VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2022 Jan Čech

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DETEKCE ZMĚN V OBRAZE

DETECTION OF CHANGES IN THE IMAGE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Jan Čech

AUTHOR

VEDOUCÍ PRÁCE — Ing. Miloslav Richter, Ph.D.

SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Jan Čech ***ID:*** 211416

Ročník: 3 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Detekce změn v obraze

POKYNY PRO VYPRACOVÁNÍ:

V obraze pořízeném pohybující se kamerou je nutné vyznačit změny, které se udály od posledního průchodu (pohledu kamery) stejným místem.

1) Nastudujte metody zpracování obrazu a identifikace objektů.

2) Navrhněte testovací pracoviště pro ověřování metod a nasnímejte několik sad pro testování.

3) Navrhněte a realizujte algoritmy pro extrakci přítomných objektů, zjištění shodných pozic v jednotlivých průchodech, a následné zjištění rozdílů. Zvolte vhodnou reprezentaci dat.

4) Otestujte realizované algoritmy a postupy na testovací sadě. Zhodnoťte výsledky.

DOPORUČENÁ LITERATURA:

Gonzalez R.C., Woods R.E.: Digital Image Processing, 4th edition, Pearson, 2017, ISBN 978-0133356724

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 23.5.2022

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

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

Abstrakt

Cílem této práce je nalezení změn ve dvou podobných obrazech, pořízených v různých okamžicích. Tato problematika spadá do oboru počítačového vidění. K řešení tohoto problému je použito různých metod počítačového vidění. Nejdříve je nutné zjistit úhel snímání. Následně obrázky projdou předzpracováním, pak jsou podrobeny segmentaci a určování deskriptorů. Nakonec dojde k porovnání obou obrázků a vyzvednutí rozdílů.

Klíčová slova

klíčový bod, SURF, předzpracování, detekce změn

Abstract

The aim of this thesis is finding changes in two similar pictures made in different periods of time. This task belongs to the field of computer vision. To solve this problem different methods of computer vision are used. First the angle of scanning has to be found. Then pictures go through the proces of preprocessing, afterwards the segmentation and the determination of descriptors are carried out. Finally, the pictures are compared and the differences are pointed out.

Keywords

point of interest, SURF, preprocessing, change detection

Bibliografická citace

ČECH, Jan. Detekce změn v obraze. Brno, 2021. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/130618. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Miloslav Richter.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	*Jan Čech*
VUT ID studenta:	*211416*
Typ práce:	*Bakalářská práce*
Akademický rok:	*2021/22*
Téma závěrečné práce:	*Detekce změn v obraze*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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 2022 ----------

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miloslavu Richterovi, Ph.D. za vždy přítomnou ochotu a vůli, vést mě trnitou cestou psaní bakalářské práce a za vytrvalý boj proti nevědomosti, konkrétně mé.

V Brně dne: 21. května 2022 ----------

podpis autora

Obsah

SEZNAM OBRÁZKŮ **8**

ÚVOD **9**

CÍLE PRÁCE 10

1. VYPRACOVÁNÍ **11**

1.1 PŘEDZPRACOVÁNÍ[1] 11

1.1.1 Metody *11*

1.1.2 Bodové jasové transformace *12*

1.1.3 Geometrické transformace *13*

1.1.4 Lokální předzpracování *14*

1.1.5 Filtrace šumu a poruch *16*

1.1.6 Rekonstrukce obrazu *7*

1.1.7 Matematická morfologie *7*

1.2 SEGMENTACE[2] 17

1.2.1 Metody vycházející z detekce hran (edge-based) *17*

1.2.2 Metody orientované na regiony v obraze (region-based) *17*

1.2.3 Statistické metody *17*

1.2.4 Hybridní metody *17*

1.2.5 Znalostní metody (knowledge-based) *17*

1.3 DETEKCE KLÍČOVÝCH BODŮ 18

1.3.1 SIFT (Scale-Invariant Feature Transformation)[9] *19*

1.3.2 BRISK metoda (Binary Robust Invariant Scalable Keypoints)[3] *19*

1.3.3 SURF metoda (Speeded Up Robust Features)[4,5] *20*

1.3.4 Popis *22*

1.3.5 Porovnání deskriptorů *23*

1.4 HOMOGRAFIE[7] 24

1.4.1 Projektivní lineární transformace *24*

1.4.2 Výpočet homografie *25*

2. NÁVRH ŘEŠENÍ **27**

2.1 PŘEDZPRACOVÁNÍ 27

2.2 VYPÁLENÍ BINÁRNÍ MASKY 27

2.3 SROVNÁVÁNÍ DESKRIPTORŮ 29

2.4 OŠETŘENÍ RŮZNÝCH ÚHLŮ 30

2.5 ZKOUŠENÍ RŮZNÝCH SCÉN 31

3. ZÁVĚR **35**

3.1 VHODNĚ PŘEDZPRACOVAT OBRÁZEK 35

3.2 VYHLEDAT KLÍČOVÉ BODY V OBRÁZCÍCH 35

3.3 OŠETŘIT RŮZNÉ ÚHLY SNÍMÁNÍ OBRÁZKŮ 35

3.4 POROVNAT OBRÁZKY A ZVÝRAZNIT ZMĚNY 36

LITERATURA **37**

OBRÁZKY **38**

SEZNAM OBRÁZKŮ

1.1 Příklad předzpracování 12
1.2 Příklad filtrace šumu 16
1.3 Příklad segmentace 18
1.4 Klíčové body 18
1.5 Deskriptor metody SIFT 19
1.6 Deskriptor metody BRISK 20
1.6 Druhá derivace Gaussovy funkce 21
1.7 Haarova vlnka 22
1.8 Hledání deskriptorů metodou SURF 23
1.9 Porovnávání poloh sesouhlasených deskriptorů 24
1.10 Homografie 25
1.11 Projektivní transformace 25
2.1 Snímky dodané na vstupu 27
2.2 Výstup z dat na obrázku 2.1 28
2.3 Snímky dodané na vstupu 28
2.4 Výstup z dat na obrázku 2.3 29
2.5 Výstup z dat na obrázku 2.3 30
2.6 Testování scény s nástroji 31
2.7 Testování scény s penězi 32
2.8 Testování scény s ovocem 33

Úvod

Téma mé závěrečné práce je Detekce změn v obraze. Toto téma spadá do oboru počítačového vidění. Počítačové vidění je disciplína zabývající se zpracováním obrazu.
Proces zpracování obrazu se skládá z několika částí: snímání, předzpracování, segmentace a popis.
Předzpracování slouží většinou k potlačení nežádoucích jevů, které by mohly vadit při dalším zpracování. Mezi tyto operace se řadí: filtrace šumu, restaurace obrazu, jasová transformace, případně převedení do odstínů šedé atd.
Dalším krokem je segmentace. Segmentace je proces dělení obrazu do částí, které korespondují s konkrétními objekty v obraze. Tyto části vždy nesou určitou informaci o obraze. K segmentaci lze využít bezpočet přístupů. Obrázek lze dělit pomocí detekce hran, regionů nebo na základě statistické analýzy.
V této práci je také využito snímků snímaných pod různými úhly. Je tedy potřeba sesouhlasit klíčové body obou snímků. K tomu bude využita homografie.

Cíle práce

Nejprve je potřeba nastudovat metody, které budou uplatněny v Programu, který bude produktem této práce. Také je potřeba sestavit scénu, na které by bylo možné program testovat. Program by měl: načíst obrázek, provést potřebné předzpracování, nalézt klíčové body, provést výpočet homografie, srovnat pozice shodných klíčových bodů a zvýraznit změny ve výsledném obrázku.

Mezi možné využití pro hotový program by mohla být kontrola výloh v obchodních domech, inventarizace nástrojů v dílnách či ordinacích, kompletace výrobku, kontrola rentgenových snímků.

Tyto cíle jsem tedy zformuloval do jednotlivých bodů.

1. Vhodné předzpracování obrázku
2. Ošetření různých úhlů snímání obrázků
3. Vyhledání klíčových bodů u obou obrázků
4. Porovnání obrázků a zvýraznění změn

1. VYPRACOVÁNÍ

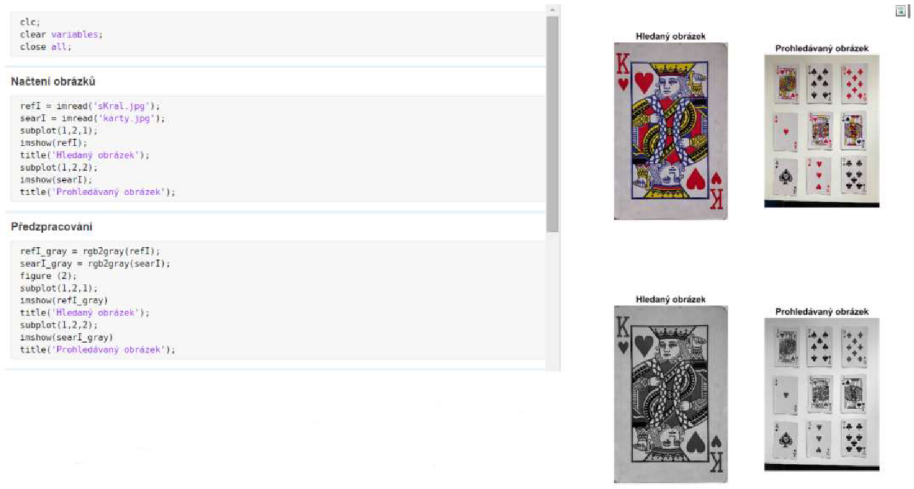
1.1 Předzpracování[1]

Předzpracování je nedílnou součástí zpracování obrazu. V této práci bude, oblasti předzpracování použit převod do černobílé, která je nutná pro další práci s obrázky a v případě nutnosti filtrace šumu, pomocí Gaussova filtru, která se stará o odstranění případného šumu. Filtrace pomocí Gaussova filtru se rovněž využívá v metodách pro detekci klíčových bodů.

Při předzpracování se využívá nadbytečnosti údajů v obrazu – sousední pixely mají většinou podobnou hodnotu jasu. Řadu operací předzpracování můžeme zjednodušit vhodným nastavením scény, výběrem senzoru nebo objektivu. Předzpracování musíme vztahovat k tomu, co chceme z obrazu získat, co s ním chceme dělat dál.

1.1.1 Metody

1. Bodové jasové transformace
   - Jasová korekce
   - Transformace jasové stupnice
2. Geometrické transformace
   - Plošná transformace
   - Jasová transformace
3. Lokální předzpracování
   - Vyhlazování obrazu
   - Detekce hran, ostření
4. Filtrace šumu
5. Rekonstrukce obrazu
6. Matematická morfologie



Obrázek 1.1 Příklad předzpracování

I převedení obrázku do černobílé může být užitečnou metodou předzpracování. Z obrázku odstraní přebytečná data a zmenší se jeho velikost, což může potenciálně zrychlit další operace.

1.1.2 Bodové jasové transformace

Jas v bodě výstupního obrazu závisí pouze na jasu bodu ve vstupním obrazu, pro úpravu jednoho konkrétního pixelu použijeme jen tento pixel vstupního obrazu.

Jasová korekce

- Je aplikována při poruchách hardwaru (systematických chybách) jako je jiná citlivost jednotlivých světlo citlivých prvků snímače (vadné pixely), nerovnoměrné osvětlení, jiná citlivost snímacího a digitalizačního zařízení.
- Při stálých světelných podmínkách pořídíme obraz o známém rozložení jasu, nejlépe obraz o konstantním jasu $c => f_c(x, y)$.

$$e(x,y) = \frac{f_c(x,y)}{c} \rightarrow f(x,y) = e(x,y) \cdot g(x,y) \qquad 1.1$$

Předpokládá se multiplikativní model poruchy e (x, y)

- Další možností je pořídit obraz s objektem I_0, obraz za stejných světelných podmínek bez objektu.

I_f – korekce osvětlení a obraz za tmy (zakrytý objektiv)

I_b – korekce nelinearity snímače

$$f(x,y) = M \cdot \frac{I_0(x,y) - I_b(x,y)}{I_f(x,y) - I_b(x,y)} \qquad 1.2$$

M je konstanta, kterou měníme kontrast výsledného obrazu

Transformace jasové stupnice

Transformace T je výchozí stupnice jasu p na novou stupnici q: $q = T(p)$. Jen určitá hodnota jasu ve vstupním obrazu je transformována na jinou hodnotu, bez ohledu na pozici, příklady: inverze, prahování, ekvalizace histogramu, roztažení histogramu, úprava kontrastu.

1.1.3 Geometrické transformace

Geometrická transformace je využívána k odstranění geometrických zkreslení (zkosení vůči snímané ploše, širokoúhlé snímače), změně rozlišení obrazu, posunutí, otočení, zkosení 2D obrazu a takzvanému „rovnání prostoru“ (např. u leteckých snímků).

Plošná transformace

- Plošná transformace najde k diskrétnímu bodu (x, y) ve vstupním obrazu odpovídající bod ve výstupním obrazu (x', y') – obecně spojité souřadnice.
- Určení transformačních vztahů:
  - Jsou dány předem – rotace, zvětšení, zkosení, ...
  - Je nutné je hledat na základě znalosti původního i transformovaného obrazu – obvykle pomocí známých bodů, které lze snadno najít v obou obrazech.
- Transformační vztahy se většinou aproximují polynomem n-tého řádu.

$$x' = \sum_{r=0}^{n} \sum_{k=0}^{n-r} a_{rk} x^r y^k, \qquad y' = \sum_{r=0}^{n} \sum_{k=0}^{n-r} b_{rk} x^r y^k \qquad 1.3$$

Pokud nedochází k náhlým změnám pozic, vystačíme si většinou s polynomem do stupně n= 3.

Koeficienty a_{rk} a b_{rk} lze určit např. metodou nejmenších čtverců (množiny sobě odpovídajících bodů (x, y) a (x', y')).

Jasová transformace

- Jasová transformace najde jas, který bude ve výstupním obrazu po geometrické transformaci odpovídat jednotlivým pixelům.
- Transformované souřadnice x', y' leží mimo rastr, přitom geometricky transformovaný obraz musí být reprezentován maticí.
- Možná řešení:
  - metoda nejbližšího souseda,
  - aritmetický průměr čtyř nejbližších sousedů,
  - lineární interpolace,
  - kubická interpolace.

1.1.4 Lokální předzpracování

Pro filtraci jednoho obrazu v jeho prostorových souřadnicích se používají lokální operace – využívá se nadbytečnosti dat (stejné pixely v okolí). Lokální operace využívají pro výpočet jasu bodu ve výstupním obrazu jen lokální okolí odpovídajícího bodu ve vstupním obrazu – jedná se buď o typický reprezentant, nebo kombinace hodnot.

Rozlišujeme několik druhů lokálního předzpracování. Záleží na kritériích dělení:

a) podle cíle – potlačení šumu, detekce hran,
b) podle funkčního vztahu – lineární, nelineární.

Lokální lineární filtry

U lokálních lineárních filtrů je intenzita bodu rovna součtu součinů intenzit bodů v okolí a příslušných váhových koeficientů (matice koeficientů = filtr).

Konvoluce

Jas v bodě (i, j) je dán lineární kombinací jasů v okolí O (velikosti MxN) vstupního obrazu g s váhovými koeficienty h (konvoluční jádro = filtr).

- Pro izoplanární systémy (nezávislé na poloze) = diskrétní konvoluce:

$$f(i,j) = \sum_{m=i-M}^{M/2} \sum_{n=j-N/2}^{N/2} h \cdot (m-i, n-j) \cdot g(m,n) \qquad 1.4$$

- Součet váhových koeficientů vyhlazovacích filtrů musí vždy být roven jedné.

Průměrování

Existuje několik různých variant

- <u>Lokální aritmetický průměr</u>
  - Lineární operace, proto můžeme řešit konvolucí
  - Rozmazává hrany
- <u>Průměr se zvýšením váhy středu</u>
- <u>Filtr s Gaussovým rozložením</u>
  - Gaussovo (normální) rozložení 1.5

$$1D: G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{x^2}{2\sigma^2}}, 2D: G(x) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \qquad 1.5$$

- <u>Průměrování s omezením změn</u>
  - Povolení jen určitých změn mezi původním jasem a výsledkem průměrování (menších/větších než).

- Spočítáme konvoluci, porovnáme s původním jasem a podle výsledku zvoleného kritéria buď dosadíme novou hodnotu, nebo necháme původní.

Lokální nelineární filtry

Intenzita bodu není dána lineární kombinací vstupních hodnot, ale jiným algoritmem, nejčastěji vybírá některou z hodnot ve stanoveném okolí.

- Medián
  - Medián číselné posloupnosti je číslo, které se po uspořádání podle velikosti nachází uprostřed této posloupnosti.
  - Výhoda: redukuje rozmazávání hran; nevýhoda: poškozuje tenké čáry a ořezává ostré rohy.
- Modus
  - Modus je nejčastěji se vyskytující hodnota v množině čísel.
  - U tohoto filtru je zpracovávaný pixel nahrazen modusem daného okolí (je to hodnota intenzity s největší relativní četností).

- Minimum/maximum
  - Filtr typu minimum/maximum je často označovaný také jako filtr eroze/dilatace.
  - Pracuje tak, že vybírá z blízkého okolí bod s minimální/maximální hodnotou intenzity a tu pak dosadí do výsledného bodu.
  - Pokud použijeme minimum blízkého okolí, dojde k potlačení šumu ve tmavých částech obrazu, ale také zeslabení čar a eroze objektu (světlý objekt na tmavém pozadí).
  - Pokud použijeme maximum blízkého okolí dojde k potlačení šumu ve světlých částech obrazu, ale také zesiluje čáry a zvětšuje objekt.

- Konzervativní filtr
  - Konzervativní filtr se používá k odfiltrování izolovaných pixelů s výjimečně vysokou nebo nízkou intenzitou (například šum typu sůl a pepř).
  - Nalezne min. a max. hodnotu intenzity z okolí počítaného pixelu.
  - Je-li hodnota tohoto pixelu mezi minimem a maximem, je ponechána původní hodnota.
  - Jinak je pixel nahrazen novou hodnotou, jež je odvozena z nepostižených sousedních pixelů. Např. v případě, že je jeho hodnota menší než minimum, je nahrazen tímto minimem a naopak.

1.1.5 Filtrace šumu a poruch

Filtrace dat je proces, který transformuje data takovým způsobem, že struktury určitého charakteru zesiluje či potlačuje. Filtrace šumu je proces vyhlazování dat, což je prováděno zeslabením statistických fluktuací.
Řeší se jako potlačení náhodného šumu, ale i jiných náhlých změn (ostré čáry hrany). Využívá se nadbytečnosti dat (stejné pixely v čase nebo okolí).

Časová filtrace

Časová filtrace je filtrace přes více snímků. Pro statické scény lze použít průměrování stejných pixelů přes více snímků nebo jiná statistika (medián atd.) Výhodou této metody je, že nerozmazává hrany

$$f(i,j) = \frac{1}{n}\sum_{k=1}^{n} g_k(i,j) \qquad 1.6$$

Pro dynamické scény, kdy se pohybuje objekt před statickým pozadím je nejjednodušší pořídit snímky bez objektu, pokud to není možné, lze použít model prostředí (pozadí), např. model s lineárním zapomínáním.

$$b(i,j) = \alpha \cdot g_k(i,j) + (1-\alpha) \cdot b(i,j) \qquad 1.7$$

Pro dynamické scény, kdy se např. pohybuje kamera, nebo chceme filtrovat šum pohybujícího se objektu, je třeba nejdříve analyzovat vlastnosti pohybu (např. optický tok).

Prostorová filtrace

- Lineární filtry

U lineárních filtrů je intenzita bodu je rovna součtu součinů intenzit bodů v okolí a příslušných váhových koeficientů (matice koeficientů = filtr).

- Nelineární filtry

U nelineárních filtrů intenzita bodu není dána lineární kombinací vstupních hodnot, ale jiným algoritmem, nejčastěji vybírá některou z hodnot ve stanoveném okolí.

Filtrace ve frekvenční oblasti

- Tato filtrace probíhá po převodu do frekvenční oblasti – nejčastěji Fourierova nebo kosinová, vlnková (wavelet) transformace
- Lze aplikovat na celém obrazu či jen na výřezu

Filtrace v prostorové oblasti

lineární kombinace vstupního obrazu s koeficienty (většinou lokálního) filtru – konvoluce obrazu s filtrem

Filtrace ve frekvenční oblasti

převod do frekvenční oblasti (např. Fourierova transformace), tam filtrace a převod zpět – součin spekter obrazu a filtru

Fourierova transformace

- 1D Fourierova transformace
  - Provádí jednoznačný obousměrný převod signálů mezi časovou reprezentací f(t) a frekvenční reprezentací F(ξ).
  - Umožňuje analyzovat frekvenční obsah (spektrum) signálu.
  - Každá 1D funkce se dá vyjádřit jako integrál (vážený součet) mnoha komplexních exponenciál – sinusovek a kosinusovek.
- 2D Fourierova transformace
  - Je to dvojnásobná 1D Fourierova transformace

1.1.6 Rekonstrukce obrazu

Snaha o nalezení modelu poruchy a odhadu jeho parametrů pro konkrétní třídy obrazů (konkrétní aplikace = stejná porucha). Řešení inverzní úlohy k úloze modelování poruchy. Obvykle se uvažuje lineární model poruchy (konvoluce přes celý obrázek)

$$g(x,y) = \iint_{(a,b)\in O} f(a,b)h(a,b,x,y)dadb + v(x,y) \qquad 1.8$$

1.1.7 Matematická morfologie

Používá se pro, předzpracování (odstranění šumu, zjednodušení tvaru objektů), zdůraznění struktury objektů (kostra, ztenčování, zesilování, výpočet konvexního obalu, označování objektů), popis objektů číselnými charakteristikami (plocha, obvod, projekce).

Úlohu vyhodnocení obrazu si geometrizuje. Základem jsou tvar objektů a transformace, které ho zachovávají. Jsou realizované jako relace obrázku s jinou menší bodovou množinou = strukturní element.

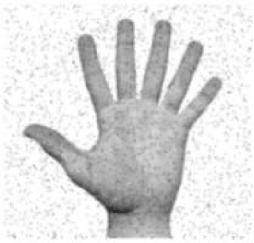
Nacteni testovaciho obrazku

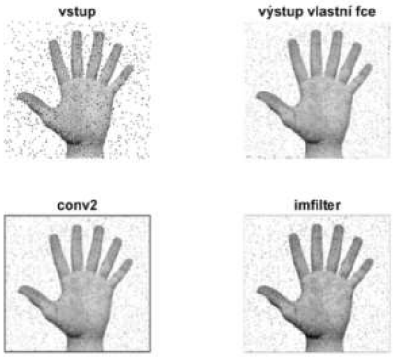
```
I = imread('obraz03.bmp');      % originalni obraz I
I = double(I)/255;
Is = imnoise(I, 'salt & pepper', 0.05);    % obraz s pridanym sumem „pepr a sul'
```

a) Potlaceni sumu

volani vlastni funkce pro vypocet konvoluce

```
xa = floor(size(Ha,1)/2);
tic;
vysIs = ImageConv(Is, Ha, xa);
timer = zeros(2,3);
timer(1,1)= toc;
% zvlast radky a sloupce
tic;
separSumu = ImageConvSep(Is, HaR, HaS, 2);
separTimer = toc;
figure(3);
imshow(separSumu);
% volani knihovni funkce conv2
tic;
c2 = conv2(Is, Ha);
timer(1,2) = toc;
% volani knihovni funkce imfilter
tic;
imf = imfilter(Is, Ha);
timer(1,3) = toc;
% casove srovnani pro jednotlive funkce
% vykresleni vysledku
figure(1);          % vyuzijte vhodne prikaz subplot pro rozdeleni okna figure
subplot(2,2,1);
imshow(Is,[])
title('vstup')
subplot(2,2,2);
imshow(vysIs, [])
title('výstup vlastní fce')
subplot(2,2,3);
imshow(c2, [])
title('conv2')
subplot(2,2,4);
imshow(imf, [])
title('imfilter')
```





Obrázek 1.2 Příklad filtrace šumu

Jako praktický příklad na obrázku 1.2 jsem zde uvedl úlohu z kurzu BPC-UIN. Tento skript realizuje jednu z metod předzpracování (konkrétně odstranění šumu).

1.2 Segmentace[2]

Jak bylo již v úvodu řečeno, segmentace je proces dělení obrazu do částí, které korespondují s konkrétními objekty v obraze. K segmentaci je možné přistoupit několika základními způsoby. Podle přístupu dělíme segmentační algoritmy na:

1.2.1 Metody vycházející z detekce hran (edge-based)

Jedná se o metody orientované na detekci významných hran v obraze. Lokální hrany jsou detekovány pomocí hranových detektorů na základě rozdílu hodnot. Hranový detektor je algoritmus, který produkuje množinu hran (bodů, pixelů, nebo fragmentů) v obraze.

1.2.2 Metody orientované na regiony v obraze (region-based)

Principiálně jsou stejné jako edge-based metody. Pokud lze identifikovat hrany, měly by teoreticky ohraničovat celý region, ani v opačném případě není zaručeno, že hranice regionů nalezené edge-based metodou budou stejné jako ty nalezené region-based metodou.

1.2.3 Statistické metody

V tomto případě je základem segmentace statistická analýza obrazových dat, nejčastěji hodnot pixelů. Strukturní informace je obvykle zanedbávána.

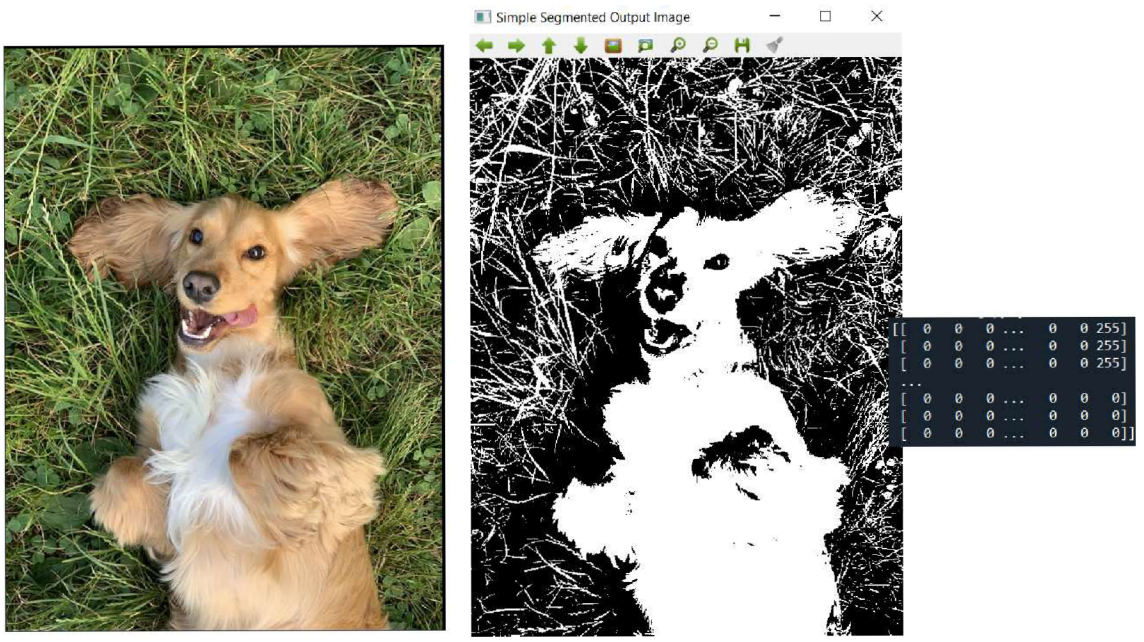
1.2.4 Hybridní metody

Některé segmentační techniky je těžké zařadit do jedné z předchozích tří kategorií, protože obsahují prvky každé z nich, mluvíme tedy o tzv. hybridních metodách. Mezi hybridní řadíme také metody založené na matematické morfologii, jedná se o skupinu metod, která pro segmentaci využívá matematických charakteristik obrazu, např. průběh gradientu.

1.2.5 Znalostní metody (knowledge-based)

Znalost vlastností segmentovaných objektů (tvar, barva, struktura, apod.) mohou segmentaci značně ulehčit. Metody patřící do této kategorie využívají atlas předloh či modelů segmentovaných objektů. Atlas je generován automaticky ze souboru trénovacích dat, nebo jsou do něj informace vloženy ručně, na základě lidské zkušenosti. V průběhu segmentace algoritmus hledá transformaci známých objektů, šablon v atlasu, na objekty nalezené v obraze. Tento proces se obvykle nazývá atlas-warping a nejčastěji využívá lineární transformace.

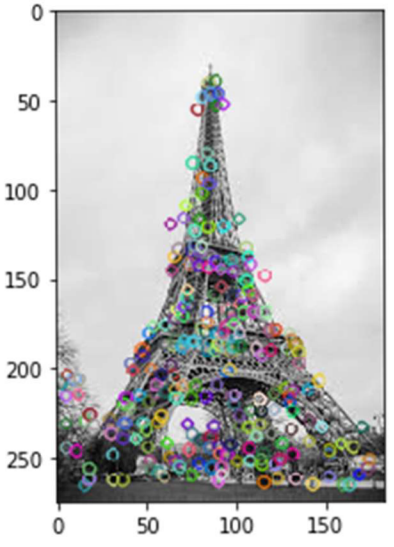
Na obrázku 1.3 níže můžeme vidět příklad segmentace, konkrétně Thresholding, tato metoda nastavuje pixely, jejichž hodnota je větší, než předem daná hodnota na maximální.



Obrázek 1.3 Příklad segmentace

1.3 Detekce klíčových bodů

Detekce klíčových bodů je první část zpracování obrazu v mém návrhu řešení. Klíčový bod je nezaměnitelné místo ve zpracovávaném obrázku typicky se jedná o roh nějakého objektu či značky. Na obrázku 1.4 jsou vyznačené příklady klíčových bodů.

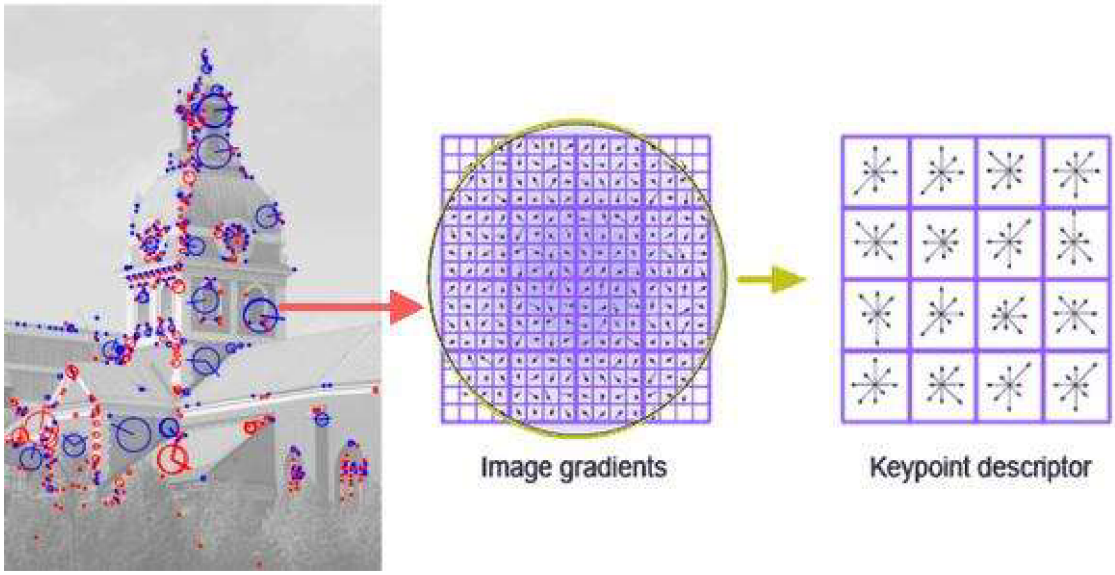


Obrázek 1.4 Klíčové body

Pro detekci klíčových bodů existuje řada metod. V této práci byla implementována metoda SURF, je zde však popsáno i několik dalších metod, které by bylo možné implementovat.

1.3.1 SIFT (Scale-Invariant Feature Transformation)[9]

Nejprve je zdvojnásobena šířka a výška obrázku pomocí bilineární interpolace. Pak je na obrázek několikrát aplikována filtrace pomocí Gaussova filtru. Následně je opakovaně snižováno rozlišení, dokud je obrázek možné zpracovat, jednotlivé mezivýsledky tvoří tzv. *oktávy*. Naskládáním oktáv za sebe získáme *měřítkový prostor*. Díky tomu je možné simulovat různé vzdálenosti bodů na vyfocených objektech. Nyní je spočítán rozdíl Gaussianů, ve kterém jsou hledány extrémy, z těch jsou brány klíčové body.
K popisu nebo k tvorbě deskriptorů je použito okolí klíčových bodů. Toto okolí je rozděleno na podoblasti, ze kterých je spočítán gradient. Z gradientů je vytvořen histogram. Hodnoty jednotlivých histogramů jsou následně diskretizovány a poskládány do vektoru dlouhého 128 bitů. Proces tvorby gradientu je na obrázku 1.5.



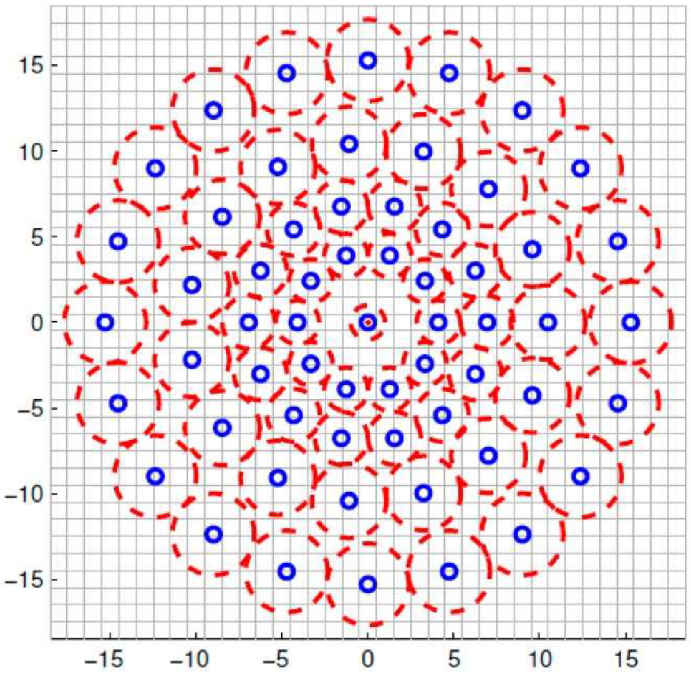
Obrázek 1.5 Deskriptor metody SIFT

1.3.2 BRISK metoda (Binary Robust Invariant Scalable Keypoints)[3]

Tato metoda se skládá ze tří hlavních modulů, detekce klíčových bodů, popis klíčových bodů a jejich sesouhlasení. K detekci klíčových bodů je využito podobného principu jako u AGAST (Adaptive and Generic Accelerated Segment Test). Detekce klíčových bodů se skládá z několika částí. Nejprve je obrázek rozdělen podle měřítek do takzvané *pyramidy měřítkových prostorů*. Dále se prochází každá vrstva této pyramidy a

hledají se potenciální klíčové body, které se nakonec porovnají napříč vrstvami, a vyberou se finální klíčové body.

Následuje popis jednotlivých bodů. K jejich popisu se obvykle používá nejbližší okolí 40x40 pixelů, ve kterém jsou vytvořeny 4 soustředné kružnice. V těch je rovnoměrně rozděleno 60 bodů, ze kterých se složí deskriptor ve formě binárního bit stringu. Ilustrace k procesu tvorby deskriptorů je na obr. 1.6.



Obrázek 1.6 Deskriptor metody BRISK

Finální část je srovnávání deskriptorů. K jejich porovnání se používá Hammingova vzdálenost, která určí míru podobnosti dvou deskriptorů. Čím vyšší je vzdálenost, tím menší podobnost je.

1.3.3 SURF metoda (Speeded Up Robust Features)[4,5]

Jedná se o robustní metodu, která popisuje obrázek pomocí deskriptorů. Je to novější obdoba metody SIFT. Velká výhoda této metody je, že popis pomocí deskriptorů je invariantní vůči rotaci a vzdálenosti popisovaného objektu od kamery.[2] Činnost SURF algoritmu je rozdělena do tří hlavních částí.

Nejprve se vyberou ‚klíčové body', pro ty jsou většinou vybrány rohy, skvrny atd. Důležité u klíčových bodů je, aby mohly být nalezeny i za jiných podmínek snímání.

Okolí každého bodu je reprezentováno vektorem příznaků. Tyto deskriptory musí být rozpoznatelné i za určitého šumu, geometrických či fotometrických chyb a i při určitých chybách detekce.

Nakonec se získané deskriptory porovnají a vyberou se shodné.

1.3.3.1 Detekce

Hessova matice [6]

Hessova matice je čtvercová matice parciálních derivací funkce o více proměnných. Využívá se k určení inflexních nebo kritických bodů, zde je využíván k detekci klíčových bodů. Pokud uvažujeme spojitou funkci dvou proměnných f (x, y), Hessova matice bude mít tvar:

$$H\big(f(x,y)\big) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix} \qquad 1.9$$

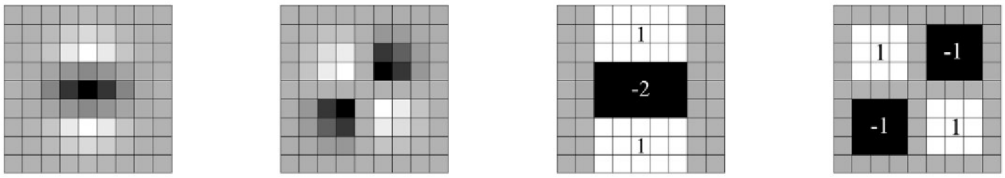
Hessenský Detektor

Základem Hessenského detektoru je výše popsaná Hessova matice. U Hessovy matice nahradíme f (x, y) intenzitami pixelů I(x, y). Hessova matice H(X, σ), v každém bodě X(x, y) obrázku I a v měřítku σ může být definován jako:

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \qquad 1.10$$

Kde L_{xx} je konvoluce druhá derivace Gaussovy funkce $\frac{\partial^2}{\partial x^2} g(\sigma)$ obrázku I v bodě X, podobně pro $L_{xx}(X, \sigma)$ a $L_{xx}(X, \sigma)$.

Maticové filtry 9x9 na obrázku 1.5 jsou aproximace pro druhou derivaci Gaussovy funkce s σ = 1,2 a reprezentací našeho nejmenšího měřítka.



Obrázek 1.7 Druhá derivace Gaussovy funkce

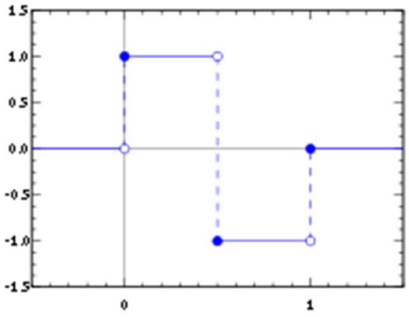
Naše aproximace značíme D_{xx}, D_{yy} a D_{xy}. Váhy aplikované na obdélníkové regiony jsou jednoduché kvůli efektivitě výpočtu.

1.3.4 Popis

Deskriptory metody SURF jsou podobné deskriptorům metody SIFT. První krok spočívá v určení orientace založené na informacích poskytnutých okolím klíčového bodu. Následně vytyčíme čtvercový region, zarovnaný podle vybrané orientace a z něj získáme deskriptor.

Určení orientace

Za účelem nezávislosti na rotaci, si určíme směr klíčových bodů. Pro tento úkol nejprve spočítáme odezvy Haarovy vlnky ,ve směrech x, y a v kruhovém
okolí klíčového bodu o poloměru 6s, kde s je měřítko, se kterým byly detekovány klíčové body. Vzorkovací krok je také závislý na měřítku a jeho velikost je s. Ke spočítání odezev v jakémkoliv měřítku je potřeba 6 operací. Délka strany jedné vlny je 4s.



Obrázek 1.8 Haarova vlnka

Jakmile jsou odezvy a váhy spočítány, jsou reprezentovány pomocí vektorů podél os x a y. Po sečtení odezev všech dvojic vektorů je výsledná orientace určena podle nejdelšího, nově vzniklého, vektoru.

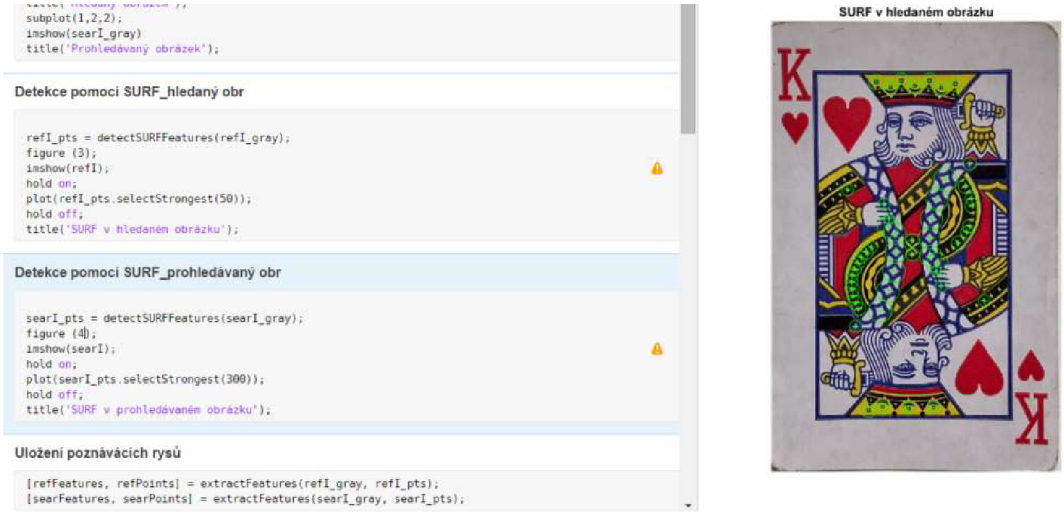
Deskriptory

Pro získání deskriptoru je prvním krokem konstrukce čtvercového regionu kolem klíčového bodu s orientací určenou podle postupu výše. Velikost regionu je 20s. Tento region je rovnoměrně rozdělen na menší 4x4 sub-regiony. Tento proces uchová důležité prostorové informace. Pro každý sub-region je spočítán několik jednoduchých rysů. Pro jednoduchost nazýváme d_x odezvu na Haarovu vlnku v horizontálním směru a d_y ve vertikálním. Pro zvýšení robustnosti proti geometrické deformaci a lokalizačním chybám jsou nejprve určeny váhy pomocí Gaussovy funkce (σ=3,3s) se středem v klíčových bodech.

Následně jsou odezvy d_x a d_y sečteny v každém sub-regionu a vytvoří první vstupy vektoru rysů. Za účelem získání informací o změně polarity a intenzity, sečteme také absolutní hodnoty d_x a d_y. Nyní má každý sub-region čtyřrozměrný deskriptorový vektor v.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \qquad 1.11$$

Výsledkem je vektor (o délce 64 prvků) pro všechny 4x4 sub-regiony. Odezvy jsou invariantní vůči biasu. Invarianci vůči kontrastu docílíme převedením deskriptorů na jednotkový vektor.



Obrázek 1.9 Hledání deskriptorů metodou SURF

1.3.5 Porovnání deskriptorů

Proces porovnání deskriptorů probíhá pomocí SURF vektorů deskriptorů, které jsou používány k porovnávání mezi odlišnými obrázky. Tento krok je nejčastěji založen na vzdálenosti mezi vektory. Čas výpočtu je u SURFu nejvíce závislý na velikosti vektorů. Proto jsou žádoucí co nejmenší vektory. Na obrázku níže můžete

Obrázek 1.10 Porovnávání poloh sesouhlasených deskriptorů

1.4 Homografie[7]

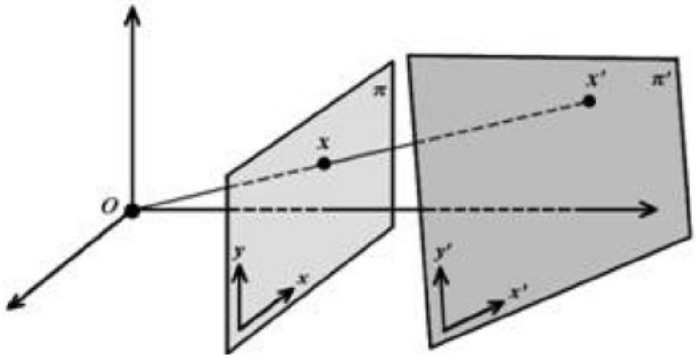
Pracuje s údaji o předmětech na snímané scéně, s jejich tvary, účely a vzájemné vzdálenosti. Zpracováním těchto dat je možné rekonstruovat okolí a umožnit tak například efektivní řízení pohybu mobilního robotu, nebo sledování předem definovaného pohyblivého předmětu.

Díky stereoviznímu kamerovému systému je možné z dvojice snímků pasivně dopočítat pozici snímaných objektů v prostoru, nezávisle na vzájemném relativním pohybu kamery a scény. Aby bylo možné prostorové vzdálenosti dopočítat, je nutné najít ve stereo páru dostatečné množství stereo korespondencí, tedy 2D bodů v levém a pravém obraze, které jsou projekcemi stejného 3D bodu na scéně.

1.4.1 Projektivní lineární transformace

Je to invertibilní transformace mezi dvěma projektivními perspektivami, kde zásadní vlastností je mapování přímek opět na přímky. Odtud vznikl synonymní název kolinearita, jenž však může mít obecnější význam. Můžeme se setkat také s názvem projektivita a homografie. Projektivita tedy vyjadřuje, jak se mění vjem pozorovaného předmětu, pokud se mění pozice a/nebo úhel pohledu pozorovatele.

Příkladem projektivní transformace je středové promítání se středem O a dvojicí ploch π a π', jež mapuje body z jedné plochy na body druhé plochy $x_i \leftrightarrow x_i'$. Z tohoto je zřejmé, že také přímky jedné plochy jsou

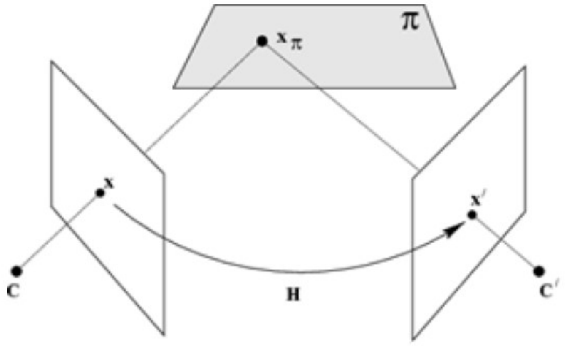


Obrázek 1.11 Homografie

mapovány opět na přímky druhé plochy. Pro body v homogenních souřadnicích lze toto mapování zapsat jako $x_i^{'} = Hx_i$, kde H je transformační matice homografie o rozměru 3x3:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \qquad 1.12$$

$$x_i' = Hx_i = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i^{'} \\ y_i^{'} \\ w_i^{'} \end{pmatrix} \qquad 1.13$$



Obrázek 1.12 Projektivní transformace

1.4.2 Výpočet homografie

Záměrem je nalézt transformační matici, pro níž platí:

$$2\, x_i' = Hx_i \qquad 1.14$$

x_i ‘ a Hx_i se však číselně nerovnají, protože se liší v měřítku (daném w_i ‘). Přesto však můžeme zapsat:

$$(x_i')_\times H x_i = 0 \qquad 1.15$$

Pokud nahradíme $(x_i')_\times$ dostaneme soustavu:

$$\begin{pmatrix} 0^T & -w_i' x_i^T & y_i' x_i^T \\ w_i' x_i^T & 0^T & -x_i' x_i^T \\ -y_i' x_i^T & x_i' x_i^T & 0^T \end{pmatrix} h = 0 \qquad 1.16$$

Tyto rovnice mají podobu:

$$A_i h = 0 \qquad 1.17$$

Kde A_i je matice 3x9 a vektor h se rovná:

$$h = (h_{11}; h_{12}; h_{13}; h_{21}; h_{22}; h_{23}; h_{31}; h_{32}; h_{33})^T \qquad 1.18$$

A_i má hodnost, tudíž je každá korespondence vyjádřena dvojicí rovnic.

Složením rovnic pro čtyři body vzniká matice A jejíž hodnost je 8 a tvoří lineární homogenní soustavu rovnic pro 9 neznámých. Z toho plyne, že stačí, když matice obsahuje 8 lineárně nezávislých řádků. Řešením je potom nulový prostor této matice. Body musí být zvoleny tak, aby žádné tři neležely na stejné přímce.

$$Ah = \begin{pmatrix} x_1 & x_1 & 1 & 0 & 0 & 0 & -x_1 x_1' & -y_1 x_1' & -x_1' \\ 0 & 0 & 0 & x_1 & x_1 & 1 & -x_1 y_1' & -y_1 y_1' & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 x_2' & -y_2 x_2' & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 0 & -x_2 x_2' & -y_2 y_2' & -y_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_n x_n' & -y_n x_n' & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_1 y_n' & -y_n y_n' & -y_n' \end{pmatrix} = 0 \qquad 1.19$$

V praxi se používá korespondencí více než 4, aby se zmenšil vliv chyby při určení korespondujících bodů. Kvůli chybám je však také nulový prostor matice A prázdný a vektor h se hledá ve smyslu nejmenších čtverců, tj. určí se jako:

$$h = \underset{||h^*||=1}{\operatorname{argmin}} ||Ah^*||^2 \qquad 1.20$$

Kde $||X||$ je Euklidovská norma $\sqrt{\sum X_i^2}$. Nalezení vektoru h se v takovém případě udělá pomocí **SVD** (Singular Value Decomposition).

2. NÁVRH ŘEŠENÍ

Předchozí část obsahuje teoretický rozbor. problematiky této práce. Tato část se zabývá praktickým řešením vytyčených cílů a sice

1. Vhodné předzpracování obrázku
2. Ošetření různých úhlů snímání obrázků
3. Vyhledání klíčových bodů u obou obrázků
4. Porovnání obrázků a zvýraznění změn

2.1 Předzpracování

Jako předzpracování většinou postačí převedení do černobílé, které je nutné pro další zpracování snímků. Pro případ, že by byl obrázek zašuměný, vlivem špatných světelných podmínek nebo použitím špatné kamery, byla ještě implementována možnost aplikovat Gaussův filtr, který by měl šum redukovat. Pro případnou filtraci Gaussovým filtrem byla využita funkce Matlabu *imGaussFilt*.

2.2 Vypálení binární masky

Prvotní pokus o porovnání obrázků a zvýraznění změn byl proveden pomocí binární masky. Jedná se o jednoduchou metodu, kde nejprve oba vstupní obrázky převedeme na binární. K tomu použijeme funkci *imbinarize*[7]. Funkce *imbinarize* je knihovní funkce Matlabu, která nahrazuje hodnoty jednotlivých pixelů obrázku za 0 nebo 1, podle thresholdu. Ten defaultně volí podle Otsuovy metody. Následně oba binární obrázky odečteme a vytvoříme tak binární masku, kterou následně vypálíme do druhého obrázku pomocí funkce, která.

Obrázek 2.1 Snímky dodané na vstupu

Funkčnost tohoto řešení je ale výrazně omezená na nejideálnější případy. Na obrázku 2.1 můžete vidět snímky dodané na vstupu a na obrázku 2.2 pak výstup.

Obrázek 2.2 Výstup z dat na obrázku 2.1

Můžeme vidět, že pro tento vstup toto řešení funguje. Pokud ale dáme na vstup snímky na obrázku 2.3, výstup již není tak ideální.

Obrázek 2.3 Snímky dodané na vstupu

Obrázek 2.4 Výstup z dat na obrázku 2.3

2.3 Srovnávání deskriptorů

Toto řešení již je o něco robustnější a využívá odlišného principu jako funkce předchozí a sice metody SURF. Výhoda tohoto přístupu oproti předchozímu spočívá v rozlišování nových objektů a pohybů objektů stávajících.
Nejprve jsou detekovány deskriptory, k tomu je použita funkce Matlabu detectSURFFeatures. Ty je nutné následně sesouhlasit, to je provedeno pomocí funkce matchFeatures. Pak jsou sesouhlasené páry procházeny a vyhledány body, které přibyly, ubyly anebo u kterých došlo ke změně polohy.

Obrázek 2.5 Výstup z dat na obrázku 2.3

2.4 Ošetření různých úhlů

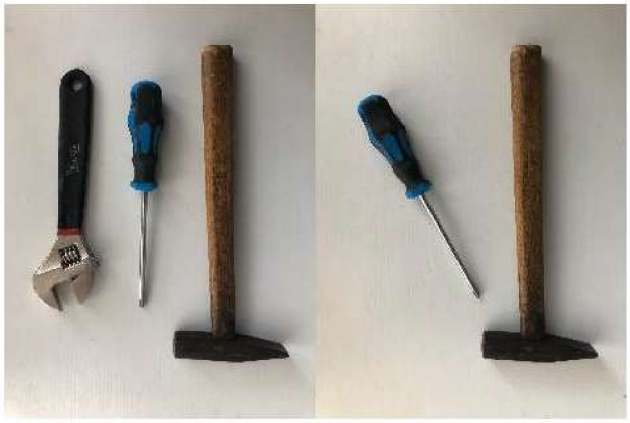
Protože řešení pomocí metody SURF funguje dobře pouze pro obrázky snímané ze stejných úhlů, je nutné ošetřit možnost použití dynamické kamery. Tato problematika byla řešena pomocí homografie, jejíž princip byl popsán dříve.
Metoda, kterou homografii aplikuji, funguje následovně. Nejdříve vybere 5 nejvýraznějších bodů, které byly získány pomocí metody SURF. Následně na tyto body uplatní vzorec 1.19. Z něj získáme koeficienty pro matici homografie H. Tyto koeficienty seskládáme do tvaru

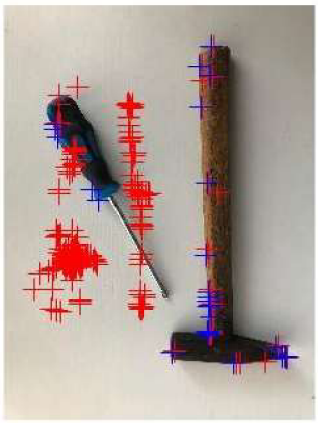
$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \qquad 2.1$$

Homografie se mi ve finální verzi bohužel nepovedla implementovat. Vypočítaná matice homografie se nedařila aplikovat na druhý obrázek. V implementaci jsem se pokoušel použít funkci *imwarp*, bohužel vždy havarovala.

2.5 Zkoušení různých scén

Program byl zkoušen na řadě scén kvůli robustnosti a spektru různých použití. Jako první příklad scény byly zkoušeny různé nástroje, které mají simulovat stěnu v dílně, kde se ukládají nástroje.

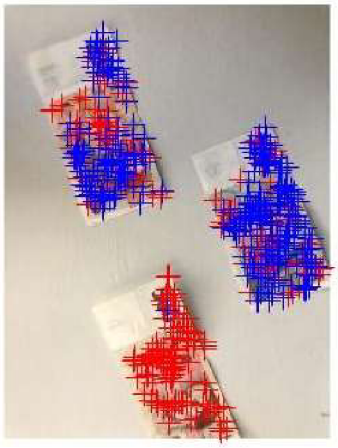




Obrázek 2.6 Testování scény s nástroji

Na obrázku 2.6 Testování scény s nástroji2.6 je zobrazen vstup a výstup z programu. Jak je na obrázku 2.6 vidět, program rozeznal odebrání francouzského klíče (červené křížky reprezentují body, které přibyly nebo ubyly, modré pak změnu polohy). Z obrázku jde poznat, že došlo nepatrnému pohybu kamery při snímání, protože program ukazuje pohyb kladiva. Je také vidět, že program dobře nerozpoznal šroubovák a na výstupu ukazuje odebrání původního a přidání nového. Tento problém může být způsoben nedostatečnou členitostí povrchu šroubováku, kvůli čemuž SURF nenašel dostatek bodů.

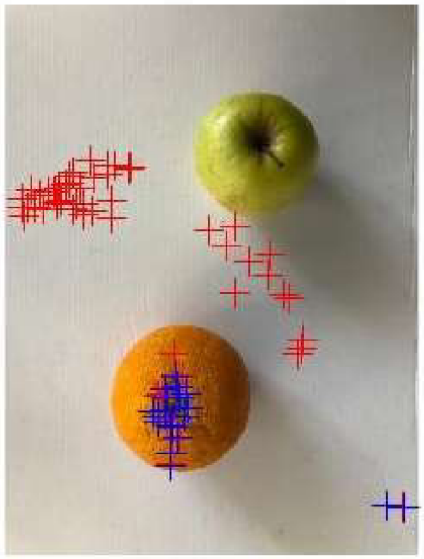
Jako další příklad scény byla simulace dna trezoru, kde jsou položené bankovky. Představa situace je, že na stropu trezoru by byla umístěná kamera, která by snímala dno s penězi. V případě krádeže nebo výměny za padělky by majitele program upozornil.



Obrázek 2.7 Testování scény s penězi

Na obrázku 2.7, je zobrazen výstup programu. Po vložení snímků scény s penězi na vstup, program správně rozeznal objevení bankovky. Na druhou stranu detekoval i pohyb bankovek, u kterých k žádnému výraznému pohybu nedošlo. To může být způsobeno malou velikostí nastavených tolerací.

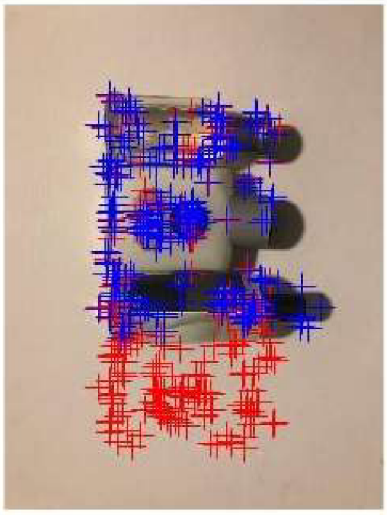
Tato scéna představuje krámek s ovocem. Zde, stejně jako v jiných obchodech by program mohl hlídat vystavené zboží. Na začátku směny by byl pořízen obrázek, který by se průběžně srovnával se současnou situací.



Obrázek 2.8 Testování scény s ovocem

Na obrázku 2.8 opět vidíme výstup programu. Na obrázku jde vidět, že program rozpoznal zmizení banánu i pohyb pomeranče. Problém nastal u jablka, kde metoda SURF nenalezla klíčové body. Pro tuto aplikaci tento program tedy není vhodný.

Poslední vyzkoušená scéna představuje regál v drogerii, kde by program mohl detekovat, jestli je doplněné zboží.



Obrázek 2.9 Testování scény s ovocem

Znovu jsou zde problémy s tolerancemi, program ale opět rozpoznal úbytek objektu.

3. ZÁVĚR

Tato práce se skládá ze dvou částí. První část je teoretická a má za úkol rozbor a vysvětlení problematiky této práce. V této části byly nejprve postupně popsány jednotlivé fáze obvyklého procesu zpracování obrazu. Dále jsou zde rozebrány různé metody detekce klíčových bodů v obrazech, konkrétně metody SIFT, BRISK a SURF. Z těchto metod je v praktické části využita poslední jmenovaná z důvodu rychlého zpracování dat a dobré podpory v programu Matlab. Poslední část teoretické rešerše se zabývá homografií. Homografie zajišťuje sesouhlasení klíčových bodů na obou snímcích tak, aby bylo možné kompenzovat různé úhly snímání obou snímků. Druhá část je praktická. V této části jsou postupně realizovány vytyčené cíle. K tomuto účelu je použit program Matlab. Ten byl vybrán pro názornost a dobrou podporu v oblasti počítačového vidění. Jednotlivé cíle práce jsou rozebrány níže.

3.1 Vhodné předzpracování obrázku

Ve většině případů postačí převedení obrázků do černobílé. Při špatných světelných podmínkách nebo špatné kvalitě kamery může vzniknout šum. Pro tyto případy je zde možnost využití Gaussovského filtru. Defaultně je tato možnost vypnuta, může ale být kdykoliv zapnuta pomocí proměnné *filtration*.

3.2 Vyhledání klíčových bodů v obrázcích

K vyhledání klíčových bodů je použita metoda SURF, tato metoda byla aplikována prostřednictvím programu Matlab, a sice metodou *detectSURFFeatures*, která slouží k nalezení klíčových bodů. Z těchto klíčových bodů jsou pak pomocí metody *extractFeatures* získány deskriptory z obou obrázků. Pomocí srovnání počtu deskriptorů je určeno, zda nějaký objekt přibyl. Tyto deskriptory jsou následně sesouhlaseny pomocí metody *matchFeatures*. U těchto sesouhlasených jsou porovnávány polohy a na základě toho se určí, jestli se poloha změnila nebo zůstala stejná.

3.3 Ošetření různých úhlů snímání obrázků

O ošetření různých úhlů snímání by se v tomto programu měla starat homografie. Homografie je aplikována v této práci formou metody *Homography.m.* V této metodě nejprve dojde k detekci SURF bodů. Z těch jsou následně vybrány nejvýraznější, které jsou použity pro výpočet homografie. Ve výsledném programu se bohužel homografii aplikovat nepovedlo z důvodu problému implementace v programu Matlab.

3.4 Porovnání obrázků a zvýraznění změn

Byly testovány dva přístupy jak obrázky porovnat. Prvním z nich je řešení pomocí binární masky. Postup spočíval v převedení obou obrázků do binárních. Jejich vzájemné odečtení a výsledná maska se vypálí do druhého obrázku. Zvýraznění změn z této metody je realizováno pomocí žlutého polygonu. Toto zvýraznění vypadá dobře a je velmi přehledné. Ovšem tato metoda lze dobře použít pouze pro snímky, kde objekty pouze přibývají nebo ubývají. U snímků, ve kterých dochází k manipulování s předměty, které na scéně už byly, není reprezentace výsledků již tak přehledná.

Druhým přístupem je detekce klíčových bodů pomocí metody SURF. Tento přístup spočívá v detekci klíčových bodů v obou obrázcích, z nichž jsou následně vybrány potřebné body a ty jsou zvýrazněny pomocí křížků ve výsledném obrázku. Tato reprezentace nevypadá tak dobře, tento přístup je ale mnohem robustnější, proto mu byla dána přednost.

LITERATURA

[1] *JANÁKOVÁ, Ilona. Filtrace šumu a poruch. Kurz ZVS [online]. Brno: VUT, 2019 [cit. 2020-12-28]. Dostupné z: http://vision.uamt.feec.vutbr.cz/ZVS/lectures/07_Filtrace_sumu_a_poruch.pdf*

[2] *ING. ŠPANĚL, Michal. Obrazové segmentační techniky [online]. 2006 [cit. 2020-12-22]. Dostupné z: http://www.fit.vutbr.cz/~spanel/segmentace/*

[3] *BRISK - Feature Detector* [online]. 2018 [cit. 2022-05-11]. Dostupné z: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6263410/

[4] *SURF. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-12-22]. Dostupné z: https://cs.wikipedia.org/wiki/SURF*

[5] *SURF. People [online]. [cit. 2020-12-24]. Dostupné z: https://people.ee.ethz.ch/~surf/eccv06.pdf*

[6] *Hessian matrix. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-12-24]. Dostupné z: https://en.wikipedia.org/wiki/Hessian_matrix*

[7] *Homografie a Epipolární Geometrie. Trilobit [online]. Zlín: Univerzita Tomáše Bati, 2010 [cit. 2020-12-26]. Dostupné z: http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie#_ftn1*

[8] *Mathworks - imbinarize. Mathworks [online]. Natick, Massachusetts, USA: The MathWorks, 2016 [cit. 2022-05-06]. Dostupné z: https://www.mathworks.com/help/images/ref/imbinarize.html*

[9] *WEITZ, Prof. Dr. Edmund. SIFT - Scale-Invariant Feature Transform [online]. 2016 [cit. 2022-05-12]. Dostupné z: http://weitz.de/sift/*

OBRÁZKY

[1] *Kličové body [online]. In: . [cit. 2022-05-11]. Dostupné z: https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/*

[2] *SURF. People [online]. [cit. 2020-12-24]. Dostupné z: https://people.ee.ethz.ch/~surf/eccv06.pdf*

[3] *Haarova vlnka. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-12-23]. Dostupné z: https://en.wikipedia.org/wiki/Haar_wavelet*

[4] *Homografie a Epipolární Geometrie. Trilobit [online]. Zlín: Univerzita Tomáše Bati, 2010 [cit. 2020-12-26]. Dostupné z: http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie#_ftn1*

[5] *Deskriptor metody SIFT* [online]. [cit. 2022-05-17]. Dostupné z: https://www.codeproject.com/Articles/619039/Bag-of-Features-Descriptor-on-SIFT-Features-with-O

[6] *Příklad filtrace šumu [online]. Brno: VUT Brno, 2020 [cit. 2022-05-18]. Dostupné z: http://vision.uamt.feec.vutbr.cz/*

[7] *Příklad segmentace. In: Towardsdatascience [online]. [cit. 2022-05-11]. Dostupné z: https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/*