

## **ABSTRAKT**

Tato práce se zabývá zpracováním snímků displeje embedded zařízení a jejich klasifikací. Je zde rozebrána problematika odstranění šumu moaré prostřednictvím filtrace ve spektru a normalizace obrazu pro další analýzu. Pro klasifikaci obrazů jsou využity detektory významných bodů a deskriptory. Hlavní důraz je kladen na detektory FAST a Harrisův detektor rohů a na deskriptory SURF, BRIEF a BRISK a jejich hodnocení z pohledu potenciálního přínosu pro tuto práci.

## **ABSTRACT**

This thesis concerns with processing of embedded terminals' images and their classification. There is problematics of moire noise reduction thought filtration in frequency domain and the image normalization for further processing analyzed. Keypoints detectors and descriptors are used for image classification. Detectors FAST and Harris corner detector and descriptors SURF, BRIEF and BRISK are emphasized as well as their evaluation in terms of potential contribution to this work.

## **KLÍČOVÁ SLOVA**

Deskriptory, hledání bodů zájmu, redukce šumu, zlepšování obrazu, detekce oblastí zájmu, přiřazování klíčových oblastí, detekce tvarů

## **KEY WORDS**

Descriptors, keypoint detection, noise reduction, image enhancement, detection of regions of interest, key regions matching, shape detection

## **BIBLIOGRAFICKÁ CITACE**

LABUDOVÁ, K. Zpracování obrazu pro ovládání robotické ruky. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 28 s. Vedoucí seminární práce Ing. Vratislav Harabiš, Ph.D.

## **PROHLÁŠENÍ**

Prohlašuji, že svoji seminární práci na téma Zpracování obrazu pro ovládání robotické ruky jsem vypracovala samostatně a pod vedením vedoucího seminární práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci této práce. Jako autorka uvedené seminární práce dále prohlašuji, že v souvislosti s vytvořením této seminární práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 3.ledna 2017

.....

podpis autora

## **PODĚKOVÁNÍ**

Chtěla bych poděkovat svému vedoucímu seminární práce Vratislavu Harabišovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

V Brně dne 3. ledna 2017

.....

podpis autora

# OBSAH

ÚVOD.....	9
1 PŘEDZPRACOVÁNÍ OBRAZU .....	10
1.1 Normalizace obrazu .....	10
1.1.1 Detekce rohů displeje.....	10
1.1.2 Normalizace jasu.....	11
1.2 Redukce šumu.....	11
2 DETEKCE OBRAZOVEK.....	14
2.1 Detekce zajímavých bodů v obraze .....	16
2.1.1 Harissův detektor rohů .....	16
2.1.2 FAST – Feature from Accelerated Segment Test .....	17
2.2 Deskriptory .....	18
2.2.1 SIFT – Scale Invariant Feature transform.....	18
2.2.2 SURF – Speeded Up Robust Features[13].....	18
2.2.3 BRIEF – Binary Robust Independent Elementary Features .....	20
2.2.4 ORB .....	21
2.2.5 BRISK – Binary Robust Invariant Scalable Keypoints .....	21
3 AKTUÁLNÍ STAV .....	24
4 ZÁVĚR .....	25
5 SEZNAM LITERATURY .....	26
6 SEZNAM ZKRATEK A PŘÍLOH .....	28
6.1 Seznam zkratk.....	28

## SEZNAM OBRÁZKŮ

Obr. 1 – Mezikroky v algoritmu pro nalezení rohů displeje – vstupní obraz (vlevo nahoře), binarizovaný obraz (vpravo nahoře), segmentovaný obraz (vlevo dole), výsledek (vpravo dole) .....	10
Obr. 2 - Snímek obrazovky před (vlevo) a po(vpravo) transformaci kontrastu .....	11
Obr. 3 – Příklady problematických obrazovek: a) jinak barevný okraj b) výrazný šum c) nízký jas (obrazovka je reálně bílá) d) nemonotónní obrazovka s výrazným zastoupením bílé, která musí pokračovat k analýze .....	12
Obr. 4 – Schéma filtrace ve výkonovém spektru .....	13
Obr. 5 – Výsledek filtrace šumu typu moaré .....	13
Obr. 6 – Výsledek filtrace šumu typu moaré na monotónním obraze.....	13
Obr. 7 – Obrazová pyramida .....	15
Obr. 8 – Reprezentace vlastností objektů v obraze na základě vlastních hodnot matice M ....	17
Obr. 9 – Test pro detekci rohů v segment 12x12 pixelů. Zvýrazněné čtverce jsou pixely v určité vzdálenosti od středu potenciálního příznaku p a jsou používány pro jeho hodnocení. Část kružnice, která je naznačená čárkovaně, označuje pixely, které jsou alespoň o prahovou hodnotu jasnější než p. [10].....	18
Obr. 10 – diskrétní a ořezaná druhá parciální derivace Gaussovy funkce Dyy, dále Dxy, krabicový filtr aproximující Dyy, krabicový filtr aproximující Dxy .....	19
Obr. 11 – Interpolace bodu zájmu ve vrstvě $c_i$ a maxim ve vrstvách $d_i$ a $d_{i-1}$ [24].....	22
Obr. 12 – Schéma vzorkování, modré body naznačují vzorkovací pozice. Poloměr čárkovaných červených kruhů koresponduje se standartní odchylkou $\sigma$ Gaussova vyhlazovacího filtru aplikovaného v bodech vzorkování [24] .....	22

## ÚVOD

Tato práce se zabývá zpracováním obrazu pro účely navádění robotické ruky po dotykovém displeji embedded zařízení tiskáren. Tento systém s robotickou rukou pomáhá ve firmě Y Soft Corporation testovat software SafeQ určený pro tiskový management. Celé oddělení Výzkumu a vývoje funguje na bázi kontinuálního vývoje, a proto je několik vybuildovaných verzí SafeQ týdně. Ideálně by se mělo provést regresní testování pro každou verzi a každého výrobce tiskáren, kterých máme v dnešní době jedenáct, pro podchycení chyb vzniklých v průběhu vývoje. Toto je velmi náročné až nemožné splnit, kvůli časové náročnosti a počtu lidí, kterých by byla potřeba. Navíc buildování se děje večer a nejlepší by bylo zjistit případné problémy v noci, aby je mohli vývojáři začít opravovat už ráno. Robot je dobrým řešením, protože může pracovat přes noc, neunaví se a může provádět repetitivní úkony, aniž by ztrácel pozornost jako lidští testéři. Další pozitivum je škálovatelnost a finanční úspora.

Tento robot už je několik let ve fázi vývoje. Robotická ruka je naváděná kamerou, která snímá displej embedded zařízení. Část pro zpracování obrazu je naimplementovaná, ale neobsahuje předzpracování obrazu a není dostatečně rychlá ani spolehlivá. Náplň této práce je vybrat vhodné metody pro předzpracování obrazu a implementovat je. Dále nastudovat algoritmy pro rozpoznávání a klasifikaci obrazů, vybrat ty nejslibnější na základě odborné literatury a poté je implementovat a porovnat. Nejlepší metoda se použije v praxi. Na závěr se zhodnotí rychlost a spolehlivost řešení.

# 1 PŘEDZPRACOVÁNÍ OBRAZU

## 1.1 Normalizace obrazu

### 1.1.1 Detekce rohů displeje

První krok předzpracování je výřez displeje z obrazu. Je tak dáno z několika důvodů, hlavním je snížení plochy pro další zpracování a tím snížení náročnosti na paměť a CPU. Další je normalizace rozměrů pro určení vzdálenosti na displeji a určení polohy ovládacích prvků z databáze bez nutnosti automatické detekce. Aktuálně je naimplementovaná afinní transformace [1], ale rohy displeje se volí ručně přes GUI. Toto není ideální z hlediska nutnosti zásahu obsluhy robota. Další nevýhodou je, že se transformace provádí jen jednou, na začátku, a pokud se kamera během testování posune, je zbytek testování zbytečné. Proto byla naimplementována automatická detekce rohů displeje v obraze a podle pozice rohu se bude program kalibrovat v pravidelných intervalech.

Jednotlivé mezikroky zpracování je možné vidět na sérii obrázků Obr. 1. Nejprve se obraz převede na šedotónový, pak na binární s vhodným prahem. Binární obraz se nasegmentuje [1] a dále se provedou morfologické operace otevření a uzavření [1] kvůli redukci malých segmentů způsobených nerovnoměrným osvětlením a kvůli zajištění celistvosti větších segmentů. Pomocí kritérií pro délku kontury kolem segmentu a obsahu segmentu se vybere oblast, kde je s největší pravděpodobností displej. Nakonec se vyberou čtyři body v segmentu, jejichž Euklidovská vzdálenost je nejmenší k rohům obrazu.



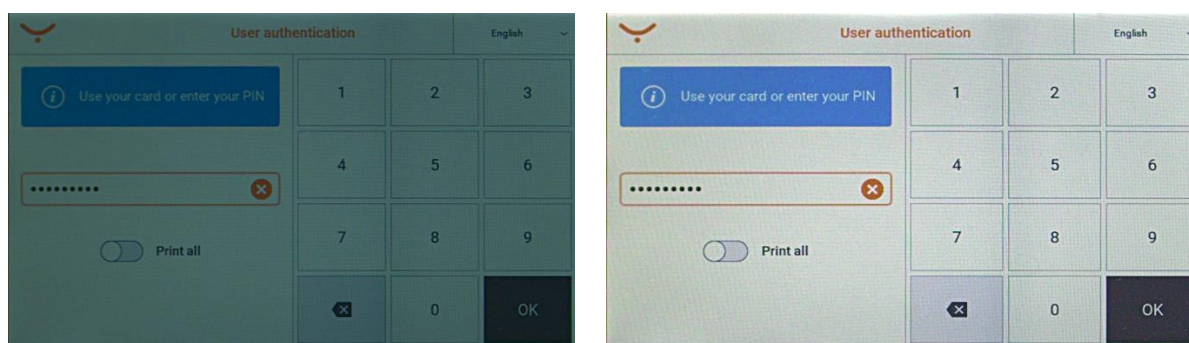
Obr. 1 – Mezikroky v algoritmu pro nalezení rohů displeje – vstupní obraz (vlevo nahoře), binarizovaný obraz (vpravo nahoře), segmentovaný obraz (vlevo dole), výsledek (vpravo dole)

### 1.1.2 Normalizace jasu

Tato část je důležitá, protože většina algoritmů je závislá na hodnotách jasu a pro lepší porovnání obrazovek je potřeba, aby měly podobný kontrast a tón barev. Ideální by bylo kalibrovat kameru pomocí referenční obrazovky, která by obsahovala saturevanou červenou, zelenou a modrou a alespoň 5 odstínů šedé. Taková obrazovka bohužel v aplikaci není k dispozici, proto se alespoň využívá faktu, že je v každé obrazovce bílá a černá. Jas se normalizuje po kanálech pomocí následujícího vzorce (1). Kde  $I_{orig}$  je původní obraz,  $I_{transf}$  je pak transformovaný.

$$I_{transf_{i,j}} = \frac{I_{orig_{i,j}} - \min(I_{orig})}{\max(I_{orig}) - \min(I_{orig})} \quad (1)$$

Tím, že se kontrast transformuje po kanálech, dosahuje se i částečného vyvážení bílé a redukuje se vliv barvy osvětlení.



Obr. 2 - Snímek obrazovky před (vlevo) a po (vpravo) transformaci kontrastu

Překážkou normalizace jasu jsou monotónní obrazovky, které značí načítání. Po tyto nemá normalizace smysl, proto se dopředu rozpoznají a žádná další analýza ani zpracování na nich neprobíhá. Kvůli občasnému výraznému moaré šumu a velkému množství různých podob monotónní obrazovek se není možné spolehnout jen na intenzitu pixelů v obraze nebo jejich barevnost. Pro rozpoznání monotónní obrazovky se tedy používá absolutní zastoupení hran v obraze a histogram intenzit pixelů po barevných kanálech. Hrany se získávají Cannyho hranovým detektorem[2].

U určení monotónnosti obrazovky bylo dosaženo stoprocentní úspěšnosti.







Obr. 3 – Příklady problematických obrazovek: a) jinak barevný okraj b) výrazný šum c) nízký jas (obrazovka je reálně bílá) d) nemonotónní obrazovka s výrazným zastoupením bílé, která musí pokračovat k analýze

## 1.2 Redukce šumu

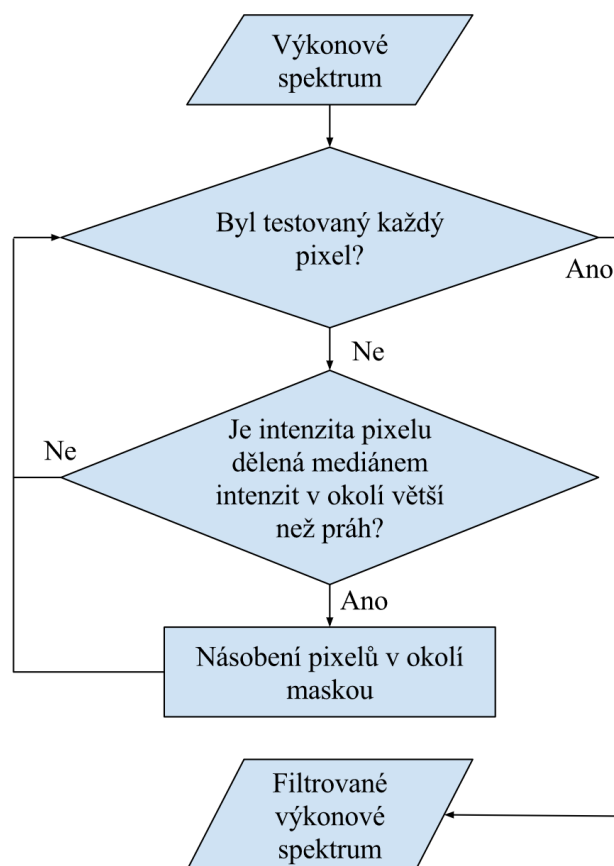
Při snímkování displeje je největším problémem šum typu moaré. Moaré vzniká v důsledku pravidelného uspořádání pixelů na displeji i na čipu kamery do matice. Tyto dvě mřížky spolu interferují, a tak vznikají pruhy. Vzhledem k výpočetní náročnosti odstranění tohoto typu šumu by bylo ideální použít hardwarový antialiasingový filtr, který ale nemáme momentálně k dispozici. Pro odstranění šumu se ukázala jako nejefektivnější filtrace ve spektru pásmovou Gaussovou zádrží. Filtrace byla provedena pro každý barevný kanál zvlášť. Pro vygenerování masky filtru byly použity následující vztahy:

$$G_{i,j}(u, v) = 1 - Ce^{-a(u^2 + v^2)} \quad (2)$$

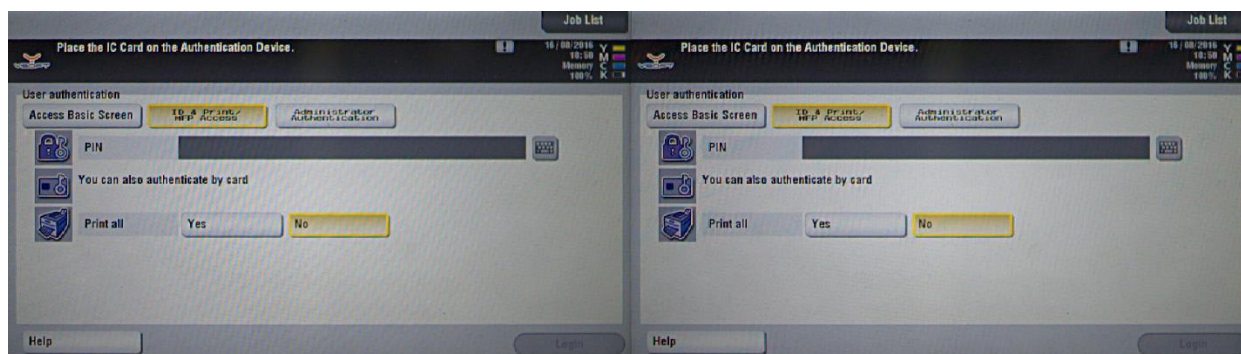
$$u = -\text{floor}\left(\frac{i}{2}\right), -\text{floor}\left(\frac{i}{2}\right) + 1, \dots, \text{floor}\left(\frac{i}{2}\right) - 1, \text{floor}\left(\frac{i}{2}\right) \quad (3)$$

$$v = -\text{floor}\left(\frac{j}{2}\right), -\text{floor}\left(\frac{j}{2}\right) + 1, \dots, \text{floor}\left(\frac{j}{2}\right) - 1, \text{floor}\left(\frac{j}{2}\right) \quad (4)$$

Po testování a optimalizaci vyšly nejlépe koeficienty  $C=1$ ,  $a=0,01$ , práh = 4, rozměr masky = 11x11. Kvalita filtrace byla hodnocena čistě subjektivně, vizuálním porovnáním filtrovaného a původního obrazu.



Obr. 4 – Schéma filtrace ve výkonovém spektru



Obr. 5 – Výsledek filtrace šumu typu moaré



Obr. 6 – Výsledek filtrace šumu typu moaré na monotónním obraze

## 2 DETEKCE OBRAZOVEK

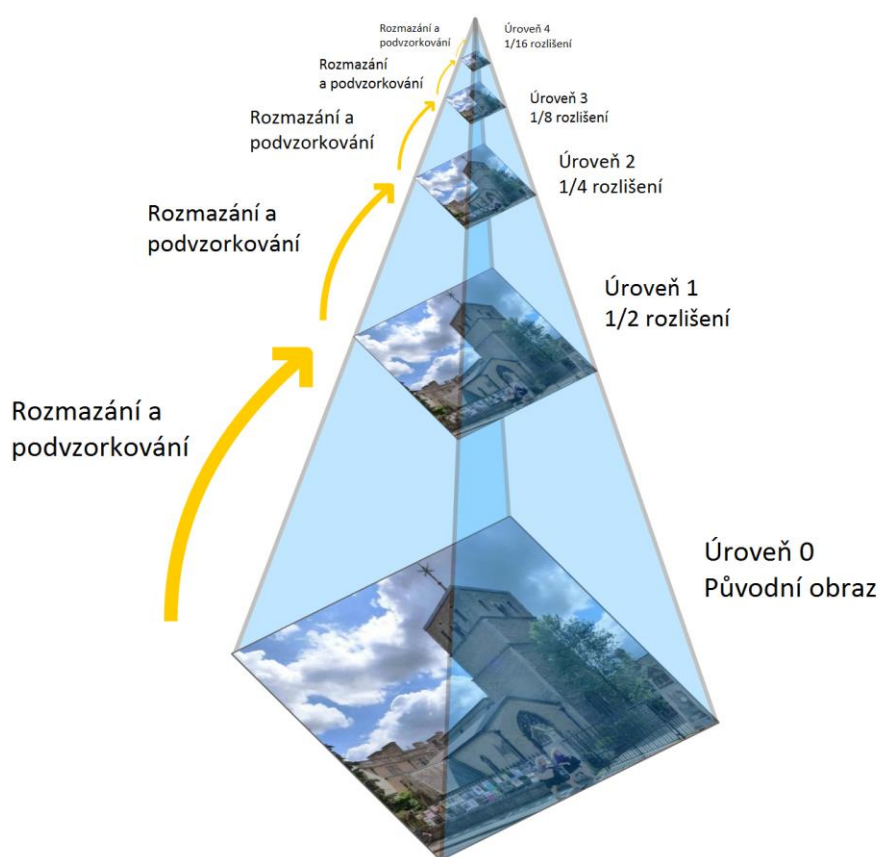
Algoritmus pro rozpoznání obrazovek na emdedded zařízení musí splňovat několik požadavků. Musí být dostatečně rychlý pro použití v reálném čase a úplně automatický, za předpokladu, že je k dispozici databáze srovnávacích dat. Musí být spolehlivý, protože na něm závisí funkce celé robotické ruky, a navíc jsou výsledky z běhu každého testování použity na hodnocení kvality produktu, který se testuje. Dalším důležitým kritériem jsou autorská práva. V této chvíli není rozpočet na zaplacení patentovaných algoritmů, proto se bude vybírat jen z těch, které jsou volně k použití. Z předzpracování vyplývá, že jsou obrazovky normalizovány, částečně zbaveny šumu a program je zkalibrován na jejich velikost, resp. souřadnice.

Pro detekci obrazovek jsem byly uvažovány nejdříve nízkoúrovňové metody, jako je porovnávání hranové reprezentace[2], 2D korelace[4], porovnávání histogramů[5], ale všechny tyto metody byly otestovány a trpěly značnou nespolehlivostí. Další metody jako vyhledávání kruhů a obdélníků pomocí Houghovy transformace [2], praahování a segmentace[5], vyhledávání kontur a segmentace [5] byly o poznání spolehlivější, ale vyžadovaly spoustu apriorní znalosti obrazovek. To znamená, že pokud by došlo k nějaké významnější změně vzhledu embedded aplikace, musely by se všechny vlastnosti znovu nadefinovat a prakticky by bylo potřeba znovu vyvinout celou aplikaci. Navíc jsou výše zmíněné metody velmi výpočetně náročné. Nakonec byla pro detekci a klasifikaci obrazovek zvolena třída metod, která používá deskriptory. Deskriptory se vyznačují svým obecným využitím, relativně rychle se dají vytvořit srovnávací data, a proto je jejich použití pro tuto aplikaci výhodné. Deskriptory jsou také způsobem, jak zobecnit aplikaci pro mnoho různých embedded zařízení, vždy stačí jen vybrat správnou množinu srovnávacích obrazů. Navíc jsou až o řád rychlejší na výpočet než výše zmíněné metody a poskytují dobrý poměr přesnost vs. výpočetní náročnost.

Klasifikace obrazovek obvykle probíhá v několika fázích. První fáze je detekce bodů zájmu. Důležitá je jejich opakovatelnost, která koreluje se spolehlivostí[6]. Opakovatelnost hodnotí, v jaké míře se detekují ty samé body v několika podobných snímcích jednoho objektu. Další důležitou vlastností je jedinečnost. Je nežádoucí, aby detektory našly několik bodů se stejnými vlastnostmi v okolí, protože je dojde k nežádoucím falešným shodám při přiřazování a srovnávání. Detektory bodů zájmu pracují na několika různých principech, které se mohou prolínat. První skupina pracuje s parciálními gradienty v okolí bodu jako např. Harrisův detektor[7], Lowova Aproximace Laplacianů Gaussianů[8], Moravcův detektor rohů[7], detektory založené na výpočtu determinantu Hessiánské matice[6] nebo detektory odvozené od zmíněných. Další skupinou jsou metody pracující na základě porovnávání intenzity centrálního pixelu a jeho okolí jako například SUSAN[9], FAST[10] nebo AGAST[11]. Třetí skupina využívá entropie v okolí pixelu a výpočtu vlastních čísel Hessiánské matice jako je Kadirův a Bradyův detektor asymetričnosti[12]. Dále je mnoho metod velmi podobných od zmíněných nebo které kombinují výše uvedené přístupy.

Po detekci bodů zájmu se výsledky zpřesňují a dále se vylučují nepravděpodobné body. To přispívá k robustnosti celého algoritmu. Vylučují se zejména body ležící v oblasti hran, které jsou nežádoucí, protože jsou málo jedinečné a zavádí falešně pozitivní shody při srovnávání[13]. Podle síly gradientu nebo podle rozdílů intenzit se může počítat skóre bodů a pro další zpracování vybrat jen ty nejlepší[14].

Způsobem, jak zajistit invariantnost deskriptoru vůči zvětšení, resp. zmenšení, je přiřazení měřítka ke každému bodu zájmu. Měřítka se obecně vyhodnocuje pomocí obrazové pyramidy. Jde o dekompozici na vrstvy střídavým rozmazáním Gaussovým filtrem a podvzorkováním[8]. V každé takové vrstvě se detekují body zájmu a podle toho, v které vrstvě se detekují, se k nim přiřadí odpovídající měřítko.



Obr. 7 – Obrazová pyramida

V obecných aplikacích počítačového vidění se také přiřazuje orientace každému bodu zájmu. V této práci a aktuální aplikaci je to přímo nežádoucí, a to z několika důvodů. Prvním je redundance. Takové vyhodnocení není potřeba, protože jsou všechny obrazy normalizované a stejně otočené. Výpočet orientace zvyšuje čas výpočtu, a naopak snižuje robustnost. V aplikaci se nachází hodně textu a rámečků. Pokud detektor najde levý dolní roh rámečku, pak v nerotovaném smyslu je jasné vidět jeho pozice a orientace. Na rozdíl od toho, pokud se hledá jakkoli orientovaný bod se stejnými vlastnostmi, pak se najdou zbylé tři rohy rámečku. Tím by

vznikaly falešně pozitivní shody. Z uvedených důvodů se budou vybírat deskriptory, které nejsou invariantní vůči otočení.

Dalším krokem je výpočet deskriptoru. Způsoby výpočtu jednotlivých deskriptorů jsou uvedeny dále v kapitole 2.2. Deskriptory mohou být binární nebo nebinární, s různou dimenzionalitou. Binární deskriptory mají velkou výhodu ve srovnávání deskriptoru aktuálního snímku s deskriptory snímků v databázi, protože se může využít Hammingova vzdálenost[15]. Výpočet Hammingovy vzdálenosti je velmi rychlý a také jednoduchý. Pro výpočet stačí binární operace XOR následovaná sumou bitů. Na rozdíl od toho, na srovnání nebinárních deskriptorů je potřeba složitějších a výpočetně náročnějších metod. Příkladem je výpočet Euklidovské vzdálenosti, následovaný Houghovou transformací pro detekci klastrů, které patří k jednomu objektu. Nakonec se výsledek verifikuje pomocí metody nejmenších čtverců kvůli konzistenci pozičních parametrů (více v [8]). Dimenzionalita deskriptoru prakticky znamená, jak dlouhý je vektor příznaků pro každý bod zájmu. Obecně platí, čím vyšší dimenzionalita, tím větší robustnost a přesnost, ale zároveň delší čas na výpočet a srovnání deskriptoru s deskriptory snímků v databázi.

Deskriptory s vyšší dimenzionalitou dosahují lepších výsledků, ale s počtem snímků v databázi se dramaticky zvyšuje čas pro srovnávání[16]. To se úspěšně řeší redukcí dimenzionality pomocí metod jako je PCA[17] a LDA[18] nebo hashovacích funkcí, jako například LSH[19]. Využití těchto metod není s této aplikací klíčové, zejména kvůli rozsahu databáze srovnávacích snímků (v našem případě desítky snímků).

## 2.1 Detekce zajímavých bodů v obraze

Budou otestovány dva detektory bodů zájmu v obraze. Oba fungují na velmi odlišném principu a budou reprezentovat dvě hlavní skupiny detektorů, ty, které fungují na základě gradientu v okolí bodu a ty, které jen srovnávají intenzity pixelů v okolí. Z těchto skupin byly vybrány právě Harrisův detektor a FAST, a to hlavně kvůli jejich hojnému využití v praxi a dobrým výsledkům ve spojení s vybranými deskriptory (více v kapitole 2.2).

### 2.1.1 Harrisův detektor rohů

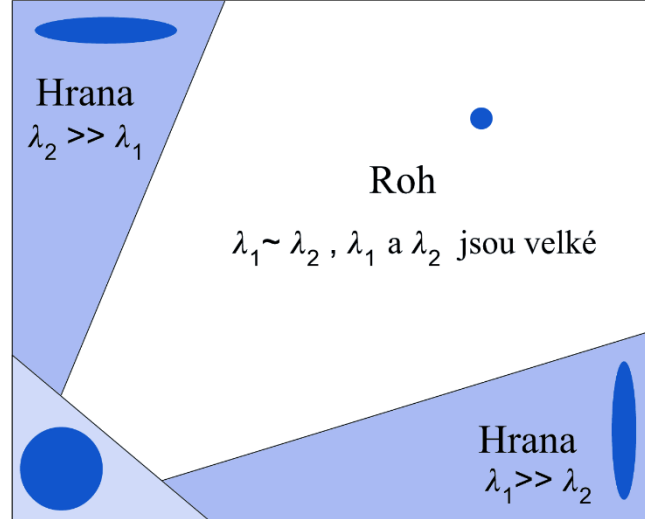
Harrisův detektor je už přes dvě desetiletí standardně používaným detektorem bodů zájmu, a to hlavně kvůli jeho robustnosti a opakovatelnosti. Naopak jeho nevýhodou je pomalý výpočet bodů a relativní citlivost na šum. Pro vyhodnocení bodů zájmu využívá parciální gradienty (7), (9) v okolí bodu váhované Gaussovým oknem  $w$  (6), (8), (10). K vyhodnocení vlastností v okolí bodu se využívá poměru vlastních hodnot matice  $M$  (5). Z Obr. 8 je jasné, že čím jsou obě vlastní čísla větší, tím je bod zájmu silnější.[7]

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (5)$$

$$A = X^2 \times w \quad (6) \quad X = Image \times [-1 \ 0 \ 1] \quad (7)$$

$$B = Y^2 \times w \quad (8) \quad Y = Image \times [-1 \ 0 \ 1]^T \quad (9)$$

$$C = (XY) \times w \quad (10)$$

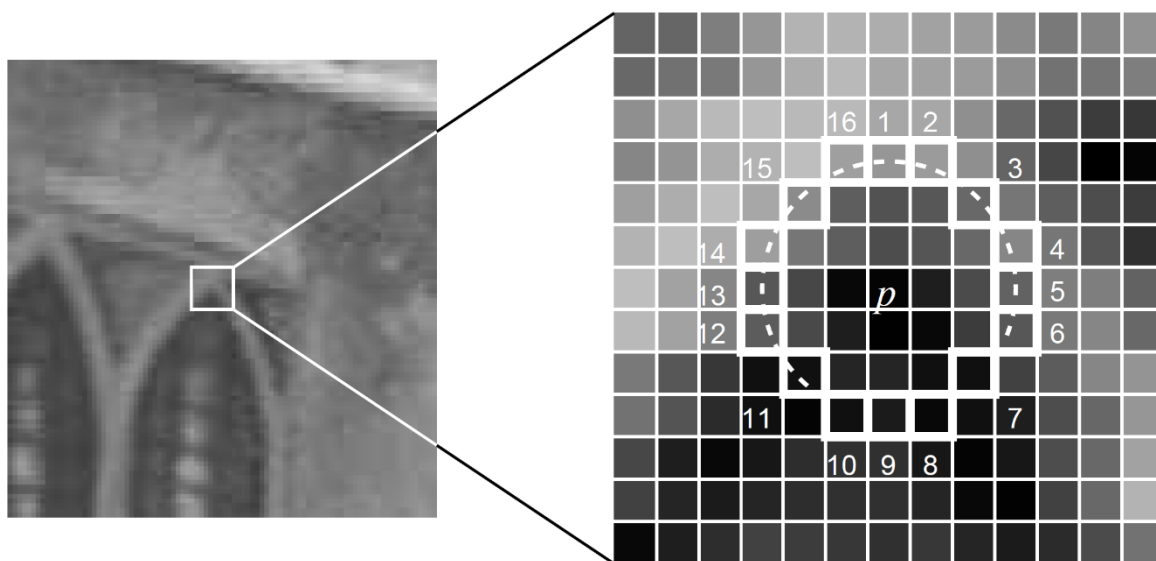


Obr. 8 – Reprezentace vlastností objektů v obraze na základě vlastních hodnot matice  $M$

### 2.1.2 FAST – Feature from Accelerated Segment Test

Feature from Accelerated Segment Test, dále jen FAST, je jeden z nejrychlejších detektorů bodů zájmu v dnešní době. FAST detektor prochází obraz pixel po pixelu a vyhodnocuje jeho  $N \times N$  okolí. Jeho základním principem je srovnávání jasu centrálního pixelu  $I_p$  s pixely, které leží ve vzdálenosti  $r$  kolem něj. Pro větší rychlost se nejdříve vyhodnotí čtyři pixely (nahore, dole, vlevo a vpravo). Pokud alespoň tři z nich mají hodnotu jasu vyšší než  $I_p + t$  nebo nižší než  $I_p - t$ , kde  $t$  je prahová hodnota, místo je vyhodnoceno jako potenciální příznak a postupuje se evaluací ostatních pixelů. Pokud se najde spojitý segment pixelů o dané délce  $m$ , které splňují podmínku  $I_p + t$  nebo  $I_p - t$ , pak je místo vyhodnoceno jako bod zájmu [10]. Podle délky segmentu a počtu pixelů v kruhu se jednotlivé modifikace FAST detektoru označují. Například FAST 9-16 a FAST 5-8 pro segment od délce 9 resp. 5 po sobě jdoucích pixelů z šestnácti, resp. osmi

pixelů.



Obr. 9 – Test pro detekci rohů v segment 12x12 pixelů. Zvýrazněné čtverce jsou pixely v určité vzdálenosti od středu potenciálního příznaku  $p$  a jsou používány pro jeho hodnocení. Část kružnice, která je naznačená čárkovaně, označuje pixely, které jsou alespoň o prahovou hodnotu jasnější než  $p$ . [10]

Výhodou tohoto detektoru je vysoká rychlost (až 20x rychlejší než Harrisův detektor). Nevýhodou je, že má vysokou odezvu kolem hran, která může být nežádoucí. Dále je detekce jednotlivých příznaků nezávislá, proto nezajišťuje uniformní pokrytí scény. Další nevýhodou je závislost na prahu. Při nastavení konstantního prahu pro celou sérii obrazů lze dojít k jinému počtu bodů vlivem rozdílného kontrastu snímků.

## 2.2 Deskriptory

### 2.2.1 SIFT – Scale Invariant Feature transform

Scale Invariant Feature transform, dále jen SIFT je jeden z nejvíce používaných deskriptorů a je považován za zlatý standard a základ těchto metod. Generování bodů zájmu probíhá na základě Harrisova detektoru rohů z vrstev obrazové pyramidy. Orientace ke každému bodu zájmu se přiřazuje podle histogramu zastoupení směrů gradientu v okolí bodu. Z toho vyplývá, že je to deskriptor, který je invariantní vůči otočení a škálování. Znamená to, že může srovnávat scény s řadou afiních transformací, změnou úhlu pohledu, osvětlení nebo se zašuměním.

SIFT je velmi přesný a robustní, a proto stále vede v aplikacích, kde není potřeba zpracování v reálném čase. Naopak jeho nevýhodou je vysoká dimenze a nebinarita. Je tudíž pomalý na výpočet a srovnávání, proto není vhodný pro aplikaci v této práci.

### 2.2.2 SURF – Speeded Up Robust Features[13]

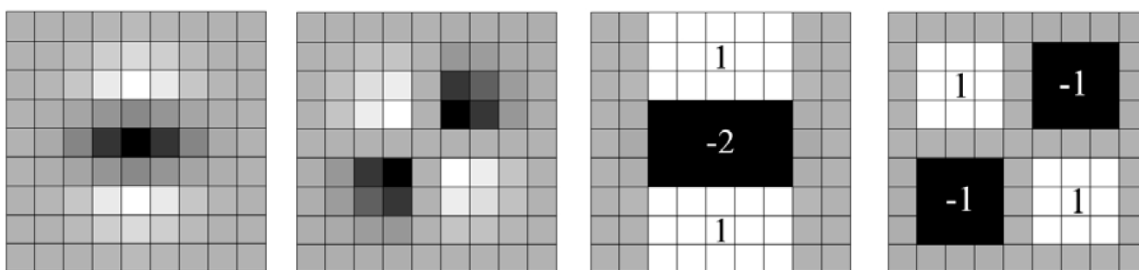
Speed Up Robust Features, dále jen SURF, je deskriptor, který je řádově rychlejší než SIFT, a proto se stal zlatým standardem pro aplikace, kde je nutné zpracování a srovnávání deskriptorů v reálném čase. Pro generování bodů zájmu se používá rychlý Hessiánský detektor. Myšlenka



je konvolovat druhou parciální derivaci 2D Gausiánské funkce s původním obrazem. Dále tyto parciální derivace poskládat do Hessiánské matice [20] a z té vypočítat determinant (11). To by bylo ale výpočetně náročné a objevoval se aliasing vlivem diskretizace a ořezání Gaussovy funkce. Proto se v praxi využívají krabicové filtry, které aproximují druhé parciální derivace Gaussovy funkce (Obr. 10). Jsou mnohem jednodušší a rychlejší na výpočet a neobjevují se u nich výše zmíněné chyby. Měřítkové prostory jsou implementovány jako obrazová pyramida. Také se může zvolit jiný přístup – místo podvzorkování se využije větší masky filtru a se zvětšováním masky se pak snižují frekvence vzorkování. K přesnějšímu lokalizování bodů zájmu se používá potlačení bodů, v kterých není maximum v okolí 3x3x3. Maxima determinantů Hessiánské matice jsou pak interpolované v měřítku a v obrazovém prostoru[21].

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (11)$$

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9 D_{xy})^2 \quad (12)$$



Obr. 10 – diskrétní a ořezaná druhá parciální derivace Gaussovy funkce  $D_{yy}$ , dále  $D_{xy}$ , krabicový filtr aproximující  $D_{yy}$ , krabicový filtr aproximující  $D_{xy}$

### Přiřazení orientace

Nejdříve se vypočítají odezvy na Haarovy vlnky[22] v ose x a y kolem bodů zájmu v okolí o průměru  $6s$ , kde  $s$  je měřítko, v kterém byl bod zájmu detekován. Krok se kterým se vzorkuje okolí je  $s$ , stejně tak odezvy Haarových vlněk jsou ve stejném měřítku. Délka jedné Haarovy vlnky je  $4s$ . Dále je výsledek vážený Gaussovým kruhovým oknem. Odezva je reprezentována dvěma vektory, jeden je síla odezvy v horizontální rovině, druhý je síla odezvy ve vertikální rovině. Dominantní orientace se počítá jako suma všech orientací kolem bodu zájmu v posouvajícím se okně o úhlu  $\pi/3$ . Horizontální a vertikální vektor jsou sečteny a vzniká nový vektor. Takovýto nejdelší vektor určuje orientaci bodu zájmu.

### Výpočet deskriptoru

První krok výpočtu deskriptoru je konstrukce čtverce, v jehož středu je bod zájmu a je stejně jako bod zájmu orientovaný. Velikost tohoto okna je  $20s$ . Tento region je rozdělený do menších regionů  $4 \times 4$ . Tohle rozdělení zachovává důležité informace o pozici. Pro každý z šestnácti subregionů se vypočítají příznaky a to následujícím způsobem. Vypočítají se odezvy na Haarovy vlnky v x a y rovině, které jsou relativně orientovány vůči bodu zájmu. Tyto odezvy



jsou sesumovány v každém regionu zvlášť a tím jsou vytvořeny 2x4 příznaků. Dále sesumujeme všechny absolutní hodnoty odezev v každém subregionu, to je dalších 2x4 příznaků, dohromady tedy 64 příznakový vektor. Vlnkové odezvy jsou invariantní vůči osvětlení. Pro invariantnost vůči měnícímu se kontrastu musíme vektor převést na jednotkový.

### ***U-SURF = Upright SURF***

U-SURF je modifikace SURF, která není invariantní vůči otočení, tudíž se dá použít s výhodou jen na některé aplikace. Například v naší práci, kdy k otočení nemůže nikdy dojít, protože je obraz vždy normalizovaný a natočený stejným směrem. Díky tomu, že se úplně vynechává informace o orientaci bodu zájmu, je tento deskriptor rychlejší a robustnější, protože nemá falešně pozitivní výsledky, které vznikly hledáním a srovnáváním deskriptorů ve všech směrech.

Podle literatury má podobné výsledky jako BRIEF a byl by vhodný pro aplikaci v této práci, ale protože je patentovaný, nebude dále implementován.

### **2.2.3 BRIEF – Binary Robust Independent Elementary Features**

Binary Robust Independent Elementary Features dále jen BRIEF, je efektivní deskriptor vlastností bodů. Dosahuje vysokých rychlostí výpočtu bez nutnosti učení a není invariantní vůči rotaci. Tato metoda pokrývá jen výpočet deskriptoru, proto společně s ní může použít jakýkoli detektor významných bodů.

Funguje na základě porovnávání pixelů v rámci oblasti zájmu, pokud je hodnota v bodě  $X(x,y)$  v integrálním obraze větší než v bodě  $Y(x,y)$ , pak funkce  $\tau$  nabývá hodnoty 1, pokud je menší, nabývá hodnoty 0 (13).

$$\tau(p; x, y) = \begin{cases} 1 & \text{pokud je } p(x) < p(y) \\ 0 & \text{jinak} \end{cases} \quad (13)$$

Toto se opakuje pro  $n_d$  dvojic pixelů. Běžně se používá 128, 256 nebo 512 dvojic. Přičemž, čím více dvojic, tím vyšší přesnost a větší výpočetní náročnost. Pomocí následující rovnice se vypočítá vektor příznaků.

$$f_{nd}(p) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p; x_i, y_i) \quad (14)$$

Dvojice pixelů se mohou vybírat několika různými způsoby, podle literatury je neúčinnější výběr nezávislých a identicky rozprostřených náhodných proměnných. Pravděpodobnost výběru bodu je váhovaná 2D Gaussovou funkcí s parametry  $(0, \frac{1}{25} S^2)$ . Tyto parametry se udávají v polárních souřadnicích, tzn., že největší pravděpodobnost výběru bodu bude ve středu oblasti zájmu a směrem od středu se bude snižovat až k  $\frac{1}{25}$  kvadrátu velikosti oblasti zájmu  $S$ . Druhý bod se vybírá stejným způsobem, ale střed Gaussovy váhovací funkce je v prvním vybraném bodu a poloměr se zvyšuje na  $\frac{1}{100}$  kvadrátu velikosti oblasti zájmu.  $(G(X, \frac{1}{100} S^2))$  [14].

Výběr bodů v každé oblasti zájmu musí být pro všechny srovnávané obrazy stejný, aby bylo možné je spolehlivě porovnat.

#### 2.2.4 ORB

ORB je vylepšená metoda založená na BRIEF, je rotačně invariantní a je odolnější vůči šumu. Právě tato metoda je využita ve stávajícím řešení rozpoznání obrazovek pro ovládání robotické ruky. ORB využívá kombinace Harrisova detektoru rohů a FAST, z kterých počítá skóre a vybírá jen nejrobustnější body zájmu. Další modifikací od BRIEF je výpočet orientace bodů zájmu pomocí centroidu intenzit pixelů, který zajišťuje rotační invariantnost[23]. Podle literatury dosahuje dobrých výsledků, ale i tak je pro rotační invariantnost pro aplikaci v této práci nevhodný.

#### 2.2.5 BRISK – Binary Robust Invariant Scalable Keypoints

Binary Robust Invariant Scalable Keypoints, dále jen BRISK, je deskriptor, který je invariantní vůči rotaci i změně měřítka. Podle literatury je přesný a rychlý, až o řád rychlejší než SURF. Jeho nerotovaná modifikace SU-BRISK má podobné výsledky jako BRIEF a proto bude vhodný pro tuto práci a porovnání s BRIEF[14].

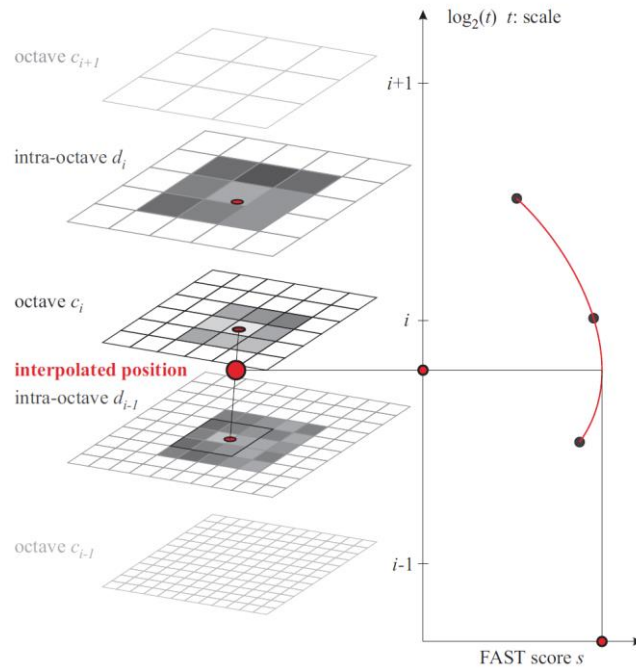
##### ***Detekce oblastí zájmu***

Pro hledání oblastí zájmu se může použít řada metod, protože výpočet deskriptoru není nijak závislý na detekci bodů zájmu. Autoři deskriptoru využívali s dobrými výsledky FAST[10], nebo jeho upravenou alternativu AGAST[11].

##### ***Přřazení měřítka***

Oblasti zájmu se detekují v několika oktávách obrazové pyramidy a ve vrstvách mezi nimi. Typicky se volí  $n = 4$  pro počet oktáv  $c_i$  a stejný počet vrstev  $d_i$  mezi oktávami. Přičemž první vrstva  $d_0$  vzniká převzorkováním 1,5krát. Nejdříve se pro každý bod zájmu počítá skóre  $s$ . Skóre označuje maximální práh FAST algoritmu, kdy je ještě pixel označen za bod zájmu. Z každé oblasti se vybere bod s nejvyšším skórem. Podmínkou je, že skóre ve vrstvách pod a nad danou vrstvou v obrazové pyramidě je nižší než  $s$ . Tímto způsobem se vyhodnotí všechny oblasti zájmu ve všech vrstvách obrazové pyramidy. Pro každý výsledný bod najdeme maximum skóre  $s$  pro vrstvu pod a nad vrstvou s aktuálním bodem. Tyto tři body proložíme kvadratickou funkcí

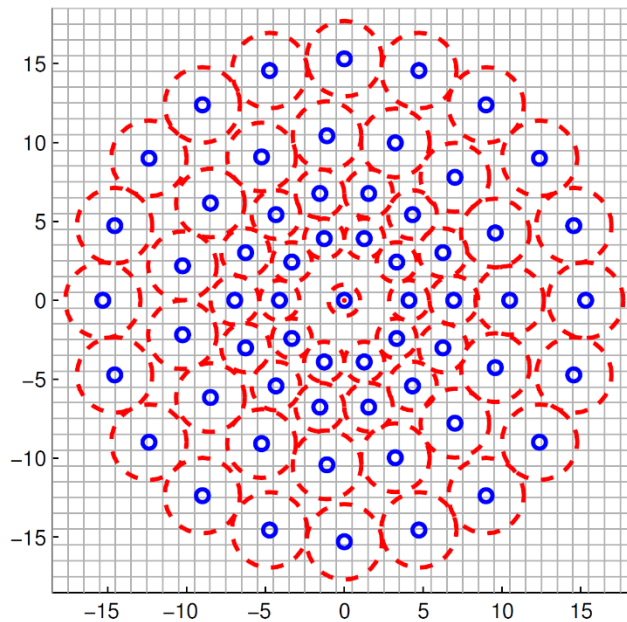
podél osy měřítka metodou nejmenších čtverců (Obr. 11). Podle maxima této paraboly se zjistí přesné měřítko bodu zájmu.



Obr. 11 – Interpolace bodu zájmu ve vrstvě  $c_i$  a maxim ve vrstvách  $d_i$  a  $d_{i-1}$  [24]

### Přiřazení orientace

Oblast kolem bodu zájmu se dále vzorkuje podle schéma na Obr. 12.



Obr. 12 – Schéma vzorkování, modré body naznačují vzorkovací pozice. Poloměr čárkovaných červených kruhů koresponduje se standardní odchylkou  $\sigma$  Gaussova vyhlazovacího filtru aplikovaného v bodech vzorkování [24]

Pomocí intenzit jasu  $I(p_i, \sigma_i)$  a  $I(p_j, \sigma_j)$  v bodech  $p_i$  a  $p_j$ , které byly vyhlazeny Gaussovým filtrem o  $\sigma_i$  a  $\sigma_j$  se určí lokální gradient  $g(p_i, p_j)$ :

$$g(p_i, p_j) = p_i - p_j * \frac{I(p_i, \sigma_i) - I(p_j, \sigma_j)}{|p_i - p_j|^2} \quad (15)$$

Páry bodů jsou vybírány náhodně. Množina párů se dělí na dvě podmnožiny podle vzdálenosti, kterou jsou od sebe body vzdáleny na  $S$  (short) a  $L$  (long). Směr bodu zájmu se určí aritmetickým průměrem gradientu podmnožiny s větší vzdáleností mezi body ( $L$ ).

### ***Vytvoření deskriptoru***

Pro vytvoření deskriptoru invariantního vůči rotaci a změny měřítka se znovu vzorkuje oblast kolem bodu zájmu. Používá se stejného schéma natočeného podél hlavního gradientu vypočítaného v předchozích krocích. Dále se vyhodnocují dvojice bodů, které patří do podmnožiny párů s menší vzdáleností od sebe. Každý pár bodů se porovná, pokud je  $I(p_i^\alpha, \sigma_i) < I(p_j^\alpha, \sigma_j)$ , zapíše se 1, jinak 0. Výsledný deskriptor je tedy přirozeně binární a jeho dimenzi určuje počet párů bodů v podmnožině  $S$ .

### 3 AKTUÁLNÍ STAV

Softwarová část funguje na Windows servrech a je připojená k databázi. V databázi se nachází referenční snímky obrazovek, zajímavé regiony a text obražený ve všech obrazovkách. Referenční obrazovky jsou uloženy ve stromové struktuře podle kontextu (verze SafeQ, výrobce a model tiskárny). Dále se tam nachází informace o pozici ovládacích prvků a na jakou obrazovku má aplikace přejít po stisknutí ovládacího prvku.

Samotné rozpoznávání obrazovek se děje ve třech fázích. Nejdříve se ověří, jestli se po stisku tlačítka objevila požadovaná obrazovka. Pro detekci příznaků se používá metoda FAST a pak jednoduché srovnání s požadovaným obrazem na základě procentuální shody detekovaných bodů. Pokud je výsledek nepřesvědčivý, nastoupí analýza pomocí deskriptorů. V současné době je implementovaná kombinace ORB a BRISK deskriptoru. Počítá se skóre pro každý deskriptor zvlášť a pak se výsledek průměruje. Pokud i deskriptory selžou, provádí se analýza textu v obraze pomocí OCR a výsledek se srovnává s databází textu na obrazovkách. Každá obrazovka má unikátní slovo, které se nachází jen na příslušné obrazovce a nikde jinde. Pokud se najde pomocí ORC unikátní slovo, pak je obrazovka detekovaná jistá, v opačném případě se počítá procentuální zastoupení slov.

## 4 ZÁVĚR

Předzpracování je z velké části hotové. V dalším semestru se chci zaměřit na extrakci textu z obrazu pro zvýšení rychlosti OCR[25].

Nastudovala jsem problematiku deskriptorů a na základě literatury jsem pro moji aplikaci vybrala BRIEF a UD-BRISK deskriptory. U-SUFT by měl mít podobné výsledky jako předchozí dva deskriptory, ale jelikož je patentovaný, budu dále implementovat jen BRIEF a UD-BRISK. Pro detekci bodů zájmu chci srovnat FAST a Harrisův dektor a vyberu ten, který bude mít lepší poměr robustnost vs. rychlost. Celou aplikaci pro zpracování a analýzy obrazu budu vyvíjet v jazyce C# za pomoci OpenCV knihovny, která funguje v .NET wrapperu OpenCVSharp.

V dalším semestru chci výše zmíněné deskriptory vyzkoušet v praxi, optimalizovat jejich parametry, srovnat jejich výpočetní náročnost a robustnost, popřípadě je zkusit zkombinovat.

## 5 SEZNAM LITERATURY

- [1] WALEK, Ing Petr, Ing Martin LAMOŠ a Jiří JAN. *Analýza biomedicínských obrazů počítačová cvičení FEKT VUT v Brně Auto ři textu* : nedatováno. ISBN 9788021447929.
- [2] JAN, Jiri. *Medical Image Processing , Reconstruction and Restoration Concepts and Methods* [online]. 2006. ISBN 9780824758493. Dostupné z: doi:10.1201/9781420030679
- [3] WEI, Zhouping, Jian WANG, Helen NICHOL, Sheldon WIEBE a Dean CHAPMAN. A median-Gaussian filtering framework for Moiré pattern noise removal from X-ray microscopy image. *Micron* [online]. 2012, **43**(2–3), 170–176. ISSN 09684328. Dostupné z: doi:10.1016/j.micron.2011.07.009
- [4] JAN, Jiří. *Číslicová Filtrace, Analýza a Restaurace Signálů*. 2002. ISBN 80-214-1558-4
- [5] SZELISKI, Richard. Computer Vision: Algorithms and Applications. *Computer* [online]. 2010, **5**, 832. ISSN 10636919. Dostupné z: doi:10.1007/978-1-84882-935-0
- [6] MIKOLAJCZYK, Krystian a Cordelia SCHMID. Scale & affine invariant interest point detectors. *International Journal of Computer Vision* [online]. 2004, **60**(1), 63–86. ISSN 09205691. Dostupné z: doi:10.1023/B:VISI.0000027790.02288.f2
- [7] HARRIS, Chris a Mike STEPHENS. A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference 1988* [online]. 1988, 147–151. ISSN 09639292. Dostupné z: doi:10.5244/C.2.23
- [8] LOWE, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*. 2004, **60**(2), 91–110.
- [9] SMITH, Sm a Jm BRADY. SUSAN—a new approach to low level image processing. *International journal of computer vision* [online]. 1997, **23**(1), 45–78. ISSN 1573-1405. Dostupné z: doi:10.1023/A:1007963824710
- [10] ROSTEN, Edward a Tom DRUMMOND. Machine learning for high-speed corner detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. 2006, **3951 LNCS**, 430–443. ISSN 16113349. Dostupné z: doi:10.1007/11744023\_34
- [11] MAIR, "Elmar, Gregory D HAGER, Darius BURSCHKA, Michael SUPPA a Gerhard HIRZINGER". Adaptive and Generic Corner Detection Based on the Accelerated Segment Test". „*European Conference on Computer Vision (ECCV'10)*". nedatováno, 1–14.
- [12] KADIR, T a M BRADY. Scale Saliency: a novel approach to salient feature and scale selection. *Visual Information Engineering* [online]. 2003, 25–28. ISSN 05379989. Dostupné z: doi:10.1049/cp:20030478
- [13] BAY, Herbert, Tinne TUYTELAARS a Luc VAN GOOL. SURF: Speeded up robust features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* [online]. 2006, **3951 LNCS**, 404–417. ISSN 03029743. Dostupné z: doi:10.1007/11744023\_32
- [14] CALONDER, Michael, Vincent LEPETIT, Christoph STRECHA a Pascal FUA. BRIEF: Binary robust independent elementary features. *Lecture Notes in Computer Science*

- (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*) [online]. 2010, **6314 LNCS(PART 4)**, 778–792. ISSN 03029743. Dostupné z: doi:10.1007/978-3-642-15561-1\_56
- [15] NOROUZI, Mohammad, David J Dj David J FLEET, Ruslan SALAKHUTDINOV a David M. BLEI. Hamming distance metric learning. *Advances in Neural Information Processing Systems* [online]. 2012, 1–9. ISSN 10495258. Dostupné z: <https://papers.nips.cc/paper/4808-hamming-distance-metric-learning.pdf>
  - [16] MIKOLAJCZYK, K., K. MIKOLAJCZYK, C. SCHMID a C. SCHMID. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2005, **27**(10), 1615–1630. ISSN 0162-8828. Dostupné z: doi:10.1109/TPAMI.2005.188
  - [17] MOORE, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* [online]. 1981, **26**(1), 17–32. ISSN 0018-9286. Dostupné z: doi:10.1109/TAC.1981.1102568
  - [18] BLEI, David M, Blei@cs Berkeley EDU, Andrew Y NG, Ang@cs Stanford EDU, Michael I JORDAN a Jordan@cs Berkeley EDU. Latent Dirichlet Allocation. *Journal of Machine Learning Research* [online]. 2003, **3**, 993–1022. ISSN 15324435. Dostupné z: doi:10.1162/jmlr.2003.3.4-5.993
  - [19] HASHING, Locality Sensitive. Locality-Sensitive Hashing. 2014, 1–6.
  - [20] HORN, Roger A a Charles R JOHNSON. *Matrix Analysis*: [online]. 2. vyd. Cambridge: Cambridge University Press, 2012. ISBN 9781139020411. Dostupné z: doi:10.1017/CBO9781139020411
  - [21] BROWN, Matthew a D LOWE. Invariant Features from Interest Point Groups. *British Machine Vision Conference, BMVC 2002* [online]. 2002, 656–665. Dostupné z: doi:10.1.1.1.8475
  - [22] EDWARDS, Tim. Discrete Wavelet Transforms: Theory and Implementation 1 Introduction 2 A Brief Discussion of Wavelets. *Computer*. 1991, (September), 1–27.
  - [23] RUBLEE, Ethan a Gary BRADSKI. ORB - an efficient alternative to SIFT or SURF [online]. 2011. ISSN 1550-5499. Dostupné z: doi:10.1109/ICCV.2011.6126544
  - [24] LEUTENEGGER, Stefan, Margarita CHLI a Roland Y SIEGWART. BRISK : Binary Robust Invariant Scalable Keypoints. nedatováno.
  - [25] MANTAS, J. An overview of character recognition methodologies. *Pattern Recognition* [online]. 1986, **19**(6), 425–430. ISSN 0031-3203. Dostupné z: doi:[http://dx.doi.org/10.1016/0031-3203\(86\)90040-3](http://dx.doi.org/10.1016/0031-3203(86)90040-3)



## **6 SEZNAM ZKRATEK A PŘÍLOH**

### **6.1 Seznam zkratk**

LoG – Laplacian of Gaussian

DoG – Difference of Gaussian

FAST – Feature from Accelerated Segment Test

SIFT – Scale Invariant Feature transform

SURF – Speeded-Up Robust Features

BRIEF – Binary Robust Independent Elementary Features

BRISK – Binary Robust Invariant Scalable Keypoints