# ABSTRAKT

# ABSTRACT

# KLÍČOVÁ SLOVA

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

# KEY WORDS

Descriptors, noise reduction, image enhancement, detection of reagions of interrest, 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. **x** s. Vedoucí bakalářské 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 S 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

Děkuji vedoucímu bakalářské práce Ing. Vratislavu Harabišovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé seminární práce.

V Brně dne 3. ledna 2017 ………………………….

podpis autora

# OBSAH

[ÚVOD 8](#_Toc469236434)

[1 PŘEDZPRACOVÁNÍ OBRAZU 9](#_Toc469236435)

[1.1 Normalizace obrazu 9](#_Toc469236436)

[1.1.1 Detekce rohů displeje 9](#_Toc469236437)

[1.1.2 Normalizace jasu 10](#_Toc469236438)

[1.2 Redukce šumu 11](#_Toc469236439)

[2 DETEKCE OBRAZOVEK 12](#_Toc469236440)

[2.1 Detekce zajímavých bodů v obraze 12](#_Toc469236441)

[2.1.1 Harissův detektor rohů 12](#_Toc469236442)

[2.1.2 Harris-Laplace and Hessia-Laplace 12](#_Toc469236443)

[2.1.3 Aproximace Laplacianů Gaussianů (LoG) pomocí rozdílu Gaussianů (DoG) 12](#_Toc469236444)

[2.1.4 Salient region Detector 12](#_Toc469236445)

[2.1.5 FAST – Feature from Accelerated Segment Test 12](#_Toc469236446)

[2.1.6 SUSAN 13](#_Toc469236447)

[2.2 Deskriptory 13](#_Toc469236448)

[2.2.1 SIFT – Scale Invariant Feature transform 13](#_Toc469236449)

[2.2.2 SURF – Speeded Up Robust Features[4] 16](#_Toc469236450)

[2.2.3 ORB 19](#_Toc469236451)

[2.2.4 BRIEF – Binary Robust Independent Elementary Features 19](#_Toc469236452)

[2.2.5 Další despriptory 20](#_Toc469236453)

[2.3 Srovnávání deskriptorů (matching) 20](#_Toc469236454)

[2.3.1 Klastrování pomocí Houghovy transformace 20](#_Toc469236455)

[2.3.2 Efektivní indexování nejbližšího souseda 20](#_Toc469236456)

[2.3.3 Vyhledávání podobných klíčových bodů (keypoint matching) 20](#_Toc469236457)

[3 AKTUÁLNÍ STAV 21](#_Toc469236458)

[4 NÁVRH ŘEŠENÍ A DALŠÍ POSTUP 22](#_Toc469236459)

[5 ZÁVĚR 23](#_Toc469236460)

[6 SEZNAM LITERATURY 24](#_Toc469236461)

[7 SEZNAM ZKRATEK A PŘÍLOH 25](#_Toc469236462)

[7.1 Seznam zkratek 25](#_Toc469236463)

[7.2 Seznam příloh 25](#_Toc469236464)

# SEZNAM OBRÁZKŮ

[Obr. 1 – Mezikroky v algoritmu pro nalezení rohů displeje 10](#_Toc469236029)

[Obr. 2 - Snímek obrazovky před (vlevo) a po(vpravo) transformaci kontrastu 11](#_Toc469236030)

[Obr. 3 – Výsledek filtrace šumu typu moiré 11](#_Toc469236031)

[Obr. 4 – Výsledek filtrace šumu typu moiré na monotónním obraze 11](#_Toc469236032)

[Obr. 5 – FAST 13](#_Toc469236033)

[Obr. 6 - Okolí bodu [3] 14](#_Toc469236034)

[Obr. 7 – Tvorba SIFT desktritoru [3] 16](#_Toc469236035)

[Obr. 8 – Aprofimace parciálních derivací krabicovými filtry 18](#_Toc469236036)

# 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 Research and Development 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 12, pro podchycení chyb vzniklých v průběhu vývoje. Toto je velmi náročné až nemožné splnit, kvůli časové náročnosti a počtu lidí, kterých by bylo potřeba. Navíc buildování se děje večer a nejlepší by bylo zjistit případné problémy už 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 testeři. 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 předzracování 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.

# PŘEDZPRACOVÁNÍ OBRAZU

## Normalizace obrazu

### 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 dipleji a určení polohy ovládácích 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 jsem naimplementovala automatickou detekci rohů displeje v obraze a podle pozice rohu se bude kamera kalibrovat v pravidelných intervalech.

Jednotlivé mezikroky zpracování je možné vidět na sérii obrázků Obr. 1. Nejprve se obraz převede na šedotónový, pak na birnární 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 segment 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. – Mezikroky v algoritmu pro nalezení rohů displeje – vstupní obraz (vlevo nahoře), binarizovaný obraz (vpravo nahoře), segmentovaný obraz (vlevo dole), výsledek (vpravo dole) | |

### Normalizace jasu

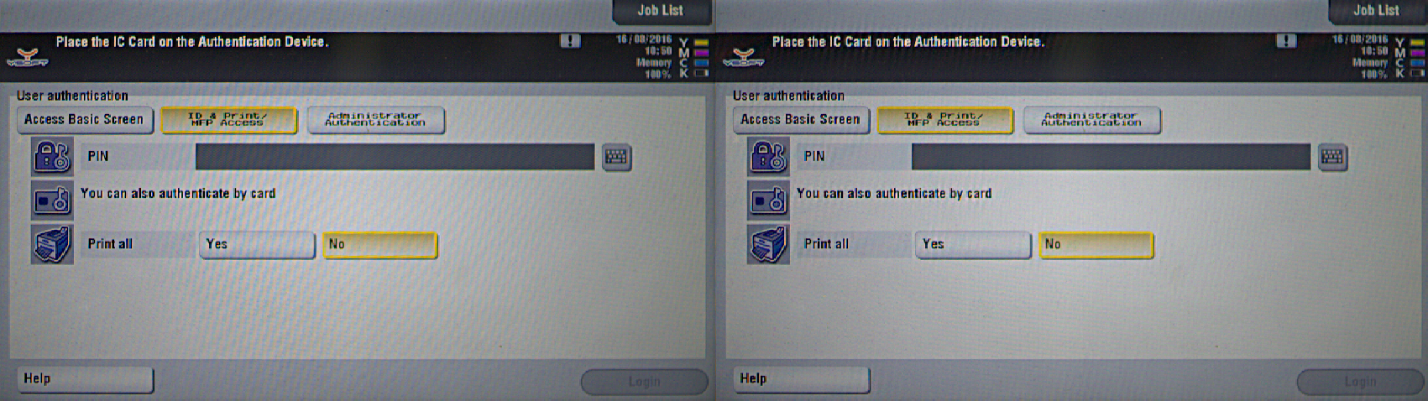
Tato část je důležitá, protože většina algoritmů je závislá na hodnotách jasu a pro lepší porovnání obrazovek je potřeba, aby měly podobný kontrast a tón barev. Ideální by bylo kalibrovat kameru pomocí referenční obrazovky, která by obsahovala saturovanou červenou, zelenou a modrou a alespoň 5 odstínů šedé. Takovou obrazovku bohužel v aplikaci nemáme, proto se aspoň využívá faktu, že je v každé obrazovce bílá a černá. Jas se normalizuje po kanálech pomocí následujícího vzorce[1]. Kde je původní obraz, je pak transformovaný.

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

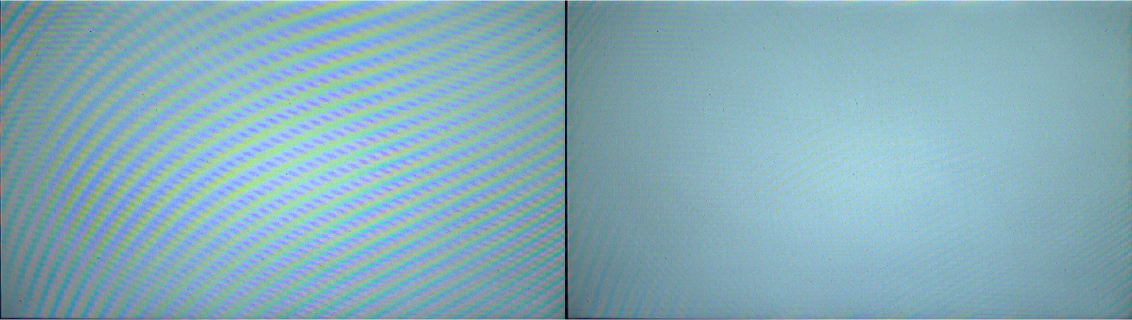
|  |  |
| --- | --- |
|  |  |
| Obr. - Snímek obrazovky před (vlevo) a po(vpravo) transformaci kontrastu | |

## Redukce šumu

* Mediánový filtr
* Šum typu moire
* Filtrace ve spektru Gaussovou úzkopásmovou zádrží
* Nejlépe by asi fungoval hardwarový antialiasingový filtr[2]



Obr. – Výsledek filtrace šumu typu moiré



Obr. – Výsledek filtrace šumu typu moiré na monotónním obraze

# DETEKCE OBRAZOVEK

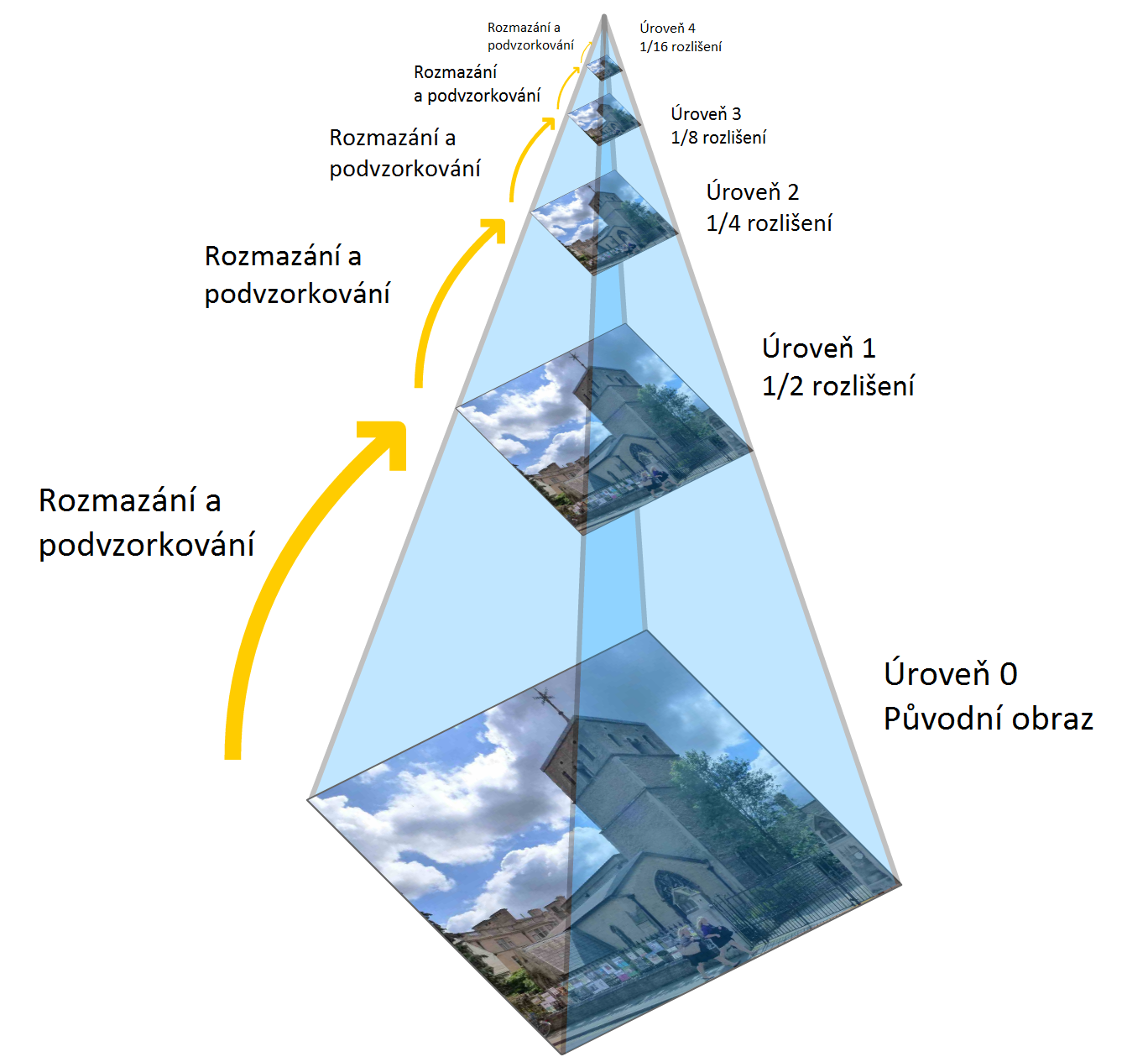
Algoritmus pro rozpoznání obrazovek na emdedded zařízení musí splňovat několik požadavků. Musí být dostatečně rychlý pro použití v reálném čase a úplně automatický, za předpokladu, že už máme k dispozici databázi srovnávacích dat. Musí být spolehlivý, protože na něm závisí funkce celé robotické ruky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

Pro detekci obrazovek jsem zvažovala nejdříve nízkoúrovňové metody, jako je porovnávání hranové reprezentace, 2D korelaci[J,J], porovnávání histogramů[Computer Vision], ale všechny tyto metody jsem otestovala a nebyly spolehlivé. Další metody jako vyhledávání kruhů a obdélníků pomocí Houghovy transformace [J.J], prahování a segmentace[Computer Vision], vyhledávání kontur a segmentace [Computer Vision] byly o poznání spolehlivější, ale vyžadovaly spoustu apriorní znalosti obrazovek. To znamená, že pokud by došlo k nějaké významější změně vzhledu embedded aplikace, museli bychom všechny vlastnosti znovu nadefinovat a prakticky znovu vyvinout celou aplikaci. Navíc jsou výše zmíněné metody velmi výpočetně náročné. Nakonec jsem pro detekci a klasifikaci obrazovek vybrala třídu 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 obyčejně probíhá v několika fázích. První fáze je detekce bodů zájmu. Důležitá je jejich opakovatelnost, která koreluje se spolehlivostí. 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, Lowova Aproximace Laplacianů Gausianů, Moravecův detektor rohů, detektory založené na výpočtu determinantu Hessiánské matice nebo detektory odvozené od zmíněných. Další skupinou jsou metody pracující na základě porovnávání intenzity centrálního pixelu a jeho okolí jako například SUSAN, FAST nebo AGAST. Třetí skupina využívá entropie v okolí pixelu a výpočtu vlastních čísel Hessiánské matice jako je Kadirův a Bradyův detektor asymetričnosti. 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. Podle síly gradientu nebo podle rozdílů intenzit se může počítat skóre bodů a pro další zpracování vybrat jen ty nejlepší.

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



V obecných aplikacích počítačového vidění se také přiřazuje orientace každému bodu zájmu. V naší aplikaci je to ale přímo nežádoucí, a to z několika důvodů. Prvním je redundance. Takové vyhodnocení nepotřebujeme, protože všechny obrazy máme 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 jasně vidět jeho pozice a orientace. Narozdíl 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 budeme vybírat deskriptory, které nejsou invariantní vůči otočení.

Dalším krokem je výpočet deskriptoru. Způsoby výpočtu jednotlivých deskriptorů jsou uvedeny dále v kapitole 2.2. Deskritory můžou být binární nebo nebinární, s různou dimenzionalitou. Birnární deskriptory mají velkou výhodu ve srovnávání deskriptoru aktuálního snímku s deskriptory snímků v databázi, protože se může využít Hammingova vzdálenost. Výpočet Hammingovy vzdálenosti je velmi rychlý a také jednoduchý. Pro výpočet stačí binární operace XOR následovaná sumou bitů. Na rozdíl od toho, na srovnání nebirnárních 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 [SIFT]). 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.

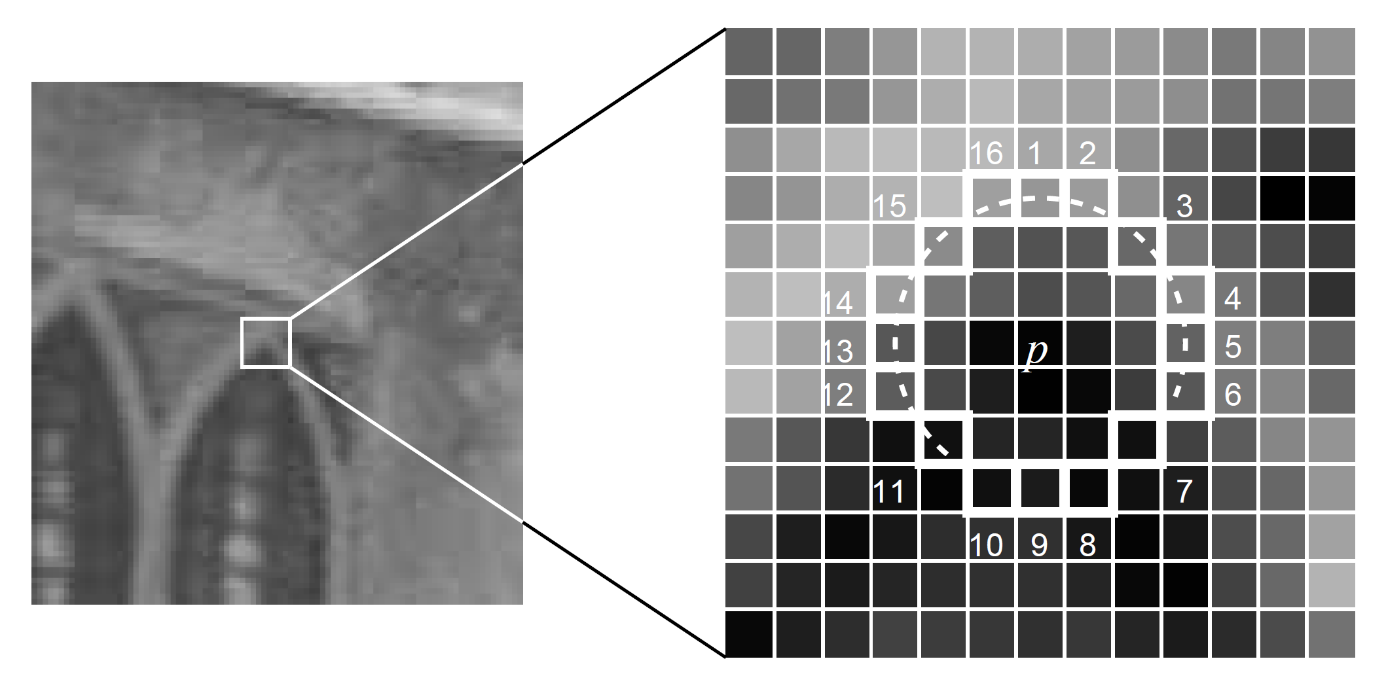
Deskritory s vyšší dimenzionalitou dosahují lepších výsledků, ale s počtem snímků v databázi se dramaticky zvyšuje čas pro srovnávání. To se úspěšně řeší redukcí dimensionality pomocí metod jako je PCA a LDA nebo hashovacích funkcí, jako například LSH. Využití těchto metod nepovažuji za klíčové, kvůli rozsahu databáze srovnávacích snímků (v našem případě desítky snímků).

## Detekce zajímavých bodů v obraze

### Harissův detektor rohů

* Není invariantní vůči změně měřítka
* pomalý
* W – váhová funkce – může být skoková (0 mimo okno, 1 vně okna) nebo Gaussova fce – větší důraz na střed okna
* Místa s velkým rozdílem intenzit v rámci místa zájmu (patch) má výsledné E velké – lčím větší E, tím lepší příznak
* Změna intensity pro posun (u,v)
* Prakticky začátek SIFT
* Citlivé na šum

### FAST – Feature from Accelerated Segment Test

Feature from Accelerated Segment Test, dále jen FAST, je jeden z nejrychlejších detektorů příznaků v dnešní době. FAST detektor prochází obraz pixel po pixelu a vyhodnocuje jeho NxN okolí. Jeho základním principem je srovnávání jasu centrálního pixelu Ip s pixely, které leží ve vzdálenosti *r* kolem něj. Pro větší rychlost se nejdříve vyhodnotí čtyři pixely (nahoře, dole, vlevo a vpravo). Pokud alespoň tři z nich mají hodnotu jasu vyšší než Ip+t nebo nižší než Ip-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 Ip+t nebo Ip-t, pak je místo vyhodnoceno jako příznak. Podle délky segmentu a počtu pixelů v kruhu se jednotlivé modifikace FAST dektoru 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. – 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. [3]

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 neýhodou je závislost na prahu. Při nastavení konstatního prahu pro celou sérii obrazů lze dojít k jinému počtu bodů vlivem rozdílného kontrastu snímků.

## Deskriptory

### SIFT – Scale Invariant Feature transform

* Jeden z nejvíce používaných, základ těchto metod
* Nevýhoda je vysoká dimenze dekriptoru
* nejpřesnější

Je to deskriptor, který je invariantní vůči otočení a škálování. To znamená, že může srovnávat scény s řadou affiních transformací, změnou úhlu pohledu, osvělení nebo se zašuměním.

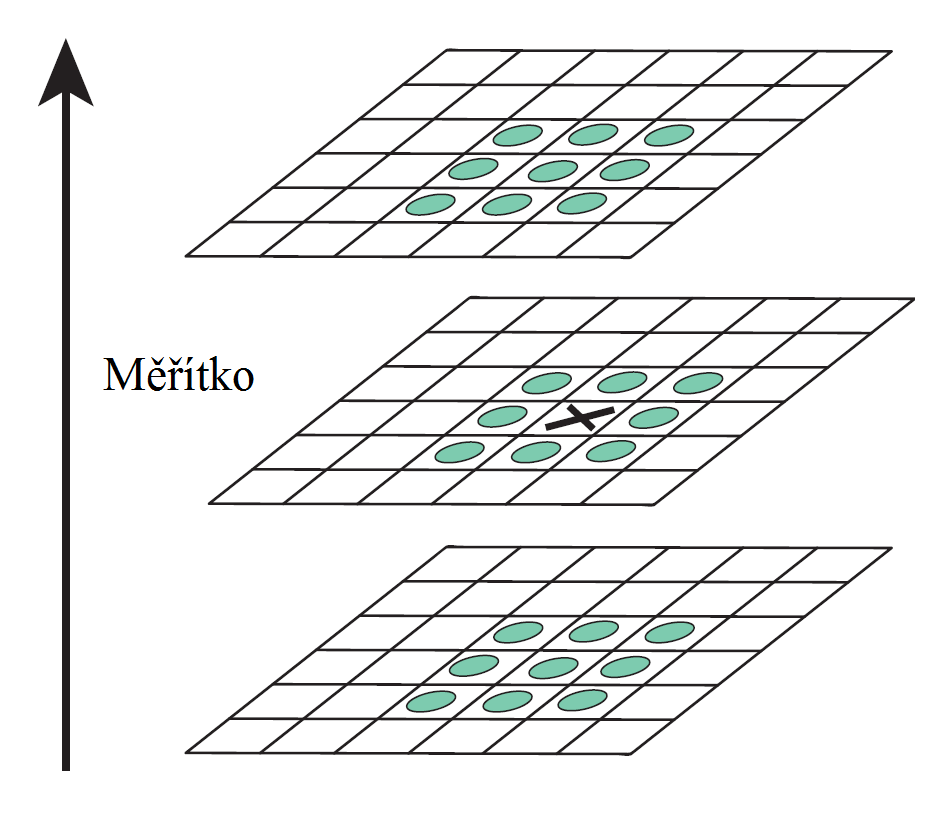
Generování příznaků probíhá v několika fázích:

#### Detekce místních extrémů

Nejdříve se vypočítá rozdíl dvou 2D Gaussových funkcí G(x,y,kσ)[4] s různým měřítkem *k* a získaný rozdíl se konvoluje originálním obrazem I(x,y). Tím vzniká rozdíl měřítkových prostorů původního obrazu D(x,y,σ).

|  |  |  |
| --- | --- | --- |
|  |  | () |

Tento postup se provádí pro několik různých měřítek, pak se obraz podvzorkuje a opakuje se stejný postup. Volba měřítka závisí našem uvážení. Menší měřítko zachytí více detailů a generuje více bodů zájmu, je to ale vykoupené vyšší výpočetní náročností.



Obr. - Okolí bodu [5]

Dalším krokem je srovnávání každého pixelu s okolím 3x3x3 tzn. s 26 sousedícími pixely. Maximum resp. minimum je deketováno, pokud má porovnávaný pixel nejvyšší resp. nejnižší hodnotu v daném okolí. Tyto body jsou dale vyhodnoceny jako spolehlivé příznaky.

#### Lokalizace bodů zájmu

Tyto příznaky se dále selektují pomocí rovnice, kde se počítá kontrast v bodě extrému a málo kontrastní body, tzn. se vyřazují.

|  |  |  |
| --- | --- | --- |
|  |  | () |

Pro zvýšení stability se vyřazují i body, které pochází z hran, protože jsou nestabilní vůči šumu. Takovéto body se vyznačují charakteristickým zakřivením, kdy mají vysoké zakřivení kolmo na hranu a podél hrany výrazně nižší.

Nejprve se vypočítá determinant a stopa 2x2 Hessiánu , která odpovídá pozici a měřítku bodu zájmu.

|  |  |  |
| --- | --- | --- |
|  |  | *(3)* |

Derivace jsou aproximovány rozdílem sousedících bodů a bodu zájmu. Podle nerovnice *(4)* se vyřadí všechny body zájmu, které mají poměr zakřivení v hlavních osách vyšší než r = 10.

|  |  |  |
| --- | --- | --- |
|  |  | *(4)* |

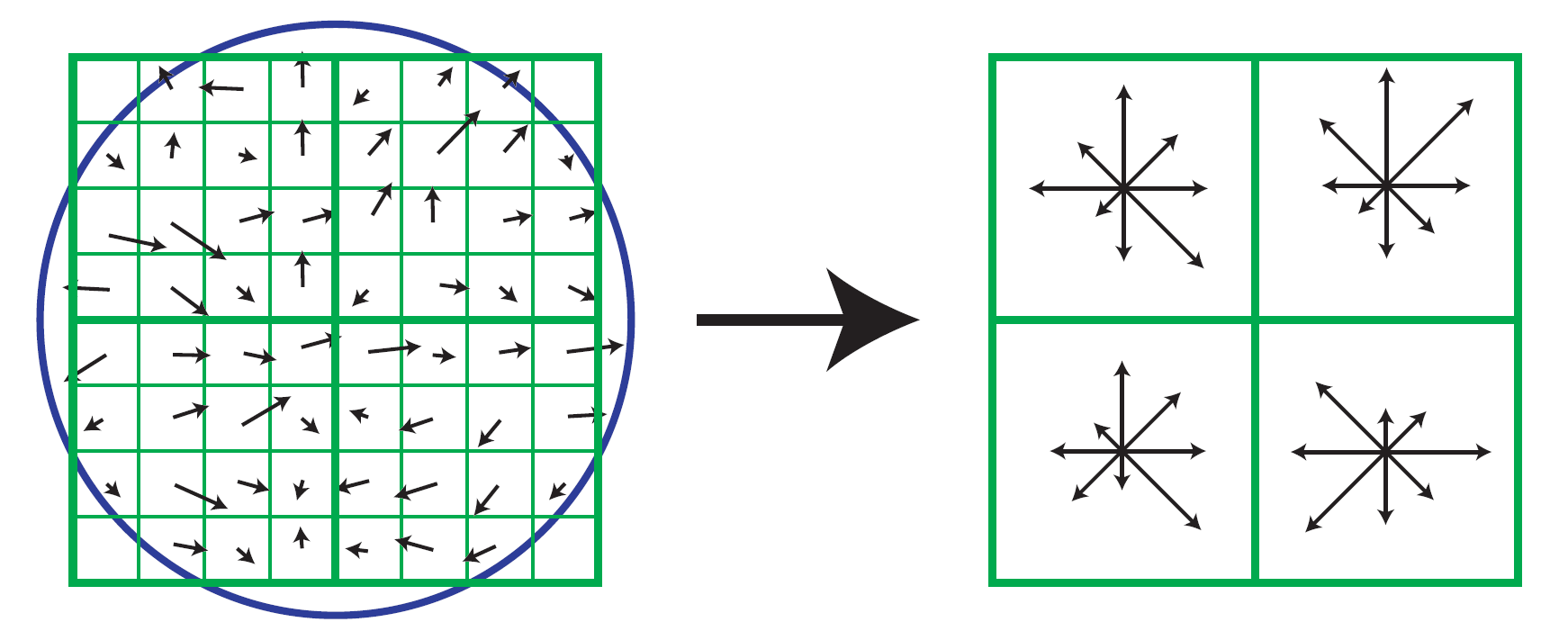
#### Přiřazení orientace

Dalším krokem je výpočet orientace, to zajišťuje invariantnost vůči otočení. Nevýhodou je vyřazení potenciálně hodnotných bodů zájmu, pokud jsou nekonzistentně otočené vůči zbytku obrazu. Pro každý pixel je vypočítána intenzita gradientu m(x,y) a jeho orientace podle následujících rovnic.

|  |  |  |
| --- | --- | --- |
|  |  | () |
|  |  | *(6)* |

Z orientací gradientů važených jejich intenzitou a kruhovým oknem váženým Gaussovskou funkcí je vypočítán histogram o 36 sloupcích, které odpovídají 360°. Píky v histogramu označují dominantní směry lokálních gradientů. Směr, který odpovídá maximu histogramu je označen jako hlavní směr bodu zájmu. Pokud se v histogramu nachází další píky, které jsou vysoké alespoň jako 80% maxima v histogramu, vytvoří se další body zájmu s danými orientacemi. Nakonec se na proloží 3 hodnoty v histogramu, které jsou nejblíže píkům k určení pozice píku pro lepší přesnost.

#### Výpočet deskriptorů



Obr. - Tvorba deskriptoru z navzorkovaného pole gradientů v bodě zájmu, délka šipek označuje velikost lokálního gradientu, modrý kruh napravo označuje váhování Gaussovým kruhovým oknem [5]

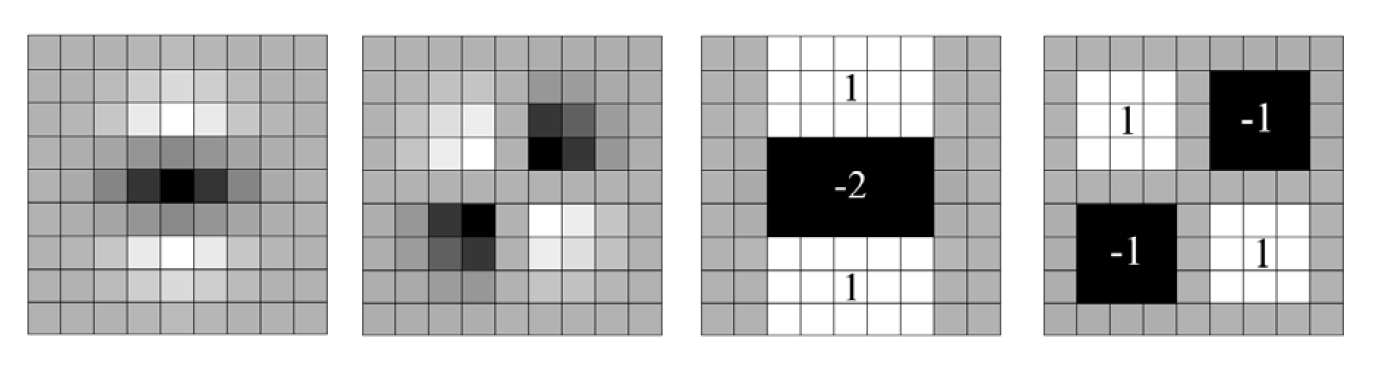
Intenzity a orientace gradientů v bodech zájmu se navzorkují a rozmažou Gaussovskou funkcí o stejném měřítku jako je měřítko bodu zájmu. Pro zachování invariance vůči rotaci, jsou souřadnice deskriptoru a směr gradientu otočeny relativně ke směru bodu zájmu. Tyto hodnoty jsou pro větší efektivitu počítány dopředu pro všechny úrovně pyramidy. Dále jsou váhované Gaussovským kruhovým oknem se σ, která se rovná 1,5násobkem šířky deskriptoru. Tímto zabráníme velkých změn deskriptoru při malé změně pozice, protože dáme větší důraz na gradient ve středu okna a menší důraz na hodnoty dále od centra deskriptoru. Podle [5] je nejlepší vzorkovat polem 4x4 v bodu zájmu a dále postupovat histogramem o 8 sloupcích. Vektor příznaků je tvořen hodnotami ve sloupcích všech histogramů, které korespondují s délkou šipek v  Obr. 2, tzn., že jde o vektor 4x4x8 = 128 hodnot pro každý bod zájmu. Nakonec je vektor normalizován, což zajistí invariantnost vůči osvětlení, protože při změně osvětlení bude přičtena nebo odečtena konstanta pro všechny hodnot vektoru, které nabudou stejných hodnot po normalizaci.

### SURF – Speeded Up Robust Features[6]

* Počítá se z integrálních obrázků. Detekce je podobná jako u SIFT, počítá se z DoG a nazývají ho Rychlý Hessian
* Deskriptor popisuje distribuci Haarových vlnek v okolí bodu zájmu
* Používá se 64 dimenzí – to redukuje čas pro srovnávání a hledání podobných deskriptorů
* Oproti SIFT je robustnější, protože používá jiné indexování

#### Rychlý Hessiánský detektor

* Myšlenka je konvolovat druhou parciální derivaci 2D Gaussiálnské funkce s původním obrazem. Dále tyto parciální derivace poskládat do Hessiánské matice [7] a z té vypočítat determinant. 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. Jsou mnohem jednodušší a rychlejší na výpočet a neobjevují se u nich výše zmíněné chyby. Měřítkové prostory jsou implementovány jako obrazová pyramida. Obrazy jsou opakovaně vyhlazeny Gaussovským filterem a podvzorkovány. Také se může zvolit jiný přístup – místo podvzorkování se využije větší masky filtru. (začíná se na 9x9 a postupně zvyšuje na 15x15, 21x21, 27x27 atd.). Stejně se zvětšováním masky se pak snižuj frekvence vzorkování. K lokalizování bodů zájmu se používá potlačení bodů, v kterých není maximum v okolí 3x3x3. Maxima determinantů Hessiánské matice jsou pak interpolována v měřítku a v obrazovém prostoru. (Brown)



Obr. – diskrétní a ořezaná druhá parciální derivace Gaussovy funkce Dyy, dále Dxy, krabicový filtr aproximující Dyy, krabicový filtr aproximující Dxy

#### Přiřazení orientace

Nejdříve se vypočítají odezvy na Haarovy vlnky[8] v ose x a y kolem bodů zájmu v okolí o průměru 6*s*, kde *s* je měřítko, v kterém byl bod zájmu detekován. Krok se kterým se vzorkuje okolí je s, stejně tak odezvy Haarových vlnek jsou ve stejném měřítku. Délka jedné Haarovy vlnky je 4*s.* 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 (přes souřadnice). Dominantní orientace se počítá jako suma všech orientaci kolem bodu zájmu v posouvajícím se okně o úhlu π/3. Horizontální a vektikální 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 20*s*. Tento region je rozdělený do menších regionů 4x4. Tohle rozdělení zachovává důležité informace o pozici. Pro každý subregion 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.

#### Modifikace tohoto deskriptoru

U-SURF = Upright SURF

* Který není invariantní vůči otočení, tudíž se dá použít s výhodou jen na některé aplikace. Například na tu naši, 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 srovnávání deskriptorů ve všech směrech.

SURF-128

* Je přesnější, ale taky pomalejší na výpočet a následné srovnávání se snímky v databázi, protože má vyšší dimenzi. Dosahuje se tím, že místo výpočtu sumy absolutních hodnot odezev na Haarovy vlnky se sesumuje d < 0 a d ≥ 0.

### ORB

* Založeno na hledání bodů zájmu pomocí FAST detektoru a vypočítání rotačně invariantního BRIEF deskriptoru
* Po výpočtu bodů se dále vypočítá orientace region zájmu pomocí centroidu intenzit
* Tento descriptor je aktuálně používaný v našem řešení
* Modifikace pro FAST:
* Tento detector má vysokou odezvu kolem hran, která je nežádoucí, proto se dále pokračuje Harissovým detektorem rohů a na základě těchto dvou metod se vybere nejlepších N bodů
* Pro invariantnost vůči měřítku používáme obrazovou pyramidu
* (poloměr 9 pixelů se ukázal jako nejoptimálnější)

### BRIEF – Binary Robust Independent Elementary Features

* BRIEF je efektivní deskritor vlastností bodů.
* Další metoda binarizace – Neighborhood Component Analysis – pomalejší
* BRIEF – přímo výpočet binárních slov, místo počítání dlouhých vektorů a dale zkracování
* Nemusí se trénovat
* Není invariantní vůči rotaci
* Nemusí dobře fungovat na velkých monotónních oblastech (test na Graffiti obrázku)
* Jen metoda pro tvorbu deskriptoru – může se zvolit řada detektorů příznaků
* Funguje na základě porovnávání pixelů v rámci oblasti zájmu (patch) , pokud hodnota v bodě X(x,y) v integrálním obraze větší než v bodě Y(x,y), pak funkce τ nabývá hodnoty 1, pokud je menší, nabývá hodnoty 0

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

* dvojice pixelů se mohou vybírat několika různými způsoby, podle článku (BRIEF)je neúčinější výběr iid(independent and identically distributed random viariables) z Gassianu(0, ) ( rozložení vážené Gaussovou funkcí a polárními souřadnicemi – centrum Gaussovy fce je v 0,0 a poloměr, z kterého se vybírá je 1/25\*S), kde S je velikost regionu zájmu
* druhý bod se vybírá iid z Gaussianu
* výběr bodů musí být pro všechny srovnávané obrazy stejný.

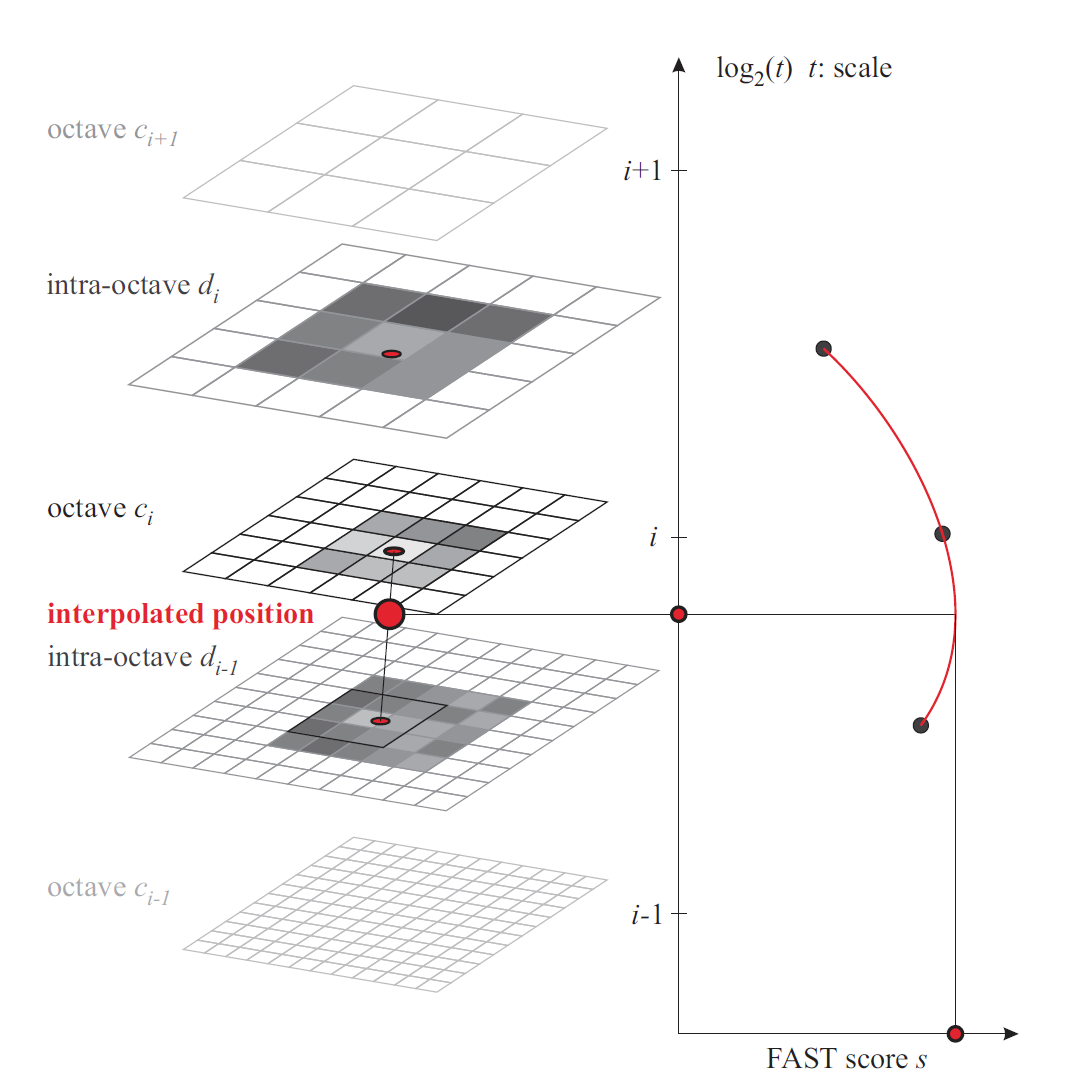
### BRISK

* Invariantní vůči rotaci i změně měřítka
* Přesný a rychlý
* BRISK detector se může používat v kombinaci s jakýmkoli deskriptorem
* Modifikace – nerotovaný SU-BRISK – ve srovnání velmi podobný performance jako BRIEF
* Řádově rychlejší než SURF

***Detekce oblastí zájmu***

Pro hledání oblastí zájmu se může použít řada metod, např. FAST, nebo jeho upravená alternativa AGAST.

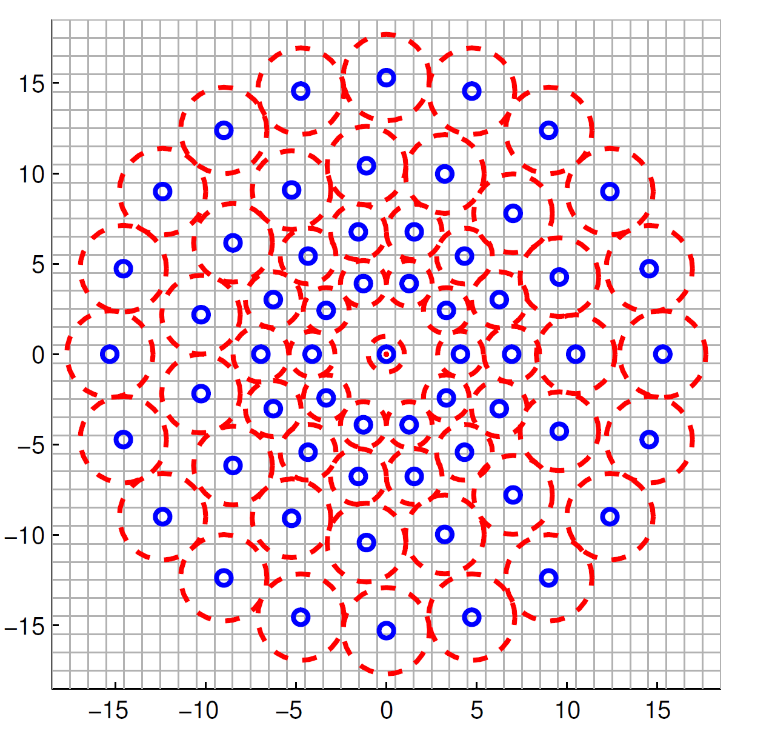
***Přířazení měřítka***

Oblasti zájmu se detekují v několika oktávách obrazové pyramidy a ve vrstvách mezi nimi. Typicky se volí n = 4 pro počet oktáv ci a stejný počet vrstev di mezi oktávami. Přičemž první vrstva d0 vzniká převzorkováním 1,5 krát. 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. Podmíkou je, že skóre ve vrstvách pod a nad danou vrstvou v obrazové pyramidě je nižsí něž *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. 9). Podle maxima této paraboly se zjistí přesné měřítko bodu zájmu. 

Obr. – Interpolace bodu zájmu ve vrstvě ci a maxim ve vrstvách di a di-1 [BRISK]

***Přiřazení orientace***

Oblast kolem bodu zájmu se dále vzorkuje podle schéma na Obr. 10.



Obr. – Schéma vzorkování, modré body naznačují vzorkovací pozice. Poloměr čárkovaných červených kruhů koresponduje se standartní odchylkou σ Gaussova vyhlazovacího filtru aplikovaného v bodech vzorkování

Pomocí intenzit jasu I(pi,σi) a I(pj,σj) v bodech pi a pj, které byly vyhlazeny Gausovým filtrem o σi a σj se určí lokální gradient g(pi, pj):

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 invariantnímu vůči rotaci a změny měřítka se znovu vzorkuje oblast kolem bodu zájmu. Používá se stejného schéma natočeného podél hlavního gradietu 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 , zapíše se 1, jinak 0. Výsledný deskriptor je tedy přirozeně binární a jeho dimenzi určuje počet párů bodů v podmnožině *S*.

## Přiřazování deskriptorů (matching)

### L2 metrika

### L1 metrika

# 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 uložené ve stromové struktuře podle kontextu (verze SafeQ, výrobce a model tiskárny). Dále se tam nachází informace o pozici ovládácích 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 descriptorů. V současné době je implementovaná kombinace ORB a BRISK deskriptoru. Počítá se skóre pro každý descriptor 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.

# ZÁVĚR

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

Natudovala 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 více robustní. Celou aplikaci pro zpracování a analýzy obrazu budu vyvíjet v jazyce C# za pomoci OpenCV knihovny, která funguje v .NET wrapperu OpenCVSharp.

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

# SEZNAM LITERATURY

[1] I. P. Walek, I. M. Lamoš, and J. Jan, *Analýza biomedicínských obrazů počítačová cvičení FEKT VUT v Brně Auto ři textu :* .

[2] X. Liu, “Analysis and Reduction of Moire Patterns in Scanned Halftone Pictures by Analysis and Reduction of Moire Patterns in Scanned,” *Science (80-. ).*, 1996.

[3] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 430–443, 2006.

[4] H. Guo, “A simple algorithm for fitting a gaussian function,” *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 134–137, 2011.

[5] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis:*, 2nd ed. Cambridge: Cambridge University Press, 2012.

[8] T. Edwards, “Discrete Wavelet Transforms : Theory and Implementation 1 Introduction 2 A Brief Discussion of Wavelets,” *Computer (Long. Beach. Calif).*, no. September, pp. 1–27, 1991.

# SEZNAM ZKRATEK A PŘÍLOH

## Seznam zkratek

LoG – Laplacian of Gaussian

DoG – Difference of Gaussian

SIFT

SURF - Speeded-Up Robust Features

## Seznam příloh