

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## DETEKCE VÝZNAČNÝCH BODŮ V OBRAZE

BAKALÁŘSKÁ PRÁCE

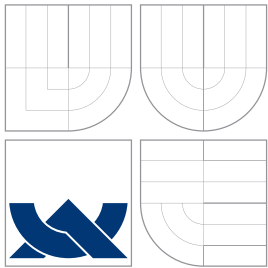
BACHELOR'S THESIS

AUTOR PRÁCE

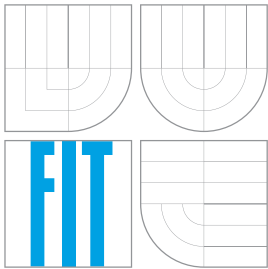
AUTHOR

ROMAN ČÍŽEK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# DETEKCE VÝZNAČNÝCH BODŮ V OBRAZE

INTEREST POINT DETECTION IN IMAGES

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ROMAN ČÍŽEK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. MICHAL ŠPANĚL

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Čížek Roman**

Obor: Informační technologie

Téma: **Detekce význačných bodů v obraze**

Kategorie: Počítačová grafika

Pokyny:

1. Prostudujte základy zpracování obrazu. Především se zaměřte na detekci hran a význačných bodů v obraze.
2. Zorientujte se v existujících metodách jejich detekce.
3. Vyberte vhodné metody a navrhnete jednoduchý detektor význačných bodů v obraze.
4. Experimentujte s vaší implementací.
5. Případně navrhnete vlastní modifikace metod.
6. Diskutujte dosažené výsledky a možnosti budoucího vývoje.
7. Vytvořte stručný plakát prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Španěl Michal, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2

L.S.  


---

doc. Dr. Ing. Pavel Zemčík  
vedoucí ústavu

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Roman Čížek**  
Id studenta: 79064  
Bytem: Mathonova 878/8a, 613 00 Brno  
Narozen: 08. 04. 1986, Brno  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Detekce význačných bodů v obraze  
Vedoucí/školitel VŠKP: Španěl Michal, Ing.  
Ústav: Ústav počítačové grafiky a multimédií  
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1  
elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel



.....

Autor

## **Abstrakt**

Tato práce se zabývá detekcí význačných bodů v obraze. Zaměřuje se především na metodu SIFT, kterou popisuje podrobněji. Obsahem je také implementace SIFT detektoru za pomoci knihovny OpenCV a zhodnocení jeho výsledků.

## **Klíčová slova**

zpracování obrazu, SIFT, deskriptor, scale-space, význačný bod

## **Abstract**

This thesis deals with interest point detection in image. Focus especially on method SIFT, which describes in more detail. Includes SIFT detector implementation with help of OpenCV library and evaluation its results.

## **Keywords**

image processing, SIFT, descriptor, scale-space, interest point

## **Citace**

Roman Čížek: Detekce význačných bodů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Detekce význačných bodů v obraze

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Roman Čížek  
14. května 2008

## Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Michalu Španělovi za jeho odbornou pomoc, kterou mi poskytoval během vytváření této práce.

© Roman Čížek, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Detektory význačných bodů</b>	<b>3</b>
2.1	Moravec detektor . . . . .	3
2.2	Harris a Stephens detektor . . . . .	4
2.3	SUSAN detektor . . . . .	4
2.4	SIFT detektor . . . . .	4
2.5	SURF detektor . . . . .	5
2.6	MSER detektor . . . . .	6
2.7	Význačný bod . . . . .	6
2.8	Deskriptor . . . . .	6
2.9	Scale space . . . . .	7
<b>3</b>	<b>SIFT (Scale Invariant Feature Transform)</b>	<b>8</b>
3.1	Princip . . . . .	8
3.2	Detekce lokálních extrémů ve Scale-space . . . . .	9
3.3	Lokalizace klíčových bodů . . . . .	9
3.4	Odstranění hranové odezvy . . . . .	10
3.5	Přiřazení orientace . . . . .	11
3.6	Vytvoření deskriptorů pro klíčové body . . . . .	12
<b>4</b>	<b>Návrh</b>	<b>13</b>
<b>5</b>	<b>Implementace</b>	<b>15</b>
5.1	sift.c . . . . .	15
5.2	detector.c . . . . .	16
5.3	Ovládání . . . . .	16
<b>6</b>	<b>Výsledky</b>	<b>17</b>
6.1	Rotace obrázku . . . . .	18
6.2	Změna osvětlení . . . . .	19
6.3	Změna velikosti . . . . .	19
6.4	Přidání šumu . . . . .	20
6.5	Změna pozorovací polohy . . . . .	21
6.6	Zhodnocení . . . . .	22
<b>7</b>	<b>Závěr</b>	<b>24</b>



# Kapitola 1

## Úvod

Detekce význačných bodů je součástí vědecké disciplíny, která se nazývá počítačové vidění. Detekce bodů bývá jednou z prvotních fází předzpracování obrazu pro další výpočty. Jelikož by bylo výpočetně náročné a také zbytečné počítat korespondence mezi všemi obrazovými body. Snahou je nalézt množinu stabilních bodů, které jsou odolné vůči změnám v obraze. Postupem času se objevovali různé způsoby detekce těchto rysů. Prozatím nemůžeme říci, že by existoval nějaký univerzální, nejlepší detektor. Každý detektor má nějaké své plusy a minusy, podle kterých musíme vybírat, jaký detektor je zrovna pro náš účel ten nejvhodnější.

Detekované významné body můžeme použít například pro hledání objektů ve scéně, sledování pohybu ve videosekvencích, konstrukci panoramat, sestavení 3D objektů z fotografií. Jaké jsou charakteristické vlastnosti význačných bodů se můžeme dočíst dále v sekci 2.7. Snahou je, napodobit vidění živých bytostí. Jelikož počítač dokáže snímat a zobrazit různé obrázky avšak vnímá je pouze jako změň čísel a nedokáže rozpoznat souvislosti, tak jako člověk. Proto jsou zkoumány různé způsoby výpočtů a různé pokusy o napodobení lidského vidění.

Cílem této práce je seznámení s metodami detekce význačných bodů v obraze. Hlavní část práce se bude zaměřovat na jednu z metod detekce nazvanou SIFT. Bude zde popsán její princip a implementace. Na závěr podrobíme metodu SIFT testům na ověření její stability.

V kapitole 2 nalezneme popis některých detektorů význačných bodů a jejich ukázky, jsou zde popsány některé pojmy, se kterými budeme dále pracovat. Kapitola 3 je zaměřena na podrobný popis metody SIFT. V další části 4 najdeme návrh, jak budeme postupovat při programování detektoru, v kapitole 5 následuje popis samotné implementace metody. V kapitole 6 najdete výsledky prováděných testů. V závěrečné kapitole 7 jsou zhodnoceny dosažené výsledky, přínos a možnosti budoucího rozšíření.

## Kapitola 2

# Detektory význačných bodů

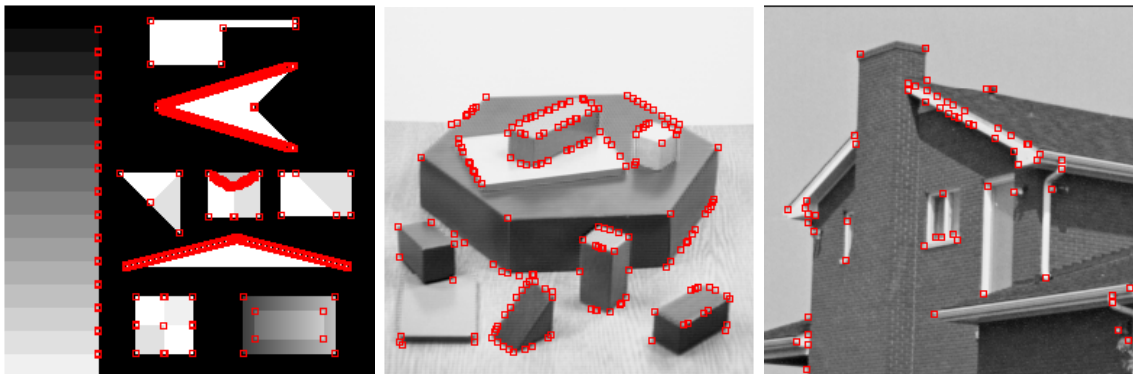
Existuje mnoho detektorů význačných bodů a různé jejich modifikace. Některé metody využívají nové, inovativní způsoby nalezení význačných bodů. Jiné se pouze snaží vylepšit stávající metody a zlepšit jejich vlastnosti. Na internetu můžeme najít i jednoduché online aplikace, které ukazují práci některých detektorů [13], [2].

Dále si uvedeme některé příklady známých detektorů a jejich ukázky detekce bodů.

### 2.1 Moravec detektor

Moravec detektor je jedním z prvních detektorů. Vynalezl ho Hanz P. Moravec v roce 1977 [12], kvůli výzkumu navigace robota. Detekce bodů je založena na porovnání lokálních bodů. Moravec nad zkoumaný bod umístil čtvercovou oblast, se kterou se dále pohybuje o jeden pixel v každém z osmi směrů. Význačné body získáme porovnáním směrových změn intenzity.

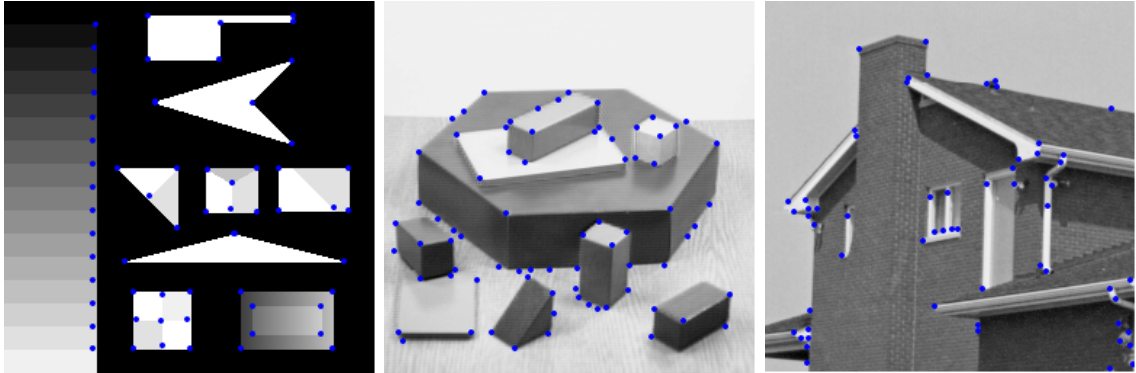
Jedinou výhodou je jeho jednoduchost a z toho plynoucí výpočetní nenáročnost. Mezi jeho nevýhody patří, že velice rád reaguje v okolí hran, je také náchylný na výskyt šumu v obraze.



Obrázek 2.1: Ukázka Moravcova detektoru [13]

## 2.2 Harris a Stephens detektor

Vyvinuli ho Chris Harris a Mike Stephens v roce 1988 [6]. Jedná se o vylepšený Moravec detektor. Používá místo čtvercové oblasti, oblast kruhovou, dosahuje tím lepších výsledků a větší odolnosti na detekci hran.

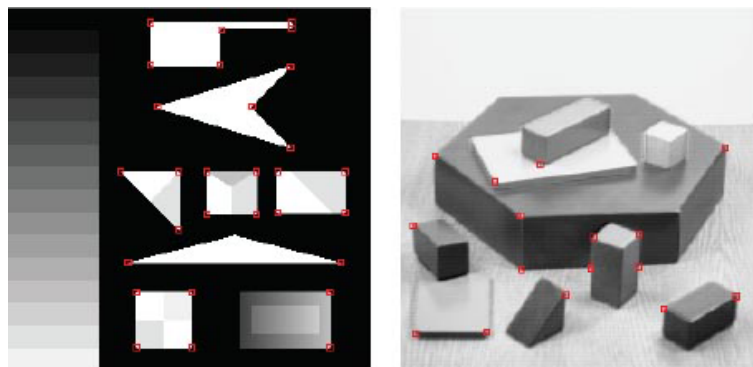


Obrázek 2.2: Ukázka Harris a Stephens detektoru [13]

Stal se velmi populárním pro svoji jednoduchost a rychlost při velmi dobrých výsledcích. Jeho nevýhodou je nestabilita při změně měřítka.

## 2.3 SUSAN detektor

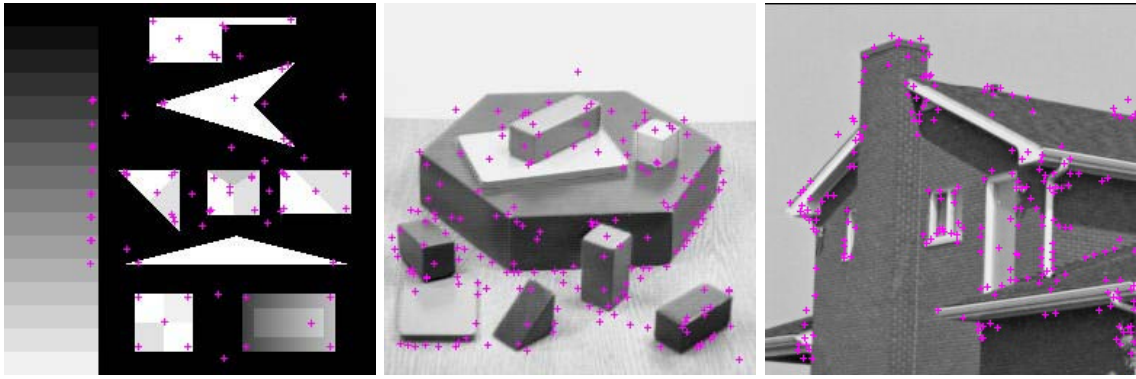
V roce 1995 vynalezli Smith a Brady SUSAN detektor [14]. Využívá odlišného přístupu k detekci bodů. Pracuje na porovnávání jasů bodu s hodnotami jasů okolních bodů v kruhovém okolí. Výhodou je rychlost detektoru, nevýhodou jeho malá stabilita.



Obrázek 2.3: Ukázka SUSAN detektoru [2]

## 2.4 SIFT detektor

SIFT neboli (Scale Invariant Feature Transform) je prvním z řady detektorů které se zaměřují na měřítkovou nezávislost. Představil ho v roce 1999 David Lowe [9]. Jeho princip bude podrobněji popsán v kapitole 3.



Obrázek 2.4: Vykreslení bodů za pomoci mé implementace SIFT detektoru

## 2.5 SURF detektor

Jedná se o jednu z novějších metod, autory jsou Bay, Tuytelaars a Van Gool. Představena byla v roce 2006 [1]. SURF v celém znění Speeded Up Robust Features. Snahou bylo urychlit výpočet význačných bodů na základě poznatků z již existujících metod detekce. Autoři se inspirovali převážně metodou SIFT, zjednodušili generování scale space. A také zmenšili počet binů u deskriptoru ovšem při zachování rozlišovací kvality deskriptoru SIFT. Podrobnější popis metody a její ukázkou implementace nalezneme [4].



Obrázek 2.5: Ukáзка detekce pomocí SURF detektoru [4]

Výhodou je rychlost detekce (přibližně 3x rychlejší, než SIFT), avšak není tak odolný vůči změně osvětlení a změně pozorovací polohy.

## 2.6 MSER detektor

Neboli, Maximally stable extremal regions [11]. Je založen na hledání maximálně stabilních oblastí, které slouží k nalezení korespondencí mezi obrazovými elementy ve dvou obrázcích pořizovaných z různých pohledů na scénu. Metoda čerpá z komplexních čísel korespondujících obrazových elementů, které přispívají k lepšímu porovnání.



Obrázek 2.6: Ukázka detekce pomocí MSER [11]

## 2.7 Význačný bod

Význačný bod [15] je bod v obraze, který může být charakterizován následujícími vlastnostmi:

- má jasnou a matematicky podloženou definici
- má přesně definovanou pozici v obrazovém prostoru
- lokální obrazová struktura okolo význačného bodu je bohatá na informace, které jsou později použity pro zpracování vizuálním systémem
- je stabilní vůči působení lokálních a globálních změn v obrazové doméně, jako jsou např. deformace, rotace, posuny, změny měřítka, změny osvětlení. Tak, aby mohl být význačný bod spolehlivě spočítán s vysokým stupněm opakovatelnosti.
- může zahrnovat vlastnosti o měřítku, k umožnění výpočtu význačných bodů z reálných obrázků i při změně měřítka.

## 2.8 Deskriptor

Deskriptor je struktura obsahující vlastnosti konkrétního bodu. Používá se k hledání korespondencí mezi nalezenými význačnými body. Na jeho kvalitě záleží zda budou určené

body opětovně nalezeny.

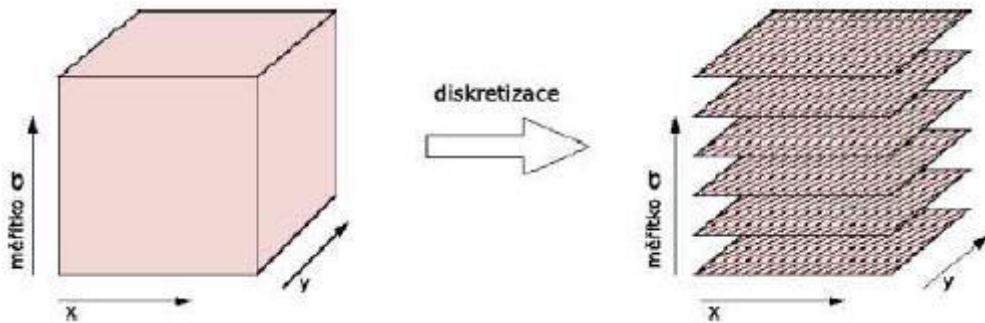
U metody SIFT se používá deskriptor tvořený vektorem o 128 prvcích tzv. binech. Je tvořen z histogramu orientací gradientů v okolí bodu. Při porovnávání významných bodů jsou porovnány euklidovské vzdálenosti deskriptorů. Čím nižší nalezneme vzdálenost, tím jsou si body podobnější.

## 2.9 Scale space

Scale space, neboli měřítkově nezávislé zpracování obrazu. Na toto téma probíhalo mnoho nezávislých výzkumů Lindeberg [7, 8], Witkin [17], Lowe [9]. K vytvoření této měřítkově nezávislé reprezentace se používá Gaussovy funkce.

Jedná se o 3D strukturu obrazu, kde  $x$  a  $y$  nám určuje velikost obrazu a  $\sigma$  určující měřítko. Scale space je spojitou funkcí, avšak pro uložení do počítače musíme využít diskretizace a rozdělit jej na vrstvy s určitým měřítkem.

Použitím Gaussova jádra z obrazu odstraňujeme oblasti s vyšší frekvencí, přicházíme tím o detaily. Můžeme si poté dovolit podvzorkovat další obraz na poloviční velikost bez ztráty důležitých informací. Scale space se dělí do oktáv, v každé oktávě mají vrstvy stejnou velikost, ale rozdílné měřítko. Nejvyšší vrstva oktávy má dvojnásobně větší měřítko, než vrstva nejmenší. Použitím rozdělení do oktáv zvyšujeme rychlost výpočtu.



Obrázek 2.7: Scale space [4]

## Kapitola 3

# SIFT (Scale Invariant Feature Transform)

Tato kapitola se podrobněji věnuje metodě SIFT, která je hlavním tématem této práce. Metoda byla pojmenována Scale Invariant Feature Transform (transformace měřítkově nezávislých rysů) zkráceně SIFT, protože transformuje obrazová data do měřítkově nezávislých souřadnic relativně k lokálním rysům. Rozebereme si zde její princip. Metoda SIFT patří mezi novější metody detekce význačných bodů. V roce 1999 David Lowe prezentoval svoji metodu na konferenci [9]. Později v roce 2004 se objevuje článek [10] o této metodě a jejím použití pro detekci objektů v obraze. Je zde detailně popsán postup metody a Lowe také zveřejnil ukázkovou implementaci.

### 3.1 Princip

Výpočet význačných bodů je rozdělen do 4 kroků, které jsou seřazeny od nejméně výpočetně náročných, až po nejsložitější. Výpočet se skládá z těchto kroků:

1. **Detekce lokálních extrémů ve Scale-space:** První fází je sestavení měřítkově nezávislé reprezentace obrázku tzv. scale-space. Pro efektivní výpočet scale-space se používá rozdíl Gaussových funkcí (DoG). Poté ve scale-space hledáme přes všechny měřítka a stupně lokální extrémy, které jsou invariantní k měřítku.
2. **Lokalizace klíčových bodů:** Získané body jsou dále detailněji zkoumány. Pomocí interpolace upřesníme polohu bodů ve zkoumaném prostoru. Klíčové body jsou vybrány na základě jejich stability.
3. **Přiřazení orientace:** Pro každý bod je přiřazena jedna nebo více orientací na základě orientací gradientů v okolí bodu. Přiřazením orientace docílíme nezávislosti bodu na rotaci.
4. **Vytvoření deskriptorů pro klíčové body:** Pomocí gradientů orientací v okolí bodu je vypočten deskriptor význačného bodu. Porovnáním deskriptorů můžeme hledat shody ve zkoumaných obrazech.

Nyní si každou fází podrobněji rozebereme.



## 3.2 Detekce lokálních extrémů ve Scale-space

Prvím krokem bude nalézt oblasti a měřítka, která mohou být opakovaně přiřazeny pro různá zobrazení stejného objektu. K detekci bodů, které budou invariantní na změnu měřítka obrazu, můžeme úspěšně použít hledání stabilních rysů skrz všechna možná měřítka. Použijeme k tomu spojitou funkci měřítka známou jako scale space [17].

Obrázek ve scale-space je definován jako  $L(x, y, \sigma)$ , k jeho vytvoření použijeme konvoluci Gaussovy funkce  $G(x, y, \sigma)$  se vstupním obrázkem,  $I(x, y)$ :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

kde  $*$  značí konvoluci v  $x$  a  $y$ , a

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.2)$$

K efektivní detekci stabilních klíčových lokací ve scale space, navrhnul Lowe [9] počítat extrémy za pomoci rozdílu Gaussových funkcí (DoG) a jejich konvolucí s původním obrázkem. Rozdílu dosáhneme odečtením dvou nezávislých Gaussových funkcí s rozdílným měřítkem.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.3)$$

Rozíl Gaussových funkcí (DoG) je aproximací Laplacianu Gaussovy funkce (LoG), která je dle prováděných výzkumů Lindeberg [7, 8], Lowe [9] nejlepší pro sestavení scale space. LoG je ovšem velice výpočetně náročné a tudíž není příliš vhodné pro detekci bodů v reálném čase. Zatímco pomocí DoG dosáhneme dobrých výsledků v rozumném čase. Její výpočtová jednoduchost spočívá v pouhém odečítání dvou obrázků.

Ukázka sestavení scale space je znázorněna na obrázku 3.1

Pro sestavení scale space musíme znát kolik budeme potřebovat vrstev pro oktávu. Při generování mnoha vrstev se nám zvýší výpočetní nároky. Lowe experimentálním zkoumáním došel k závěru, že maximální počet stabilních struktur dávají 3 vzorky na oktávu a tento počet použil ve svém detektoru. Použitím více vrstev nedosáhneme lepších výsledků, nalezneme sice více struktur, avšak tyto už nejsou, tak stabilní vůči různým změnám v obraze.

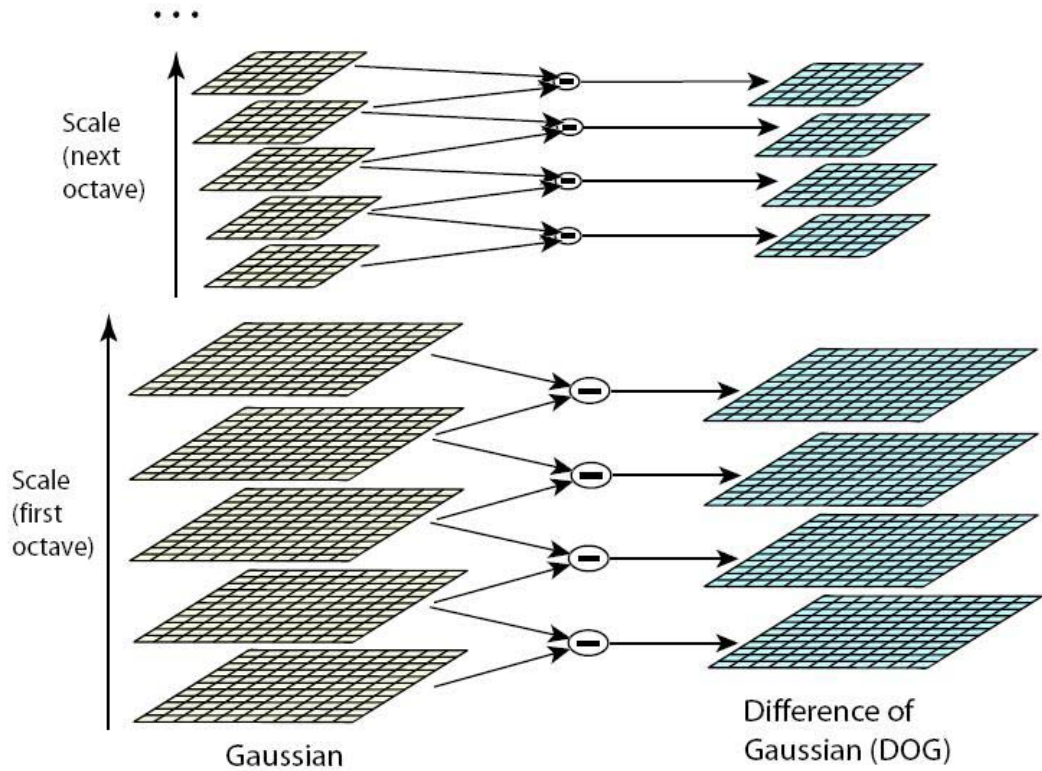
Jakmile máme vygenerováno měřítkově nezávislé znázornění obrazu, můžeme hledat klíčové body. Budeme hledat lokální extrémy. Projdeme všechny body napříč celým scale-space a budeme každý bod porovnávat s jeho nejbližším okolím. Bod tedy porovnáme s jeho 26 sousedy viz. obázek 3.2, kde osm sousedních bodů leží na stejné vrstvě jako zkoumaný bod, dalších devět bodů se nachází na nižší vrstvě a také devět bodů na vyšší vrstvě. Pro další zkoumání vybereme pouze body, které jsou lokálním maximem a tudíž jsou větší, než všechny sousední body a nebo jsou lokálním minimem a tím pádem menší, než všechny sousedící body. Porovnání probíhá většinou velice rychle, jelikož většina bodů je z dalších výpočtů vyřazena velice brzy, již po pár porovnáních.

## 3.3 Lokalizace klíčových bodů

Dalším krokem je upřesnění pozice bodu v souřadnicích a měřítku. V původní implementaci Lowe používal původní polohu nalezeného bodu. Ovšem poté s Brownem [3] ojevil metodu pro upřesnění polohy bodu v prostoru i v měřítku. Jeho experimenty ukazují, že poskytuje značné zlepšení k rozpoznání a stabilitě. Jeho metoda využívá Taylorův rozvoj

$$D(x) = D + \frac{\partial D^T}{\partial \mathbf{x}} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (3.4)$$





Obrázek 3.1: Ukázka konstrukce Scale-space, nalevo máme scale space sestrojený pomocí Gaussových funkcí, jejich rozdílem dosáhneme požadovaného DoG

Zderivováním funkce podle  $\mathbf{x}$  a položením rovno nule dosáhneme upřesněnou polohu bodu. Pozice extrému je určena offsetem  $\hat{\mathbf{x}}$  vzhledem k původnímu umístění bodu.

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (3.5)$$

Jestliže je  $\hat{\mathbf{x}}$  offset větší, než 0,5 v kterémkoli směru, znamená to, že extrém leží blíže k vedlejšímu bodu. V tomto případě původní bod nahradíme tímto bodem a provedeme výpočet znovu. Pokud dostaneme offset menší, přičteme jej k poloze bodu, abychom ho tím upřesnili.

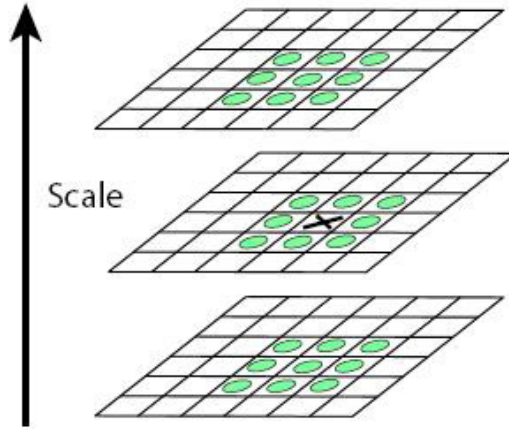
Funkční hodnota extrému  $D(\hat{\mathbf{x}})$  je užitečná k odstranění nestabilních bodů s malým kontrastem. Substitucí dostaneme tento vzorec

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad (3.6)$$

Low z dalších výpočtů odstraní všechny body s kontrastem menším jak 0,03.

### 3.4 Odstranění hranové odezvy

Použitím rozdílu Gaussových funkcí dostaneme mnoho bodů okolo hran. Avšak tyto body nejsou příliš stabilní, proto je potřeba jejich odstranění. Využijeme zde druhých derivací. Hlavní zakřivení můžeme spočítat pomocí 2x2 matice nazvané Hessian,  $\mathbf{H}$ .



Obrázek 3.2: Ukázka okolí bodu, se kterým je porovnán pro získání lokálních extrémů

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.7)$$

Body mají velkou hlavní křivost podél hrany, ale malé zakřivení v kolmém směru. Počítání hlavní křivosti je ovšem náročné, můžeme se vyhnout dlouhým výpočtům, tím že se zaměříme na jejich poměr. Vezmeme-li  $\alpha$  jako největší velikost křivosti a  $\beta$  nejmenší. Pak můžeme spočítat součet hodnot a výsledek výpočtem determinantu.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.8)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.9)$$

V případě, že vyjde determinant záporný, znamená to, že zakřivení mají různá znaménka a bod je odstraněn z dalších výpočtů. Nechť  $r$  je poměr mezi největší a nejmenší hodnotou, tak  $\alpha = r\beta$ . Potom poměr

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (3.10)$$

bude záležet na podílu vlastních hodnot, což je výhodnější, než výpočet s přesnými hodnotami. Velikost poměru je nejmenší, jestliže jsou dvě vlastní hodnoty stejné a roste s velikostí  $r$ . Potom nám stačí jednoduché porovnání

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r} \quad (3.11)$$

Lowe experimentálně stanovil hodnotu  $r = 10$ .

### 3.5 Přiřazení orientace

V tomto kroku budeme přidělovat detekovaným bodům jednu nebo více orientací. Tímto dosáhneme nezávislosti na rotaci. Podle měřítka klíčového bodu vybereme Gaussův vyhlazený obraz, který je nejbližší měřítku a tím dosáhneme měřítkové nezávislosti. Pro každý

bod  $L(x, y)$  předpočítáme velikost gradientu  $m(x, y)$  a orientaci  $\theta(x, y)$  za pomoci rozdílu pixelů.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (3.12)$$

$$\Theta(x, y) = \tan^{-1} \left( \frac{(L(x, y + 1) - L(x, y - 1))}{(L(x + 1, y) - L(x - 1, y))} \right) \quad (3.13)$$

Sestavíme histogram orientací gradientu zkoumaného bodu na základě vypočtených gradientů v jeho okolí. Histogram obsahuje 36 binů, které pokrývají 360 stupňů. Velikost gradientu je váhově upravena pomocí kruhového okna Gaussova rozložení vah s 1,5 násobkem  $\sigma$  daným pro aktuální vrstvu.

V histogramu poté hledáme maximální hodnotu orientace. Pokud nalezneme ještě orientaci o velikosti 80% maximální, vytvoříme nový bod s totožnými souřadnicemi a přiřadíme mu tuto orientaci. Těchto bodů se v obraze příliš nevyskytuje, avšak poskytnou nám výrazné zlepšení stability při hledání korespondencí.

### 3.6 Vytvoření deskriptorů pro klíčové body

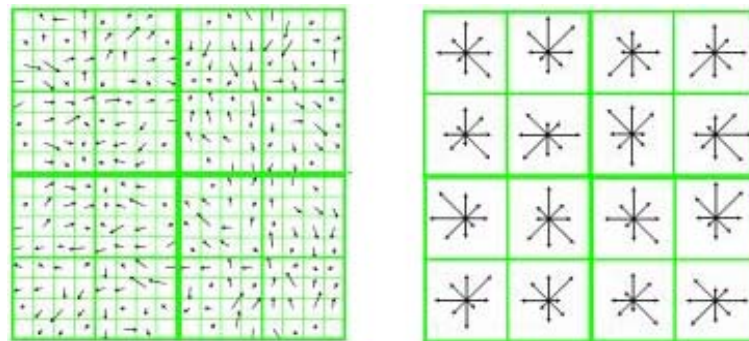
Dalším krokem bude sestavit deskriptor pro význačný bod. Deskriptor obsahuje informace o okolí bodu a využívá se k porovnávání bodů z více obrázků. Je tedy nutné, aby byl stabilní i na další změny jako například změna osvětlení, či polohy pozorovacího bodu.

Metoda SIFT využívá deskriptory na bázi modelu biologického vidění. Výzkum v tomto směru prováděli Edelman, Intrator a Poggio [5]. Základem je orientace gradientů v okolí bodu, nezávisí na přesné lokalizaci, umožňuje i malé posuvy těchto gradientů.

Před samotným výpočtem gradientů a následných histogramů, natočíme zkoumanou oblast podle orientace bodu, tím dosáhneme nezávislosti na rotaci.

Jsou zde využívány deskriptory o 128 binech, tedy deskriptor má 128 hodnot podle kterých je pak následně porovnáván k zjištění shody bodů. Okolí bodu je rozděleno na 4x4 oblasti, v každé oblasti se spočítá histogram orientací podle orientací gradientů v dané lokalitě. Tímto dostáváme 16 histogramů a každý histogram obsahuje 8 binů. Celkem tedy dostáváme deskriptor o 128 binech.

Ke stabilitě deskriptorů vůči změnám osvětlení využijeme normalizaci deskriptoru. Po této úpravě dosáhneme nezávislosti na změně kontrastu v obraze.



Obrázek 3.3: Vlevo vidíme orientace gradientů pro okolní body. Napravo pak máme oblast rozdělenou na 4x4 podoblastí, v každé je sestaven histogram orientací. Všechny tyto histogramy, pak tvoří deskriptor význačného bodu.

# Kapitola 4

## Návrh

Cílem bude naimplementovat detektor význačných bodů za pomoci metody SIFT, jejíž princip byl popsán v přechozí kapitole. Programová část bude v jazyce C, snahou bude docílit přenositelného kódu jak pro windows, tak linux. K implementaci bude použita knihovna OpenCV [16]. Jedná se o knihovnu od firmy Intel, tato knihovna obsahuje množství funkcí a struktur pro práci s obrazovými daty.

Bude se jednat o jednoduchou konzolovou aplikaci, kterou uživatel bude moci ovládat zadáním parametrů při spuštění aplikace. Jaké parametry budou potřeba zadat se uživatel dozví pomocí výpisu nápovědy při spuštění s parametrem -h, parametry budou také upřesněny v další kapitole.

Program nalezne význačné body, které budou uloženy ve struktuře k tomu určené. Po nadetekování budou zaznačeny do obrázku a vykresleny na obrazovku uživateli.

Program bude naimplementován podle postupu v předchozí kapitole, tedy nejdříve se budou provádět jednodušší výpočty, kterými se sníží počet kandidátních bodů. Později se provedou náročnější výpočty s již klíčovými body.

Nejprve bude načten obrázek, který bude muset být předzpracován a to tak, že bude zvětšena jeho velikost na dvojnásobek a rozostřen pomocí Gaussovy funkce s původním měřítkem sigma.

Následuje sestavení scale-space za pomoci Gaussových funkcí s rozdílným měřítkem. Pro vytvoření následující oktávy se vezme poslední obraz vrstvy, který je podzvorkován na poloviční velikost. Poté se za pomoci rozdílu gaussových funkcí sestrojí konečná pyramida, ve které budou hledány význačné body. Projde se celá pyramida a každý bod s dostatečným kontrastem se zkontroluje, zda není maximem nebo minimem ve svém okolí. Porovnání se provádí na jeho 26 sousedech. Osm sousedních bodů na stejné vrstvě jako je zkoumaný bod, dále devět bodů ležících ve vrstvě nad a pod aktuální zkoumanou vrstvou.

Jakmile najdeme všechny extrémy, přistoupíme k jejich přesnější lokalizaci. Po upřesnění lokalizaci jsou opět odstraněny z dalších výpočtů body s malým kontrastem. Následně jsou odstraněny body, které mají velkou hranovou odezvu (tzn. leží na hraně nějakého objektu).

Tímto dostáváme význačné body s přesnou polohou v obraze. Nyní zbývá každému bodu přiřadit orientaci a deskriptor. Nejdříve přiřadíme významnou orientaci bodu a to tak, že si sestrojíme histogram orientací pro každý bod a vybereme maximální hodnotu orientace a tu přiřadíme. Jak jsme se zmiňovali v předchozí kapitole, tak pro zlepšení nalezení korespondencí při porovnávání obrázků zjistím zda se ještě u bodu nenachází další významná orientace. Pokud ano vytvoříme nový bod a přiřadíme mu tuto orientaci.

Dále pro každý bod spočítáme deskriptor, který bude využíván k porovnání bodů z více obrázků a nalezení jejich korespondencí. Deskriptor bude mít 128 binů a rozdělen do 16

oblastí, kde každá bude obsahovat 8 binů. Abychom zajistili nezávislost deskriptoru na rotaci obrázku, tak nejprve natočíme zkoumanou oblast podle orientace bodu. Spočítáme orientace v okolních bodech a vytvoříme deskriptor bodu.

Dále naimplementuji jednoduché porovnávání bodů z více obrázků. Za pomoci porovnání deskriptorů a jejich Euklidovské vzdálenosti. Jedná se o jednu z metod, která se dá pro porovnání použít. Pro implementaci použiji MATLAB.

Výsledný program by se měl skládat ze tří zdrojových souborů. Prvím bude rozhraní pro ovládání detektoru, budou zde zpracovány parametry, které uživatel zadá při spuštění. Bude se spouštět samotný detektor. Výpočet význačných bodů bude umístěn do samostatného zdrojového souboru. Provede se zde celý výpočet a vrátí se seznam obsahující nalezené body. Posledním souborem bude porovnání deskriptorů z více obrázků a nalezení jejich korespondencí. Bude neimplementováno za pomoci MATLABu. Ke své práci bude potřebovat seznam deskriptorů, které získáme z předchozích výpočtů.

## Kapitola 5

# Implementace

Detektor jsem naprogramoval v jazyce C. Za použití knihovny OpenCV od firmy Intel. Pro načítání, ukládání a zpracování obrazu jsem využil funkce dostupné v této knihovně.

Implementace je rozdělena do dvou souborů `sift.c` a `detector.c`.

Zdrojové soubory jsou uložena na příloženém CD.

### 5.1 `sift.c`

Hlavní funkcí v souboru `sift.c` je

```
CvSeq* siftDetector(char* fileName, CvMemStorage* storage),
```

která očekává název obrázku a místo, kam se bude ukládat sekvence bodů. Zapouzdřuje celý výpočet detekce význačných bodů a vrací sekvenci obsahující nalezené body. K výpočtu využívá ostatní funkce.

Nejprve musíme předzpracovat původní obrázek, k tomu použijeme funkci `initImg`, která původní obrázek zvětší na dvojnásobnou velikost, převede jej do odstínů šedi a použije Gaussovu funkci k předzpracování obrazu. Následujícím krokem je stestavení měřítkově nezávislé reprezentace obrazu tzv. scale space. K tomuto využíváme funkci `gaussPyramid`. Obrázek je za pomoci Gaussovy funkce rozostřen a při zdvojnásobení měřítka je podvzorován na poloviční rozměry. V každé oktávě generujeme 6 vrstev. Poté za pomoci rozdílu vrstev ve funkci `DoGPyramid` dostáváme konečnou reprezentaci scale space, kde každá oktáva obsahuje 5 vrstev (z toho 3 vrstvy ve kterých se budou hledat lokální extrémy)

Hledání lokálních extrémů má na starosti funkce `findExtrems`. Nejprve prohledává za pomoci `findMinMax` všechny body obrazu s jejich 26 sousedy a testuje zda-li se nejedná o maximum `isMax`, či minimum `isMin`. Následuje upřesnění polohy význačného bodu `accurateKeypointLoc`, ke kterému je využito počítání posunů ve třech směrech za pomoci výpočtu prvních a druhých derivací. Dále počítáme kontrast bodů `computeContrast` a body s malým kontrastem odstraňujeme z dalších výpočtů. Potom zjistíme zda bod nemá hranovou odezvu `eliminateEdgeResponse` tzn. leží na nějaké hraně, pokud má tak ho také vyřadíme z následujících výpočtů, jelikož dosahuje malé stability.

Dalším krokem bude výpočet orientací klíčových bodů `orientationAssignment`. Nejdříve si sestrojíme histogram orientací `orientHistogram` na základě velikosti a orientací v okolí bodu `computeMagnitudeOrientation`. Vyhladíme histogram `smoothHistogram` k dosažení lepších výsledků. A naposledy přiřadíme bodu významnou orientaci `addOrientations`.

Poslední fází sift detektoru je sestrojení deskriptorů `createDescriptors` pro význačné body. To se opět děje ve více fázích nejprve sestavujeme histogram `descriptorHistogram`,

pro sestavení deskriptoru. Pro lepší stabilitu deskriptorů upřesníme histogram pomocí interpolace `interpolateDescrHistogram`. Poté převedem data získaná z okolních bodů a uložená v histogramu na samostatný deskriptor `histogramToDescriptor`, což je vektor o 128 binech (128 celočíselných hodnotách). Závěrečnou úpravou je normalizace deskriptoru `normalizeDescriptor`. Normalizace se provádí, kvůli dosažení odolnosti klíčových bodů na změny osvětlení.

## 5.2 detector.c

Druhým souborem je `detector.c` zde jsou zpracovány parametry zadané při spuštění aplikace. Poté je volána funkce `siftDetector`, která nalezne význačné body v uvedeném obrázku a vrátí je v sekvenci. Podle zadaných parametrů můžeme dále postupovat:

- vyznačíme nalezené body do obrázku a jeho zobrazení uživateli a následné uložení do specifikovaného výstupního souboru.
- vyznačíme orientace bodů do obrázku, zobrazíme uživateli a uložíme do souboru.
- uložíme deskriptory bodů do určeného souboru. Tento soubor budeme moci použít pro hledání korespondencí mezi porovnávanými obrázky.

## 5.3 Ovládání

Příklady spuštění aplikace:

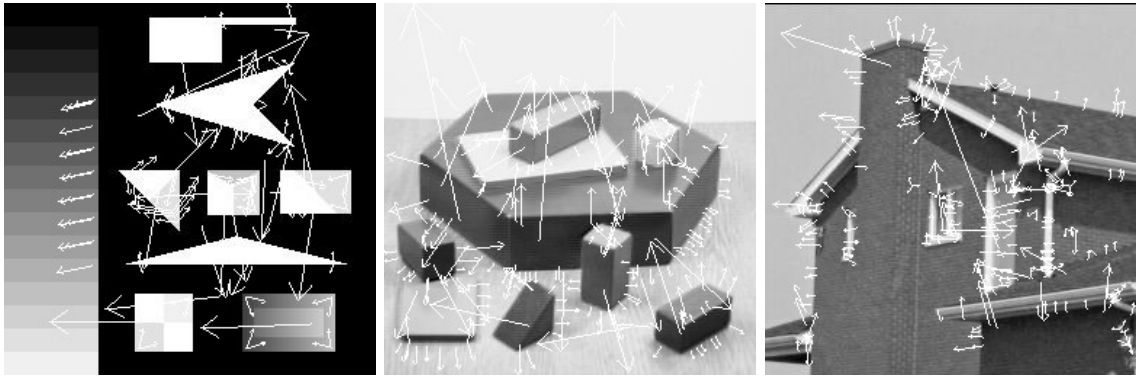
```
sift -h  
sift obrazek -p pointSoubor -o oriSoubor -d descrSoubor
```

parametr	popis
<b>-h</b>	Vypíše obrazovku s nápovědou ke spuštění
<b>obrazek</b>	Název obrázku, povinný údaj, musí být hned za názvem programu. Soubor, ve kterém se budou hledat význačné body.
<b>-p</b>	Nepovinný parametr, následovaný souborem k uložení obrázku s vyznačenými body
<b>pointSoubor</b>	Soubor pro uložení význačných bodů
<b>-o</b>	Nepovinný parametr, následovaný souborem k uložení obrázku s vyznačenými orientacemi bodů
<b>oriSoubor</b>	Soubor pro uložení orientací bodů
<b>-d</b>	Nepovinný parametr, následovaný souborem k uložení deskriptorů bodů
<b>descrSoubor</b>	Soubor pro uložení deskriptorů bodů

## Kapitola 6

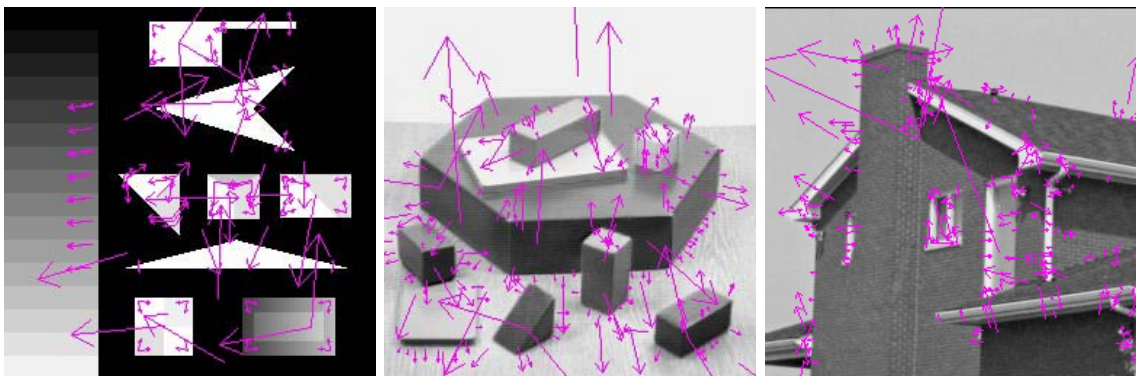
# Výsledky

Nejprve si ukážeme výsledky implementace na pár obrázcích a srovnáme je s Loweho verzí detektoru. Loweho detektor nám vrací obrázek se zvýrazněnými orientacemi nalezených bodů, a tak i v mojí implementaci použijeme zobrazení orientací, abychom mohli provést zhodnocení správnosti.



Obrázek 6.1: Loweho detektor

Použijeme stejné obrázky s vyznačenými orientacemi bodů, které jsem dostal pomocí mojí implementace detektoru.



Obrázek 6.2: Moje implementace SIFT detektoru

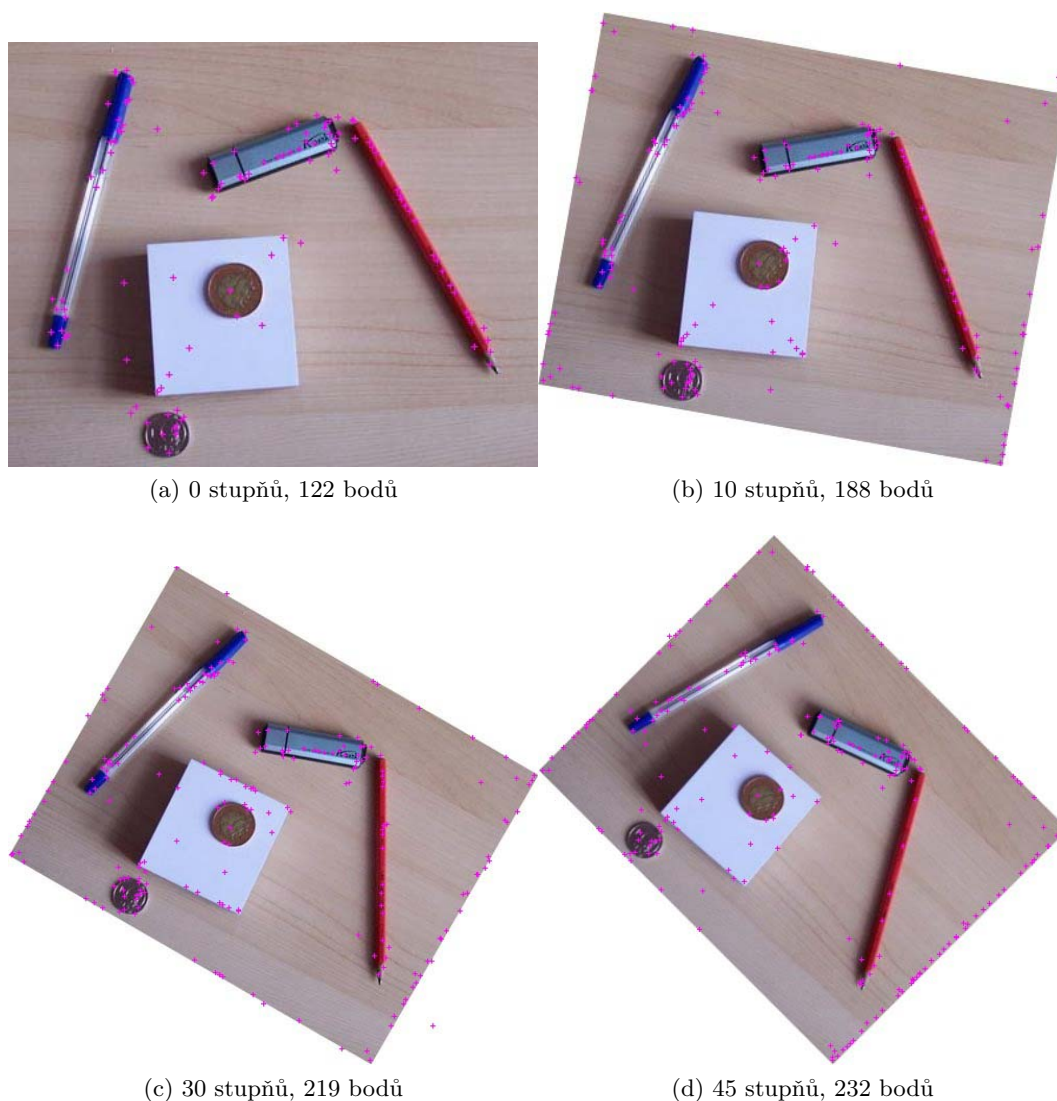


Na obrázcích můžeme vidět, že se orientace u některých bodů mírně liší. Avšak tento fakt je způsoben použitými konstantami při výpočtech.

Následně si ověříme odolnost metody na použité transformace v obraze.

## 6.1 Rotace obrázku

Začneme odolností na rotaci. Na obrázcích 6.3 máme zobrazenou scénu a její tři natočení. Okrajové body u natočených příkladů zanedbáme, jelikož jsou způsobeny kontrastem mezi obrázkem a bílým pozadím.



Obrázek 6.3: Rotace obrázků

Můžeme zde pozorovat, že při různých natočeních nacházíme podmnožiny stejných stabilních bodů, které jsou vhodné pro porovnávání snímků. Například body na hřbetu tužky, body na usb flash disku. Ovšem vyskytnou se nám i takové body, které jsou detekovány

jen při určitém natočení.

## 6.2 Změna osvětlení

Dále si ukážeme chování detektoru na změnu osvětlení 6.4. Máme zde čtyři obrázky se zvětšující se mírou dopadajícího světla na objekty.



(a) 24 bodů

(b) 97 bodů



(c) 118 bodů

(d) 113 bodů

Obrázek 6.4: Změny v osvětlení scény, na obrázku (a) máme scénu s nejmenším osvětlením, míra osvětlení se zvětšuje a na obrázku (d) je největší

Lze pozorovat, že při nízkém osvětlení je detekováno minimum bodů, avšak některé z bodů jsou dosti stabilní a jsou nalezeny i při vyšších intenzitách světla. Opět zde najdeme stabilní množinu bodů, která se vyskytuje ve všech ukázkách.

## 6.3 Změna velikosti

Další transformací bude změna velikosti. Detekce je prováděna s obrázky různých velikostí obrázky 6.5.



(a) původní velikost, 122 bodů

(b) o 20% větší, 179 bodů



(c) o 20% menší, 80 bodů

(d) o 50% menší, 48 bodů

Obrázek 6.5: Změny velikosti rozlišení scény

Vidíme, že změna velikosti nemá přílišný vliv na změnu detekovaných bodů. Avšak čím větší rozměry obrázku použijeme, tím výpočet nalezení význačných bodů trvá déle. Za stabilní body můžeme např. označit levý dolní roh bločku, body jeho stínu a další.

## 6.4 Přidání šumu

Do původního obrázku 6.6 přidáme šum, aby chom otestovali stabilitu metody na tuto vlastnost.

Přidáním malého šumu, který můžeme vidět na obrázku (b) nám nijak rapidně nezvedne počet detekovaných bodů a ani nenaruší schopnost detekovat stabilní body. Zatím co při zvětšení šumu, obrázek (c) dostáváme skupinu bodů, které se vyskytly na základě šumu, ale i tak stále dokážeme detekovat stabilní význačné body jako v původním obraze (a).



(a) 122 bodů



(b) 158 bodů

(c) 298 bodů

Obrázek 6.6: Přidání šumu do obrazu scény

## 6.5 Změna pozorovací polohy

Posledním testem bude změna pozorovacího bodu na obrázku 6.7

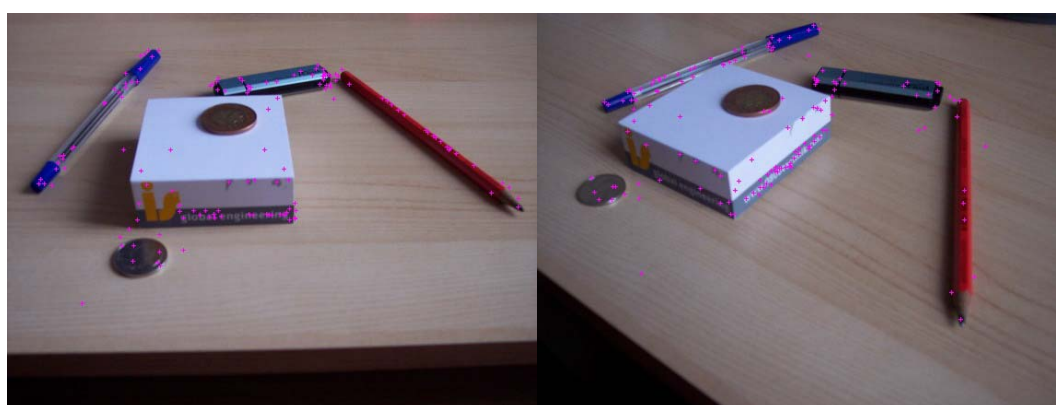
Vidíme, že se opět najdou stabilní body, které nám pomohou v porovnávání více obrázků. Dostí, ale záleží jak moc se pozorovací bod změní, jelikož při velkých posunech se různé objekty mohou jevit úplně jinak.





(a) 120 bodů

(b) 93 bodů



(c) 140 bodů

(d) 148 bodů

Obrázek 6.7: Změna pozorovací polohy

## 6.6 Zhodnocení

Metoda je odolná vůči

- změně velikosti
- rotaci
- výskytu šumu
- změně osvětlení
- částečně změně úhlu pohledu

rozlišení	čas
1152x864	~4s
576x432	1-2s
288x216	~1s

Tabulka 6.1: Dosažené výsledky při zkoumání obrázku 6.6a na počítači s procesorem 2GHz.

Nevýhodou metody SIFT je fakt, že její výpočet je náročný a tudíž metoda není schopna podávat výsledky v reálném čase pro větší obrázky. Tímpádem metoda není vhodná pro real-time aplikace. Při zvětšování rozlišení se i délka výpočtu prodlužuje 6.6.

Ověřili jsme si stabilitu metody na uvedené transformace v obraze. Detailnější zobrazení testovacích obrázků můžeme nalézt na příloženém CD.

# Kapitola 7

## Závěr

Cílem bylo seznámení se s metodami detekce význačných bodů. Po prostudování metod mě zaujala metoda SIFT, kterou jsem se rozhodnul naimplementovat. Implementace byla úspěšně realizována, ovšem již jsem nestihnul naprogramovat porovnávání význačných bodů jak bylo v plánu. Ověřil jsem si, že metoda SIFT je stabilní, vůči uváděným změnám. Přínosem bylo získání informací v oblasti počítačového vidění, zpracování obrazu a detekce bodů.

Nevýhodou metody SIFT je fakt, že její výpočet je náročný a tudíž metoda není schopna podávat výsledky v reálném čase, viz. tabulka 6.6. Z toho vyplývá, že není vhodná pro aplikace, které potřebují výsledky okamžitě.

Práce by se dala rozšířit o plánované porovnání bodů ve dvou obrázcích na základě deskriptorů nalezených bodů. Pomocí tohoto porovnání by se dala práce dále vylepšit například o spojování fotografií do panoramatických snímků nebo k hledání určitých objektů v náročnějších scénách.

# Literatura

- [1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, pages 404–417, Graz, Austria, May 2006. <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>.
- [2] S. Bres and J.M. Jolion. Detection of interest points. <http://rfv.insa-lyon.fr/~jolion/Cours/ptint.html>.
- [3] M. Brown and D. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference*, pages 656–665, Cardiff, Wales, September 2002. <http://citeseer.ist.psu.edu/brown02invariant.html>.
- [4] Petr Břelk. Významné body v obraze: detekce, korespondence a lokalizace ve 3d. Master's thesis, 2007. [https://dip.felk.cvut.cz/browse/pdfcache/bilekp3\\_2007bach.pdf](https://dip.felk.cvut.cz/browse/pdfcache/bilekp3_2007bach.pdf).
- [5] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. <http://kybele.psych.cornell.edu/~edelman/Archive/nips97.pdf>.
- [6] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988. [http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A\\_Combined\\_Corner\\_and\\_Edge\\_Detector.pdf](http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A_Combined_Corner_and_Edge_Detector.pdf).
- [7] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [8] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):pp. 77–116, 1998. <ftp://ftp.nada.kth.se/CVAP/reports/cvap198.pdf>.
- [9] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999. <http://www.cs.ubc.ca/~lowe/papers/iccv99.pdf>.
- [10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):pp. 91–110, 2004. <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.
- [11] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of British Machine Vision Conference*, pages 384–396, 2002. <http://cmp.felk.cvut.cz/~matas/papers/matas-bmvc02.pdf>.



- [12] Hans P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, Robotics Institute, Carnegie-Mellon University, September 1980. [http://www.ri.cmu.edu/pub\\_files/pub4/moravec\\_hans\\_1980\\_1/moravec\\_hans\\_1980\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub4/moravec_hans_1980_1/moravec_hans_1980_1.pdf).
- [13] Donovan Parks and Jean-Philippe Gravel. Corner detectors. <http://www.cim.mcgill.ca/~dparks/CornerDetector/index.htm>.
- [14] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. Technical Report TR95SMS1c, Chertsey, Surrey, UK, 1995. [citeseer.ist.psu.edu/smith95susan.html](http://citeseer.ist.psu.edu/smith95susan.html).
- [15] WWW stránky. Detekce význačných bodů. [http://en.wikipedia.org/wiki/Interest\\_point\\_detection](http://en.wikipedia.org/wiki/Interest_point_detection).
- [16] WWW stránky. knihovna opencv. <http://www.intel.com/technology/computing/opencv/>.
- [17] Andy P. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.