMERY MOBILNÍHO ROBOTU

DETECTION OF ENVIRONMENT CHANGES USING MOBILE ROBOT CAMERA IMAGE PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Vojtěch Venglář

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Jiří Krejsa, Ph.D.

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	**Vojtěch Venglář**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	**doc. Ing. Jiří Krejsa, Ph.D.**
Akademický rok:	2015/16

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

Detekce změn prostředí zpracováním obrazu z kamery mobilního robotu

Stručná charakteristika problematiky úkolu:

Jednou z aktuálních úloh v mobilní robotice je střežení zadaného prostoru. Během provozu robot opakovaně sleduje okolní prostředí pomocí palubních senzorů, typicky kamer, ultrazvukových snímačů a laserových dálkoměrů. Základní úlohou je detekce změny prostředí v porovnání s předchozím stavem. Jedním ze způsobů takové detekce je zpracování obrazů z kamery (kamer). Podstatou bakalářské práce je nalezení vhodné metody, její implementace a ověření na reálných datech.

Cíle bakalářské práce:

1. Vypracujte rešerši s přehledem metod vhodných pro detekci změn v obraze.
2. Vyberte vhodnou metodu, s přihlédnutím k implementačním a výpočetním nárokům.
3. Vybranou metodu implementujte s využitím knihoven OpenCV.
4. Ověřte na reálných datech

Seznam literatury:

Radke, RJ; Andra, S; Al-Kofahi, O; et al., Image change detection algorithms: A systematic survey, IEEE TRANSACTIONS ON IMAGE PROCESSING, Vol: 14, Issue: 3 ,pp. 294-307, 2005

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt

Tato práce se zabývá zpracováním obrazu a jeho využitím k detekci změn v prostředí. V první části jsou uvedeny základní metody a postupy užívané ke zpracování obrazu. Ve druhé části je pak ukázka implementace řešení daného problému pomocí knihovny OpenCV.

Abstract

This thesis describes use of image processing in detection of environmental changes. The first part contains basic description of image processing methods and commonly used procedures. In the second part, an implementation of given problem solution using OpenCV library is shown.

Klíčová slova

Detekce změny prostředí, zpracování obrazu, registrace obrazu, OpenCV

Keywords

Detection of environment changes, image processing, image registration, OpenCV

Bibliografická citace

VENGLÁŘ, V. *Detekce změn prostředí zpracováním obrazu z kamery mobilního robotu.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 45 s. Vedoucí bakalářské práce doc. Ing. Jiří Krejsa, Ph.D..

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím odborné literatury a dokumentačních materiálů k dané problematice.

25. 5. 2016 ……

Vojtěch Venglář

Poděkování

Děkuji tímto doc. Ing. Jiřímu Krejsovi, Ph.D. za cenné připomínky a rady při vypracování bakalářské práce.

Obsah

1. Úvod

Vývojovým trendem posledních let je co nejvíce nebezpečných či často opakovaných pracovních činností přenechávat robotům. Robot pak může být dálkově ovládaný nebo plně automatizovaný. Dálkově ovládaný robot může být uplatněn při zneškodňování výbušnin nebo jako špionážní zařízení nebo jako hračka. Plně automatizované roboty pak najdeme například v továrnách jako obsluhu pásové výroby.

Dálkově ovládaný robot zpravidla potřebuje nějaký senzor, který bude operátorovi umožňovat orientaci a manipulaci robota v prostoru. Pro takové účely může sloužit například radar, ultrazvukový senzor vzdálenosti nebo laserový senzor vzdálenosti. Pokud ovšem operátor potřebuje přímo vidět, co se nachází před robotem (např. pyrotechnik), musí se přenášet obraz, tedy je jako senzor použita kamera. Operátor pak vidí, jako by byl v robotu a může ho navádět.

Když se rozhodneme použít robota pro sledovací účely, často budeme chtít, aby sledoval stejný prostor a my tedy mohli zjistit, co se v daném prostoru děje. Použijeme tedy robota, který bude schopen provádět cestu sám, aby operátor nemusel manuálně projíždět (případně prolétat) stejnou trasu neustále znovu a znovu. Do této oblasti můžeme zařadit například drony přelétající nad územím, mapující pohyb jednotek. Zvláštním typem sledovacího zařízení je pak stacionární kamera používaná ke hlídání bank, skladišť atd.

Ve výše uvedených případech je pro vyhodnocení obrazu stále potřeba operátora, který může kontrolovat už více obrazovek najednou, protože se nemusí starat o ovládání robota, ale stále musí on sám vyhodnocovat, co se ve sledovaném území děje.

Účelem této bakalářské práce je právě tuto práci zjednodušit. Pokusit se najít způsob, jak pouze snímáním obrazu z kamery robota vyhodnotit změnu prostředí, tedy že se například někdo vloupal do objektu, nebo že pracovník po směně zapomněl uklidit vozík s chemikáliemi a nechal ho stát v kritickém místě v případě požáru. Dále chceme tuto změnu zvýraznit na obrazovku operátora, aby jasně viděl, že se něco děje.

2. Rozdělení změn v obraze

Změny v obraze dělíme dle rozsahu na globální a lokální. Ty jsou popsány níže a znázorněny na obrázku 2.1.

2.1 Globální změny

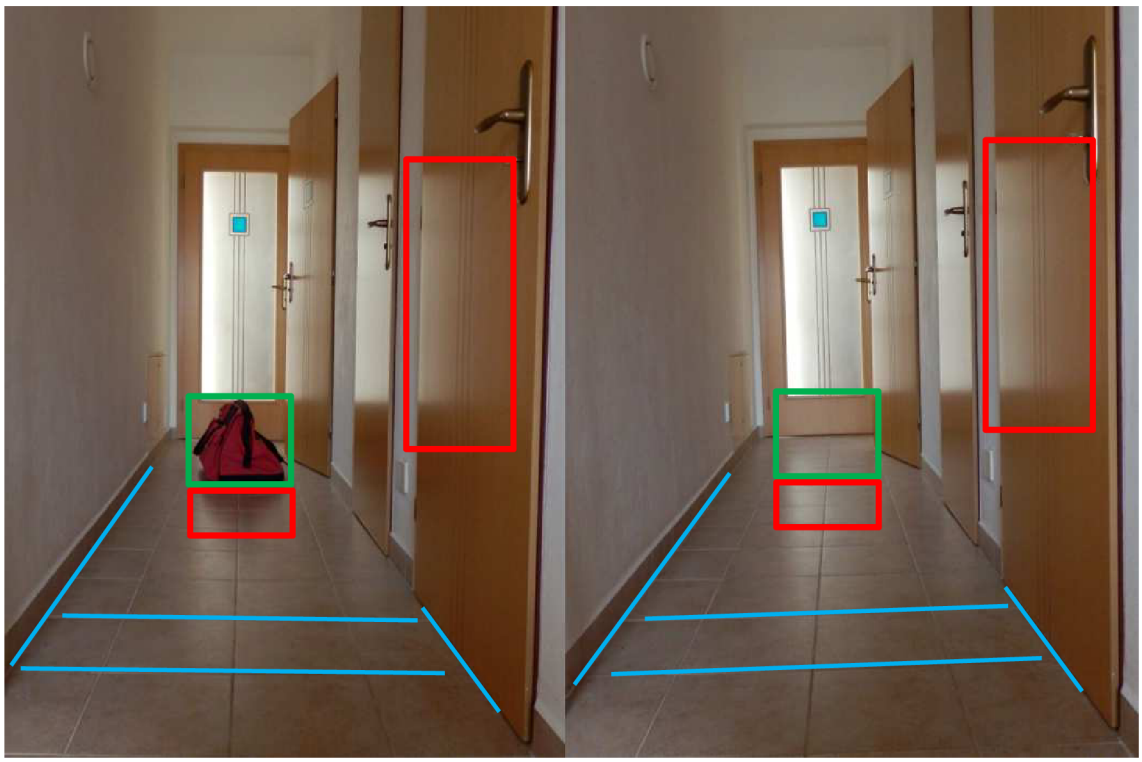
Globální změny jsou takové, které se při zpracování obrazu snažíme potlačit. Jsou to změny, ke kterým dojde například změnou osvětlení, otočením snímače nebo změnou polohy snímače. Je důležité je odstranit, aby zůstaly ke srovnání pouze změny lokální.

V obrázku 2.1 se projevují například jako změna úhlů, které svírají spáry, nebo poloha dveří vůči okrajům fotografie.

2.2 Lokální změny

Lokální změny jsou zpravidla ty, které se snažíme najít. Tedy takové, ke kterým dojde například pohybem předmětu, objevením se nového předmětu, či zmizením předmětu. Bohužel mezi ně částečně patří i změny osvětlení, protože způsobují posun odrazů na lesklých plochách. Právě vlivu odlesků na vyhodnocení se zbavuje nejhůře.

Na obrázku 2.1 je objevení se objektu hledanou lokální změnou, stín objektu nebo změna odrazu na dveřích jsou nechtěné lokální změny.



Obrázek 2.1: Znázornění globální změny (modrá), hledané lokální změny (zelená) a nechtěné lokální změny (červená)

3. Registrace obrazu a modely pohybu

Registrace obrazu je způsob, jak odstranit globální změny v obraze. Slouží k potlačení pohybu snímače, či změně typu snímače. To dělá tak, že hledá předpis, podle kterého přiřazuje pixely jednoho obrazu pixelům druhého obrazu, (například pokud budeme snímat jedno místo ze stejného bodu přesně stejným směrem, ale jednou použijeme fotoaparát s nižším rozlišením a jednou s vyšším, registrace přiřadí jednomu pixelu menší fotografie více pixelů fotografie větší). Pro registraci používáme metody (modely), které odpovídají očekávané globální změně. Ty budou podrobněji rozvedeny níže.

Způsob registrace obrazu i model pohybu volíme dle zamýšlené aplikace.

3.1 Registrace obrazu

Podle (BROWN 1992) může být registrace definována jako tzv. mapování mezi dvěma obrazy, a to s ohledem na prostorové umístění i intenzitu jednotlivých pixelů. Zmíněné obrazy můžeme definovat jako matice. Černobílé obrazy jako dvourozměrné matice, barevné obrazy jako trojrozměrné matice, nebo jako upravené dvourozměrné matice. Uvedené vztahy budou platné pro dvourozměrné matice. Pokud označíme tyto obrazy jako matice $O_1(x,y)$ a $O_2=(x,y)$, pak registrace mezi nimi může být obecně zapsána následovně:

$$O_2(x,y) = g(O_1(f(x,y))) \tag{3.1}$$

kde g je funkce měnící hodnoty O_1 podle jednorozměrné matice, symbolizující změnu intenzity a f je funkce, která prostorově upravuje souřadnice O_1, aby se shodovaly se souřadnicemi O_2:

$$(x', y') = f(x,y) \tag{3.2}$$

Předmětem registrace je najít vhodnou transformaci intenzity nebo prostoru tak, aby mohly obrazy být později porovnány například právě k nalezení rozdílů mezi nimi. Často není transformace intenzity potřebná, protože používáme stejný senzor za stejných světelných podmínek. Její použití může být potřeba při výrazné změněně úhlu pohledu nebo při změně polohy snímaného objektu vůči zdroji světla. Transformace intenzity pro tyto případy může být obtížná, protože většinou není stejná pro celý obraz. Téměř stejná by mohla být například při porovnávání snímků pořízených obyčejnou kamerou v osvětlené místnosti a kamerou s nočním viděním (mimo zachycených zdrojů světla).

Při řešení problému registrace tedy hledáme vhodné parametry pro prostorovou (geometrickou) transformaci. Pro implementaci je často jednodušší parametrický zápis:

$$O_2(x,y) = O_1(f_x(x,y), f_y(x,y)) \tag{3.3}$$

3.2 Způsoby registrace obrazu

Níže budou přiblíženy nejčastější metody používané k registraci obrazu dle (BROWN 1992).

3.2.1 Korelace a sekvenční metody

Korelace měří shodnost či podobnost matic. Zpravidla se používá při hledání vzorců či vzorů. Výstupem je, jak moc se testovaný obraz shoduje se vzorem či hledaným vzorcem. Tedy korelace samotná není registrační metodou, je ovšem důležitou součástí registračních metod a algoritmů. Pokud označíme vzor jako V a obraz jako O, pak normalizovaná korelace měřící jejich shodnost bude:

$$K(u,v) = \frac{\sum_x \sum_y V(x,y) O(x-u, y-v)}{\sqrt{[\ \sum_x \sum_y O^2(x-u, y-v)\ ]}} \tag{3.4}$$

Pokud se obraz prostorově shoduje se vzorem v poloze (*i,j*), pak bude mít korelace maximum právě v tomto bodě. Tím, že je korelace normalizovaná, neovlivňuje ji změna v intenzitě.

Při porovnávání obrazů tedy můžeme podle nějakého algoritmu zkoušet transformace a na základě výsledku korelace usuzovat, zda jsme získali správnou transformaci, nebo jestli se při navrhování transformací ubíráme správným směrem. Toho právě využívá například sekvenční algoritmus detekce shodnosti navržený Barneou a Silvermanem v roce 1972.

Navrhli měření shodnosti s nižší výpočetní náročností, které má normalizovaný tvar:

$$E(u,v) = \sum_x \sum_y \left| (V(x,y) - V') - [O(x-u, y-v) - O'(u,v)] \right| \tag{3.5}$$

kde *V'* a *O'* jsou střední hodnoty intenzity vzoru a obrazu. Druhým vylepšením je sekvenční strategie vyhledávání. Jedním z nejjednodušších případů, je translační registrace pomocí hraniční hodnoty. Vzor se vůči obrazu posouvá. Obraz se rozdělí na podoblasti, pak se postupně prochází podoblasti a počítá se shodnost. Shodnosti se sčítají, než dosáhnou zadané hraniční hodnoty. Poté je zaznamenán počet podoblastí, které stihl program projít, než dosáhl hraniční hodnoty, posune vzor a počítá znovu. Za nejlepší výsledek pak považujeme ten, který dosáhl hraniční hodnoty po projití největšího počtu polí.

Z výše uvedeného je tedy zřejmé, že jde o dobré metody, pokud se obraz příliš neliší od vzoru a pokud nedošlo ke změně ve větším počtu os volnosti. Pokud totiž ano, dojde k výraznému nárůstu výpočetního času. Při navrhování aplikací pak zpravidla chceme, aby pracovaly co nejrychleji. Také si tyto metody nedokáží příliš dobře poradit s neshodami v obrazech, které chceme ve výsledku zvýraznit, stejně tak pokud jsou obraz a vzor výrazně odlišné.

3.2.2 Fourierovy metody

Tyto metody využívají vlastností Fourierovy transformace. Ve Fourierově transformaci mají totiž své protějšky translace, rotace, četnost výskytu i měřítko. Od výše uvedených metod se tyto liší tím, že hledají shodu pomocí frekvence. Tím, že Fourierovy metody pracují s frekvencemi, jsou odolné vůči frekvenčnímu šumu. Jejich nevýhodou je, že jsou použitelné pouze na obrázky, které se neshodují pouze rigidně.

Nejprve budou uvedeny základní vztahy pro Fourierovu transformaci:

$$F(\omega_x, \omega_y) = R(\omega_x, \omega_y) + i \cdot I(\omega_x, \omega_y) \tag{3.6}$$

kde $R(\omega_x, \omega_y)$ je reálná část a $I(\omega_x, \omega_y)$ je imaginární část pro každou frekvenci (ω_x, ω_y). $F(\omega_x, \omega_y)$ je Fourierovou transformací obrazu, který můžeme vyjádřit jako $f(x, y)$, a i je imaginární jednotka ($i = \sqrt{(-1)}$). Exponenciální způsob zápisu rovnice (3.6):

$$F(\omega_x, \omega_y) = \left| F(\omega_x, \omega_y) \right| \cdot e^{i\varphi(\omega_x, \omega_y)} \tag{3.7}$$

Kde $|F(\omega_x, \omega_y)|$ je amplituda Fourierovy transformace a $\varphi(\omega_x, \omega_y)$ je fáze. Amplitudu v rovnici (3.7) lze spočítat z rovnice (3.6) následovně:

$$\left| F(\omega_x, \omega_y) \right| = \sqrt{R^2(\omega_x, \omega_y) + I^2(\omega_x, \omega_y)} \tag{3.8}$$

Fáze ve vzorci (3.7) se pak spočte ze vzorce (3.6) takto:

$$\varphi(\omega_x, \omega_y) = \arctan \frac{I(\omega_x, \omega_y)}{R(\omega_x, \omega_y)} \tag{3.9}$$

Právě na znalosti fáze je založena metoda fázové korelace. Pokud označíme obrazy jako funkce (x, y) a posun obrazů jako (d_x, d_y), pak můžeme vztah mezi nimi zapsat jako:

$$f_2(x, y) = f_1(x - d_x, y - d_y) \tag{3.10}$$

Odpovídající tvar ve Fourierově transformaci pak vypadá takto:

$$F_2(\omega_x, \omega_y) = F_1(\omega_x, \omega_y) \cdot e^{-i(\omega_x d_x + \omega_y d_y)} \tag{3.11}$$

Tedy oba obrazy mají ve Fourierově transformaci stejnou amplitudu a liší se pouze o fázový posun, který je závislý právě na posunu mezi obrazy. Ukazuje se, že pokud dosadíme do rovnice (3.12), lze inverzní funkcí k Fourierově transformaci získat hledaný posun pro registraci obrazů.

$$\frac{F_1(\omega_x, \omega_y) \cdot F_2^*(\omega_x, \omega_y)}{\left| F_1(\omega_x, \omega_y) \cdot F_2^*(\omega_x, \omega_y) \right|} = e^{(\omega_x d_x + \omega_y d_y)} \tag{3.12}$$

$F_2^*(\omega_x, \omega_y)$ je komplexně sdružené s $F_2(\omega_x, \omega_y)$.
Fourierova metoda hledá maximum v inverzní Fourierově transformaci. Jelikož fázový posun ovlivňuje všechny frekvence stejně, je metoda odolná vůči šumu v omezeném

pásmu. Také je odolná vůči změně osvětlení či změně snímače právě proto, že využívá fázového posunu.

Pokud bude ovšem šum přítomen napříč všemi frekvencemi, Fourierova transformace ho neodstraní a bude způsobovat chyby, protože bude zašuměno velké množství frekvencí. Tomuto se lze zčásti vyhnout, pokud známe přibližně frekvence nebo frekvenční rozsah, ve kterém se rušení vyskytuje. Pak můžeme těmto frekvencím při výpočtech přisoudit menší váhu, nebo je úplně zanedbat.

K výše uvedené metodě bylo navrhnuto zlepšení (De Castro a Morandi [1987]). Metoda byla doplněna tak, aby dokázala zpracovat translační a rotační pohyb zároveň. Zpracování translačního pohybu je uvedeno výše. Rotační pohyb se zpracovává obdobně, převedením do polárních souřadnic. Kombinace translace a rotace však představuje komplikaci. De Castro a Morandi navrhli dvoukrokový postup. V prvním kroku je zjištěn úhel, o který jsou obrazy otočeny a ve druhém kroku je nalezen posuv. Jelikož úhel neznáme, zvolíme iterační metodu, která bude odhadovat úhel. Použijeme následující vzorec, využívající polární souřadnice:

$$G(r,\theta;\varphi)=\frac{F_1(r,\theta)\cdot F_2^{*}(r,\theta-\varphi)}{\left|F_1(r,\theta)\cdot F_2^{*}(r,\theta-\varphi)\right|} \tag{3.13}$$

Podle této metody budou prověřovány úhly φ tak dlouho, dokud funkce G nenabyde tvaru odpovídajícímu pouhé translaci. Pak budeme považovat daný úhel φ za úhel natočení a pro další registraci už budeme pokračovat jako při pouhé translaci. Často je třeba použít metody interpolace, protože program dokáže procházet úhly jen v diskrétních hodnotách. Proto pokud nalezne například dva úhly v těsné blízkosti, které odpovídají tvaru pro translaci, interpoluje je a použije výsledek k určení translace.

Toto vylepšení rozšiřuje použitelnost Fourierových metod, problémem ovšem je, že obrazy mohou být pouze translačně nebo rotačně odlišné a změna mezi nimi nesmí být příliš velká.

3.2.3 Bodové mapování

Tato metoda je nejvhodnější pro srovnávání obrazů, u nichž je typ změny neznámý. Obzvlášť užitečná je v případech, kdy dojde ke změně úhlu pohledu. V takových případech nemůžeme určit změnu perspektivy, protože bez dalších senzorů není známá hloubka pole.

Doposud byly uvedeny pouze globální metody. Pokud je prostor zachycený ve snímcích hodně členitý, globální metody kvůli nesrovnalostem v perspektivě nefungují. V takovém případě je lepší použít lokální bodové mapování, které zdokonaluje globální metodu a umožňuje jí zpracovávat členitější záběry.

Proces bodového mapování se skládá ze tří fází. V první fázi program nalezne význačné body (tzv. kontrolní body) v obrazu i ve vzoru. Ve druhé fázi páruje kontrolní body obrazu a vzoru. Ve třetí fázi nadefinuje předpisy pro přepočet souřadnic mezi kontrolními body. Do výsledné registrační podoby se výstup přepočítá pomocí interpolací.

Další výhodou bodového mapování je možnost zavedení zpětné vazby, která umožňuje zpřesnění registrace. Například jsou-li v obraze kontrolní body blízko sebe a ve vzoru nějaký z nich chybí, (například je v zákrytu), program zkusí provést více způsobů spárování a pomocí zpětné vazby zjistí, který způsob byl nejlepší.

3.3 Modely pohybu (Motion models)

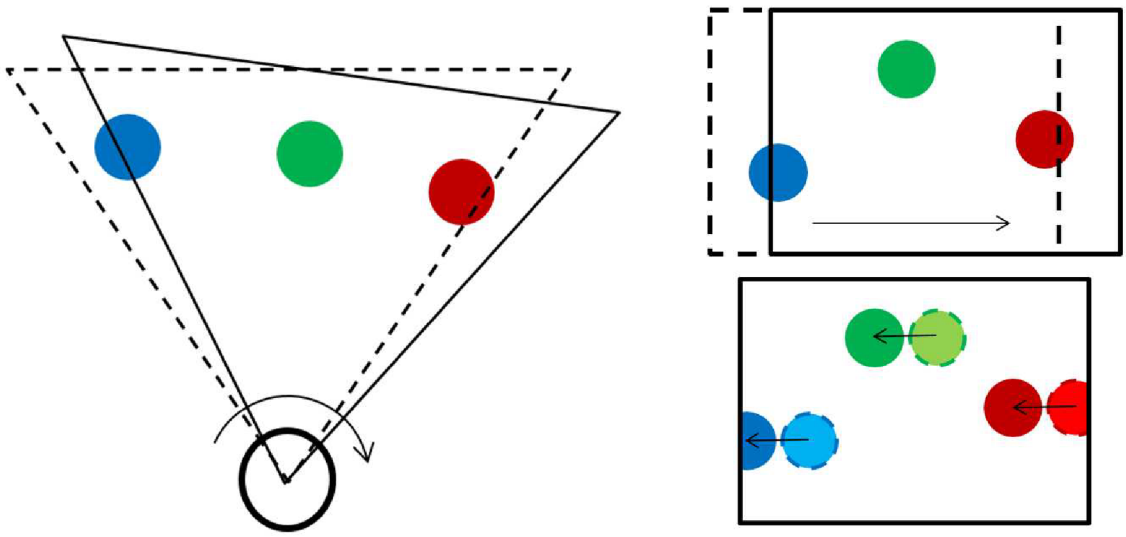
V následující části budou uvedeny pohybové modely užívané knihovnami OpenCV. Tyto pohybové modely udávají typ pohybu snímače a tedy pohyb scény, potažmo pixelů scény. Na obrázcích uvedených ke znázornění pohybových modelů je vždy vlevo znázorněn pohyb čidla, vpravo nahoře ukázka změny záběru a vpravo dole je ukázáno, jak se posunuly objekty v záběru.

3.3.1 Posuvný pohybový model (Translation motion model)

Translace záběru patří mezi nejzákladnější globální změny (spolu s rotací), je proto nejjednodušší ji potlačit. K translaci záběru dojde rotací snímače okolo osy kolmé k podložce, rotací okolo osy rovnoběžné s podložkou nebo jejich kombinací. Detekce této změny je vhodná například pro vytváření panoramatických fotografií, kdy je nalezena shoda mezi částmi dvou snímků po pootočení snímače, a pomocí shody jsou snímky spojeny. Funguje lépe při zabírání více vzdálených objektů při nepříliš velkém kroku otáčení. Čím blíže k okraji fotografie, tím více se projevují chyby způsobené čočkou. Snímáme-li vzdálenější objekty, jsou zachovány úhly a vzdálenosti mezi nimi. Na bližší vzdálenost nebo při kombinaci blízkých a vzdálených objektů začne mít na obraz vliv perspektiva, se kterou si snímač neporadí (na rozdíl od lidského oka), a je tedy nutné zvolit jinou metodu.

Pokud se budeme zajímat o využitelnost v monitorování prostoru, je možné použít tuto metodu na bezpečnostní kamery, které se pouze otáčejí okolo nějaké své osy.

Model je znázorněn na obrázku 3.1, kde je ukázán pohyb čidla, pohyb záběru a nakonec pohyb scény v záběru.



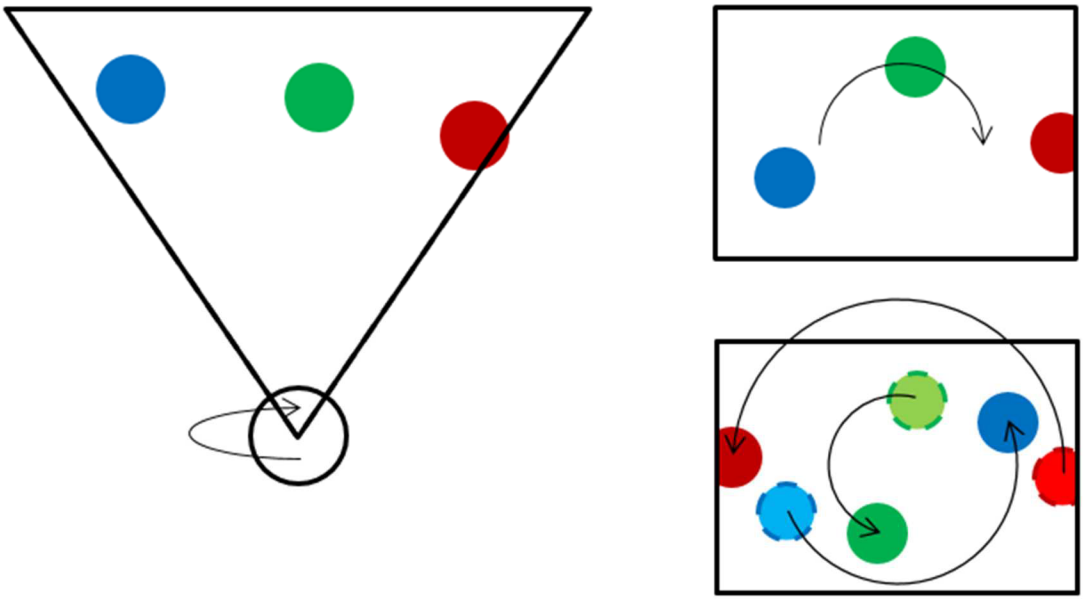
Obrázek 3.1: Znázornění posuvného modelu

3.3.2 Pohybový model natočení (Rotation motion model)

K rotaci záběru teoreticky dochází otočením snímače kolem optické osy. O opravdovém samostatném otočení však můžeme mluvit pouze v případě, pokud otočíme snímač přesně o 180°. Důvodem je, že drtivá většina (ne-li všechny) čipů pro zachycení obrazu má obdélníkový tvar. Tedy při rotaci snímače o 180° se na čip opět zobrazuje stejná oblast, jen otočená o 180°. Ve všech ostatních případech se na část čipu promítá původní záběr, ale na část se promítá ta část, která se do původního obdélníku nevešla. Registrace v takových případech je možná, využitelná obzvlášť pokud sledujeme předmět uprostřed snímku.

Z hlediska monitorování prostoru je uvažování pouhé rotace téměř nepoužitelné.

Model natočení je znázorněn v obrázku 3.2.



Obrázek 3.2: Znázornění modelu natočení

3.3.3 Eukleidovský pohybový model (Euclidean motion model)

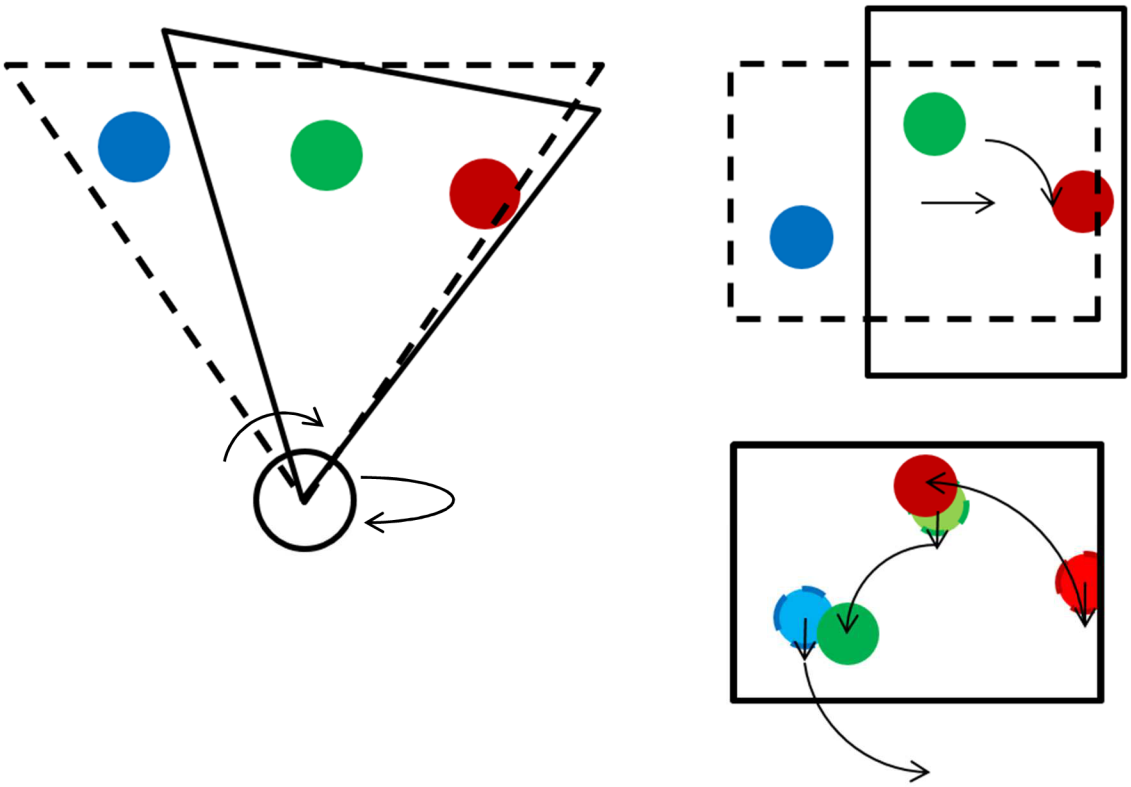
Tento model využívá obou výše uvedených modelů. Jde tedy o případ, kdy se čidlo pohnulo ve všech svých osách, ale stále je umístěno ve stejném bodě. Tento model je tedy dokonalejší než předchozí dva zvlášť, protože na rozdíl od translačního počítá s natočením a na rozdíl od rotačního počítá s tím, že se mohl pohnout střed fotografie.

Užití Eukleidovské transformace je vhodné například, chceme-li porovnat snímky pořízené fotoaparátem ze stativu, pokud si nejsme jistí jeho stabilitou. Tedy že se při opětovném stisku spouště mohl snímač pohnout (směr stisku spouště zpravidla neprotíná optickou osu a upevnění ve stativu nemusí být spolehlivé).

Případem, kdy je možno využít Eukleidovského modelu je oprava fotografie, ve které se „rozjely“ jednotlivé barevné složky. Čidlo je tedy ve stejném bodě, ale například chybou čočky jsou obrazy v jednotlivých barevných složkách vzájemně posunuté. Nalezneme-li změnu polohy pixelů mezi barevnými složkami navzájem, můžeme složky sesadit zpět k sobě a tedy opravit snímek.

Registrace podle Eukleidovského modelu je vhodná k monitorování statickou kamerou, nebo kamerou, která se otáčí okolo nějaké své osy, máme-li podezření, že se může vlivem počasí či provozu v hlídaném prostoru rozkmitat.

Znázornění Eukleidovského modelu je na obrázku 3.3.



Obrázek 3.3: Znázornění Eukleidovského modelu

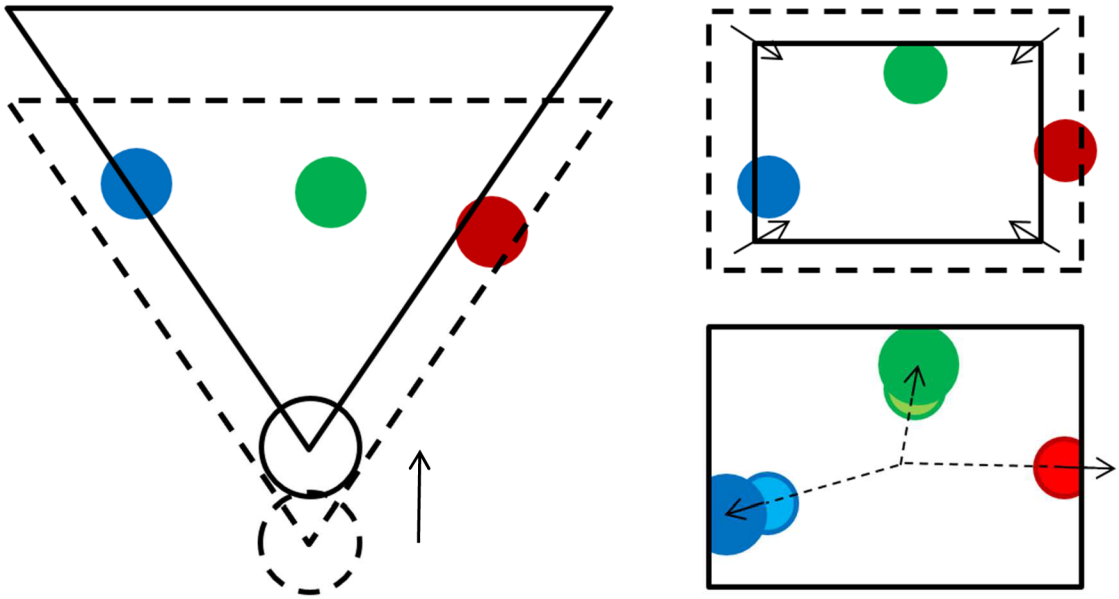
3.3.4 Afinní pohybový model (Affine motion model)

Výše uvedené modely uvažují pevně umístěný snímač, který se může pohybovat pouze okolo svých os. Pro zpracování obrazu z robota, který se pohybuje sledovaným prostorem, musíme uvažovat i jeho pohyb. Bez vodicí linie (například černé čáry na podlaze) robot nikdy neprojede dvakrát přesně stejným místem.

Afinní model předpokládá zachování rovnoběžnosti stejně jako Eukleidovský, ale na rozdíl od Eukleidovského počítá se změnou úhlů. Tedy je schopný zpracovat případy, kdy se obraz posunul, natočil a ještě k tomu zploštil nějakým směrem. K tomu může dojít například natočením snímaného objektu. Dále počítá i se zoomem, což je zploštění ve dvou vzájemně kolmých směrech o stejnou procentuální hodnotu.

Hledání afinní transformace je již výpočetně náročnější a kvůli předpokladu zachování rovnoběžnosti hran není vhodná pro monitorování interiéru.

Afinní model je znázorněn na obrázku 3.4.



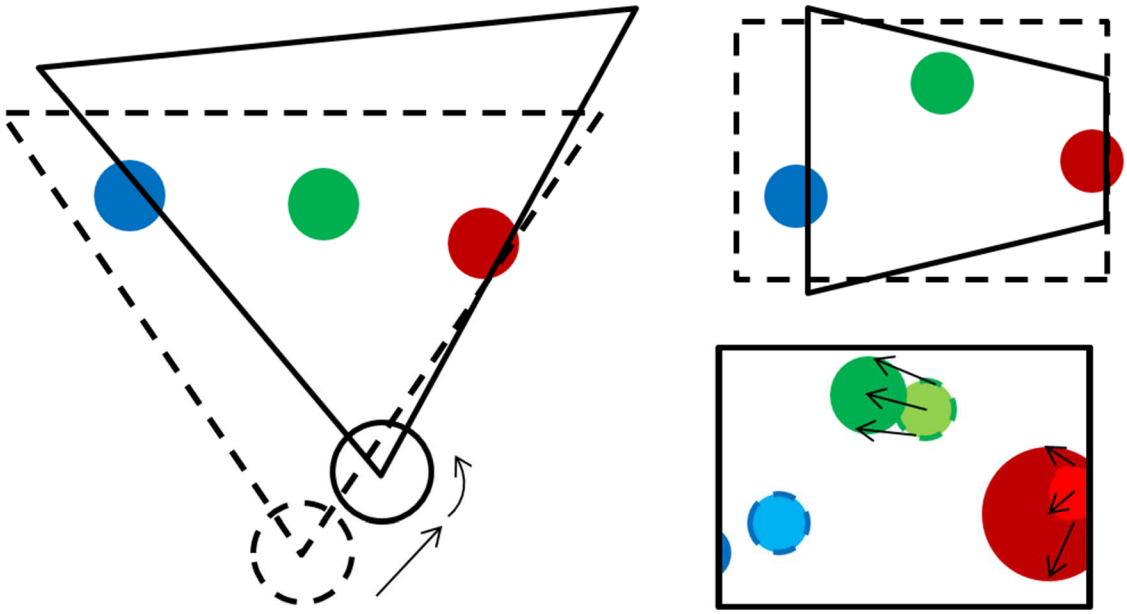
Obrázek 3.4: Znázornění afinního modelu (případ „zoom“)

3.3.5 Homografický pohybový model (Homography motion model)

Tento model na rozdíl od výše zmíněných počítá s perspektivou. Proto je vhodný do interiérů. Uvnitř budov jsou všechny vzdálenosti poměrně malé a jakýkoliv pohyb robota mění v obraze úhly, neplatí rovnoběžnost, významně se poměrově mění obsahy obrazů snímaných ploch. V některých případech může plocha kvůli změně pozice robota úplně zmizet.

Z hlediska spolehlivosti je tento model určitě nejlepší pro monitorování prostoru. Je ovšem velmi výpočetně náročný. Čím větší změnu připustíme, tím víc stoupá výpočetní náročnost.

Na obrázku 3.5 je přiblížen homografický pohybový model.



Obrázek 3.5: Znázornění homografického modelu

4. Shrnutí rešerše a určení cílů

4.1 Shrnutí rešerše

Účelem navrhovaného programu je zpracovávat obraz z robota, který bude projíždět střežené území. Je tedy potřeba využít registraci. Je zde předpoklad, že robot neprojede vždy stejnou trasu a zaznamenané snímky mohou být pohybem lehce rozmazané. To vše vyžaduje započítání perspektivy. Rozhodně tedy bude třeba použít homografický pohybový model.

Pokud by byl k dispozici velký výpočetní výkon a nebyl by požadavek na rychlost výpočtu, byla by vhodnou metodou metoda bodového mapování se zpětnou vazbou. S přihlédnutím k výpočetní náročnosti a faktu, že knihovny OpenCV poskytují zabudovanou metodu registrace založenou na korelaci a sekvenčních metodách, rozhodl jsem se využít ji.

4.2 Určení cílů

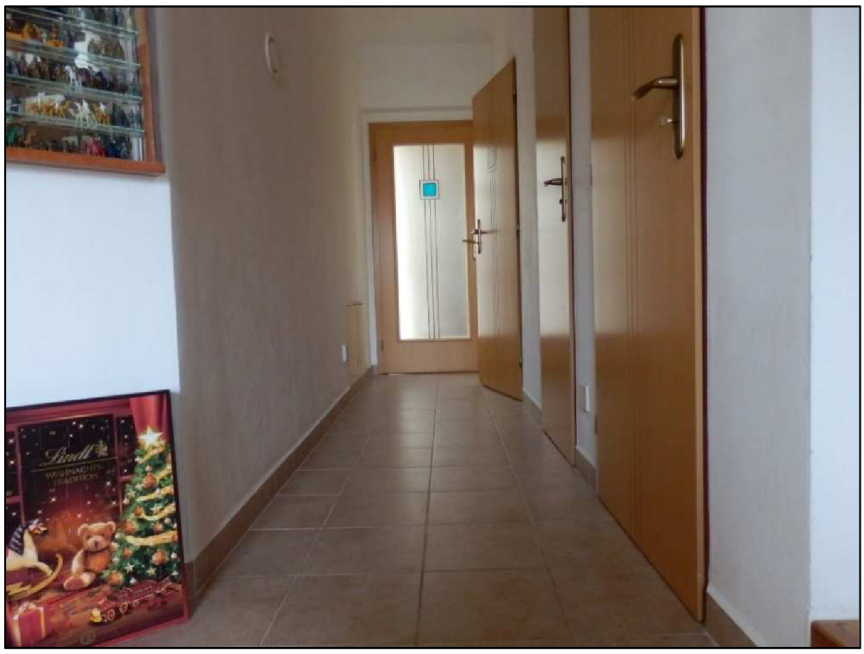
Prvním krokem bude implementování výše uvedených poznatků, tedy vytvoření programu, který bude využívat homografický pohybový model. Ideální by bylo požití bodového mapování, použiji ovšem přednastavenou metodu knihovny OpenCV.

Dále se zaměřím na snížení výpočetního času při zachování co nejlepších výsledků.

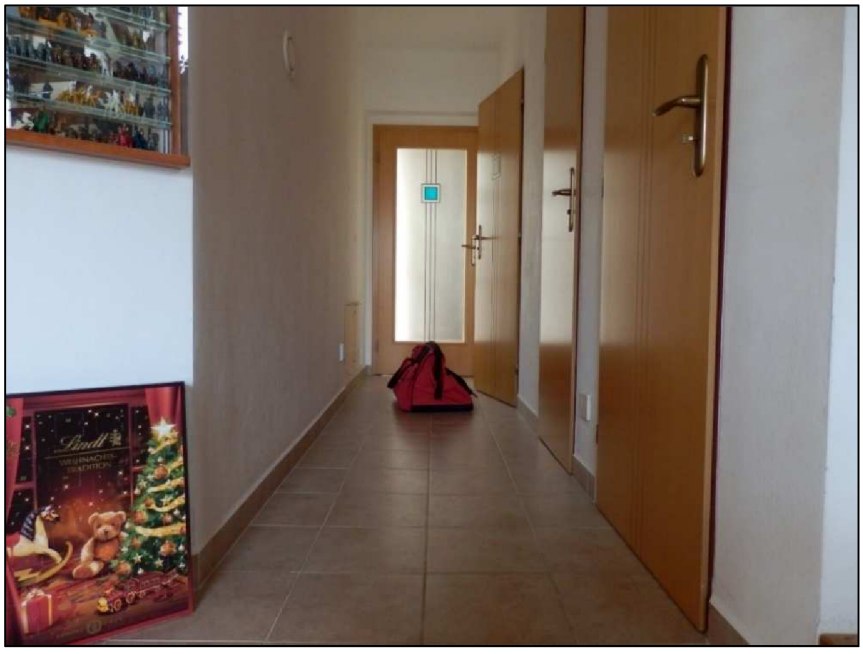
5. Detekce změny prostředí

5.1 Provedení

Na základě výše uvedeného a možností knihoven OpenCV jsem navrhnul program využívající tzv. „Image Registration using Enhanced Correlation Coefficient Maximization," (registrace obrazu využívající maximalizace vylepšeného koeficientu korelace). Program načítá dva obrazy, jeden z nich považuje za vzor, ten se právě transformuje a porovnává s druhým obrazem. Na obrázcích 5.1 a 5.2 jsou ukázány porovnávané fotografie:



Obrázek 5.1: Fotografie považovaná za vzor



Obrázek 5.2: Fotografie, ve které bude hledána odlišnost

Jsou tedy nahrány barevné obrazy. OpenCV pracuje s barevnými obrazy jako se zvláštní dvojrozměrnou maticí. Každý výsledný pixel se skládá ze tří složek (zelená, červená a modrá). Matice barevného obrazu podle OpenCV má stejnou výšku jako je počet pixelů v obrazu, ale šířku trojnásobnou, protože se složky rozkládají právě do šířky.

Je zřejmé, že čím větší budou matice, tím větší objem dat musí program zpracovávat. Proto je obraz i vzor převeden do černobílé, čímž se celé matice třikrát zúží. Nyní je v každém bodě matic pouze hodnota intenzity. Matice jsou ale stále velké.

Rozhodl jsem se tedy obraz i vzor zmenšit, a to ze zdrojových 2048x1536 pixelů na 320x240 pixelů (viz obrázky 5.3, 5.4). Tím dojde k výraznému urychlení výpočtu, ale také ztrátě detailů. Ztráta detailů by mohla být vnímána jako negativní jev, ovšem v tomto případě by změna ve ztraceném detailu zanikla v šumu vzniklém posunutím vůči zdroji osvětlení. V obrázku 5.5 je ukázán rozdíl vzoru a obrazu bez registrace. Tento i všechny další rozdíly budou uváděny v dvojhodnotovém tvaru (booleanovské hodnoty). Tento tvar získáme srovnáním rozdílu s nějakou hraniční hodnotou.

Obrázek 5.3: Zmenšený vzor převedený do černobílé verze

Obrázek 5.4: Zmenšený srovnávaný obraz převedený do černobílé verze

Obrázek 5.5: Rozdíl obrazu a vzoru bez registrace

V obrázku 5.5 je vidět, že změna úhlu záběru způsobuje výrazný rozdíl obzvlášť v okolí hran. Proto je vzor zpracován pomocí registrace. Konkrétně je použita registrace s homografickým pohybovým modelem. Rozdíl obrazu a vzoru po homografické registraci je zobrazen v obrázku 5.7.

Obrázek 5.6: Homograficky registrovaný vzor

Obrázek 5.7: Rozdíl obrazu a vzoru po homografické registraci

Nyní lze pozorovat černé oblasti na okrajích registrovaného vzoru, viz obrázek 5.6. Ty by zanášely do vyhodnocovacího procesu chybu. Experimentálně jsem zjistil, že tato oblast může být až 10% rozměru fotografie. Proto jsem obraz i registrovaný vzor z každé strany o 10% oříznul. V rozdílu je také možno vidět, že se stále nedokonale sesadily hrany. Pro odstranění tohoto problému jsem registrovaný vzor ještě jednou registroval, tentokrát afinně, jelikož perspektiva je již z větší části napravena první homografickou registrací. Výsledek je vidět na obrázku 5.8.

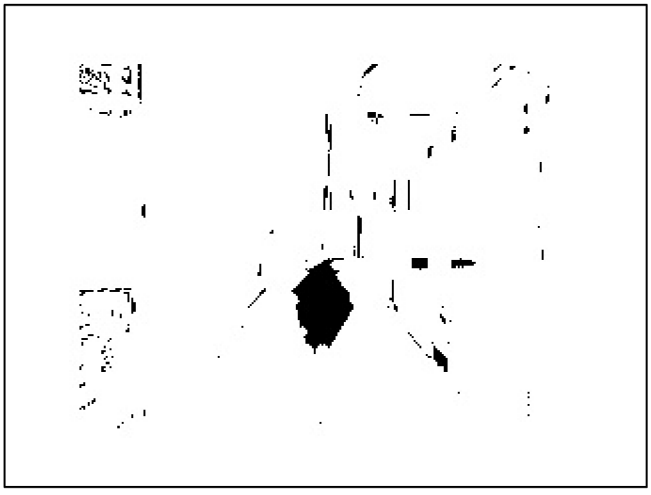
Obrázek 5.8: Afinně registrovaný vzor

Obrázek 5.9: Rozdíl obrazu a vzoru po afinní registraci

Na okrajích rozdílu (obrázek 5.9) jsou opět vidět černé oblasti. Proto jsem rozdíl opět oříznul, tentokrát pouze o 5 pixelů z každé strany. Pro další zpracování je třeba rozdíl doplnit o všechno, co bylo oříznuto. Výsledek je vidět v obrázku 5.10.

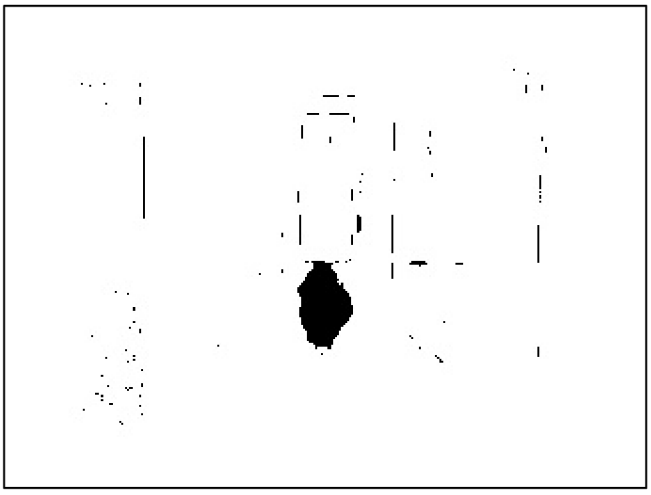
Obrázek 5.10: Rozdíl po doplnění oříznutých okrajů

Tento rozdíl jsem už použil jako masku pro zvýraznění změny v obraze. Je vidět, že i přes dvojnásobnou registraci je maska značně zašuměná, proto jsem se rozhodl masku vyčistit pomocí detekce rovných linií. Většina rovných linií v masce vzniká na hranách, které se nedokonale sesadily. Existuje určitá pravděpodobnost, že by jako rovná linie mohla být identifikována i hledaná změna, je ale pravděpodobnější, že hledaná změna zabírá větší plochu než jedinou rovnou linii. Výsledek odstranění těchto linií je na obrázku 5.11.



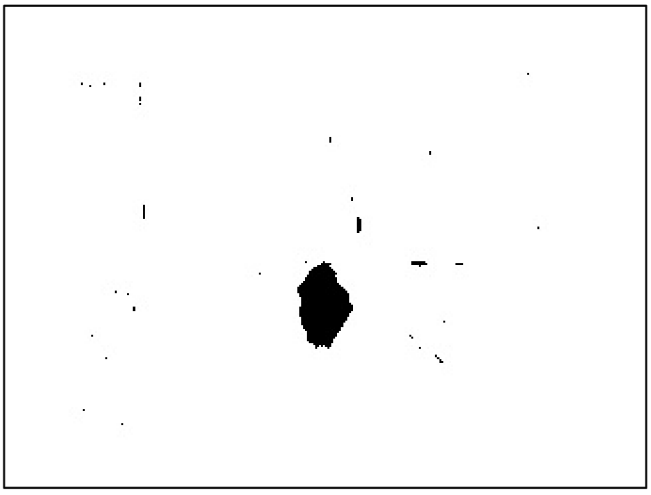
Obrázek 5.11: Maska po odstranění zjištěných rovných linií

Druhou metodou, kterou jsem použil pro odstranění šumu, byla detekce okrajů. Program zaznamenává přechody černá/bílá a určí je jako okraje. Okraje pak vybělí. Tímto procesem sice dojde ke ztrátě části hledaného obrysu, ale hledaný objekt by měl být výrazně větší než šum, aby byl výsledek průkazný, proto si tuto operaci mohu dovolit. Výsledek je vidět na obrázku 5.12.



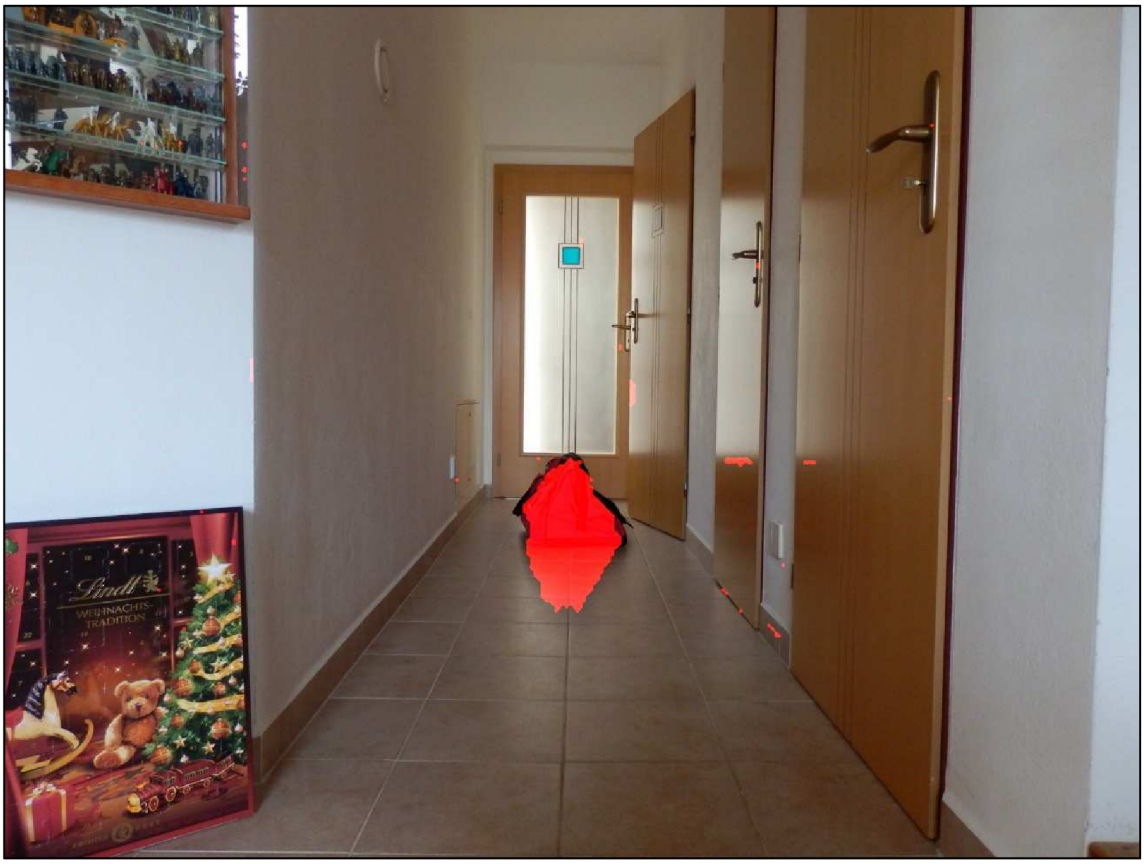
Obrázek 5.12: Maska po odstranění zjištěných okrajů

Obě masky (obrázky 5.11, 5.12) byly vypočítány z masky na obrázku 5.10. Abych se co nejlépe zbavil šumu, vynásobil jsem upravené masky mezi sebou. Bylo třeba je nejprve invertovat, pak vynásobit a pak invertovat zpět, protože bílá má hodnotu 255 a černá hodnotu 0. Výsledek této operace je na obrázku 5.13.



Obrázek 5.13: Maska po zkombinování metod pro odstranění šumu

V masce stále zůstal šum, ale už je značně redukovaný. Tuto masku jsem nakonec zvětšil zpět na původní velikost obrazu a zvýraznil v obrazu červeně ty oblasti, kde je maska černá. Výsledek detekce změny je ukázán na obrázku 5.14. Na něm je vidět, že šum zůstal na místech, která byla zmíněna v rešerši jako obtížná na odšumění. Tedy u lokálních změn osvětlení (na hraně odrazu) na povrchu, který dobře odráží světlo.



Obrázek 5.14: Výstup – obraz se zvýrazněnou změnou pomocí masky

5.2 Shrnutí

Porovnávané fotografie byly nahrány, převedeny z barevného provedení do černobílého a zmenšeny funkcí *resize*, která používá ke změně rozměrů obrazů interpolaci. Konkrétně byla použita metoda *INTER_LANCZOS4*. Ta používá k výpočtu pole sousedních pixelů o rozměrech 8x8 pixelů. Došlo ke zmenšení z 2048x1536 pixelů na 320x240 pixelů.

Poté byly obrazy registrovány, nejprve podle homografického modelu, aby byl potlačen vliv perspektivy, následně podle afinního modelu pro zlepšení přesnosti registrace. Z registrovaných obrazů byla vytvořena maska ukazující rozdíl mezi obrazy. Z masky byly pomocí detekce okrajů a detekce úseček odstraněny rušivé vlivy, tím dochází ke ztrátě detailů, ty by ale byly ztracené v šumu, tedy by výsledky nebyly průkazné.

Nakonec byla maska zvětšena zpět na velikost původních fotografií a s její pomocí bylo vše, co program vyhodnotil jako změnu zvýrazněno červeně.

Program zvládne do jisté míry porovnávat fotografie pořízené z různých míst. V ukázaném příkladu byl fotoaparát posunut do strany, dopředu, natočen okolo optické osy a pootočen okolo osy kolmé k zemi, čímž bylo dosaženo téměř nejnepříznivější změny, jejíž zpracování by mohlo být po programu požadováno. V daném případě se program ukázal být funkčním.

Mohlo by dojít k problému, pokud by byly snímky pořízeny za různých světelných podmínek. V takovém případě by registrace měla proběhnout, výpočet masky by ovšem proběhl chybně. S poklesem osvětlení by plošně klesla intenzita pixelů a to by ovlivnilo vyhodnocení, protože se maska počítá z rozdílu fotografií.

Tabulka 5.1: Výpočetní časy jednotlivých procesů

Proces	**t[ms]**	**[%]**
Nahrávání fotografií	280	13.9
Zmenšování fotografií	436	21.6
Převod na černobílé verze	1	0.0
Homografická registrace	495	24.5
Ořezávání fotografií	1	0.0
Afinní registrace	424	21.0
Vytváření masky	97	4.8
Vytváření výstupu	286	14.2
SUMA	2020	

Celý výpočet trvá přibližně 2 sekundy, jak je vidět v tabulce 5.1. Z toho vždy zhruba 20 - 25% zabírá každá registrace. Kdybych fotografie nezmenšil, zabraly by registrace více času. Ostatní procesy nejsou tolik ovlivněny velikostí fotografií. Procentuální podíl registrací by pak samozřejmě vzrostl. Kvůli zachování určité kvality výsledku už si nemohu ani dovolit více zmenšit fotografie. Výpočet je v tomto stavu zároveň rychlý a poměrně přesný.

K napsání programu jsem použil jazyk C++ a rozhraní Visual Studio 2015.

6. Závěr

Vypracoval jsem rešerši pro získání přehledu o metodách vhodných ke zpracování obrazu. Na základě této rešerše a dostupných prostředcích jsem si stanovil postup návrhu programu. Navrhnul jsem program, který plní zadanou funkci. Jeden uvedený výpočet trvá přibližně 2 sekundy.

Výstup programu by samozřejmě mohl být přesnější, pokud bych použil vhodnější metodu a pracoval s neredukovanými barevnými fotografiemi, ale výrazně by stoupnula výpočetní náročnost a tedy výpočetní čas. Dosažený výpočetní čas je dobrý. Z tabulky 5.1 je vidět, že podíl jednotlivých procesů už neumožňuje příliš velké zlepšení při zachování kvality výstupu.

Všechny cíle práce byly splněny.

7. Seznam použitých zdrojů

BROWN, Lisa. 1992. A SURVEY OF IMAGE REGISTRATION TECHNIQUES. . s. 325-376.

Image Alignment (ECC) in OpenCV (C++ / Python) [online]. 2016. [cit. 2016]. Dostupné z: http://www.learnopencv.com/image-alignment-ecc-in-opencv-c-python/

OpenCV 2.4 Documentation [online]. 2016. [cit. 2016]. Dostupné z: http://docs.opencv.org/2.4/

OpenCV 3.0.0 Documentation [online]. 2014. [cit. 2016]. Dostupné z: http://docs.opencv.org/3.0-beta/index.html

RADKE, RJ, S ANDRA a O AL-KOFAHI. 2005. Image change detection algorithms: A systematic survey. . Roč. 14, č. 3, s. 294-307.

8. Seznam obrázků

9. Seznam příloh

Tabulka 9.1: Obsah přiloženého CD

bakalarska_prace\	Složka se zdrojovými kódy pro oba programy. (viz níže) (Projekt prostředí Visual Studio 2015)
output\	Složka s výstupními soubory programu *bakalarska_prace.exe* pro přiložené fotografie. (viz níže)
2016_BP_Venglar_Vojtech_161917.pdf	PDF verze bakalářské práce.
bakalarska_prace.exe	Porovnávací program s průběžným výpisem na obrazovku. Tento program vytváří průběžně soubory.
bakalarska_prace_cas.exe	Porovnávací program upravený k přesnějšímu měření časů jednotlivých operací (bez průběžného výpisu na obrazovku). Tento program vytváří pouze výstupní soubor.
image1.jpg	Fotografie použitá výše jako obraz s hledanou odlišností.
image2.jpg	Fotografie použitá výše jako vzor pro porovnání.
readme.txt	Textový soubor s návodem na obsluhu programů.