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 ROHŮ V OBRAZE

CORNERS DETECTION IN IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE Petr Hýna
AUTHOR

VEDOUCÍ PRÁCE Ing. Michal Španěl
SUPERVISOR

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Hýna Petr**
Obor: Informační technologie
Téma: **Detekce rohů v obraze**
Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte základy zpracování obrazu. Především se zaměřte na detekci hran a význačných bodů v obraze.
2. Zorientujte se v metodách detekce rohů a hran v obraze a popisu objektů.
3. Vyberte vhodné metody detekce rohů a navrhněte jednoduchý rozpoznávač objektů v obraze.
4. Experimentujte s vaší implementací.
5. Případně navrhněte 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 zadaní.

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 paměťovém médiu (disketa, CD-ROM), 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 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 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: **Petr Hýna**
Id studenta: 84351
Bytem: Prokopova 1576, 258 01 Vlašim
Narozen: 25. 09. 1980, Vlašim
(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 rohů 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 tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....	.....
Nabyvatel	Autor

Abstrakt

Detekce rohů je důležitá z hlediska získání podmnožiny bodů z množiny všech bodů obrazu, podle které lze následně zkoumat vztahy mezi obrazy. Zkoumání takových vztahů je pak méně časově a paměťově náročné. Tato práce se zabývá detekováním významných bodů se zaměřením na metodu Harris Stephens. Implementace je součástí MDSTk (Medical data segmentation toolkit). Program je napsán v jazyce C/C++ s využitím knihoven MDSTk.

Klíčová slova

Zpracování obrazu, významný bod, Harris Stephens, MDSTk, modul, gradient, detekce.

Abstract

Corner detection is important because of getting a subset of points from the set of all picture points. This subset is used for enquiry into relations of images. Searching of such relations is then less time and memory consuming. The text is about corner detection with focus on method Harris Stephens. The implementation is part of MDSTk and it is written in C/C++ with use of MDSTk's libraries.

Keywords

Image processing, point of interest, Harris Stephens, MDSTk, module, gradient, detection.

Citace

Petr Hýna: Detekce rohů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2007

Detekce rohů 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.

……
Petr Hýna
30. dubna 2007

Poděkování

Chtěl bych poděkovat vedoucímu své bakalářské práce Ing. Michalu Španělovi za jeho cenné rady a odbornou pomoc, kterou mi poskytoval v průběhu řešení. Zejména bych chtěl ocenit pomoc svého vedoucího, kterou mi poskytoval při seznamování se s MDSTk.

© Petr Hýna, 2007.

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

Obsah 1
1 Úvod 2
2 Teoretická část - úvod 4
2.1 Obraz 4
2.2 Gradient, Sobelův operátor, Gaussův filtr 5
3 Přehled metod 8
3.1 Rozdělení metod 8
3.2 Moravec algoritmus 8
3.3 SUSAN algoritmus 10
3.4 Trajkovic Hedley algoritmus 11
3.5 Wang Brady algoritmus 13
3.6 Harris Stephens algoritmus 13
3.7 Modifikace algoritmu Harris Stephens 17
4 Návrh implementace 20
5 Implementace 24
6 Výsledky 28
6.1 Úloha Gaussova filtru 30
7 Závěr 32
Literatura 33
Seznam zkratek a symbolů 35
Seznam příloh 36

1 Úvod

Detekce rohů i jiných významných bodů je nepostradatelnou metodou při zpracování obrazu. Základem mnoha aplikací je porovnávání dvou či více obrázků. Jde například o navazování jednotlivých snímků kamery či určování podobnosti dvou obrázků. Při zkoumání podobnosti obrázků metodou porovnávání všech bodů by bylo nezbytné porovnávat obrovské množství dat a jednalo by se o zdlouhavý proces. Intuitivně lze použít pouze omezenou množinu bodů, která zahrnuje body nějakým způsobem zajímavé a významné. Tyto body známe jako významné body. K nalezení vztahu mezi dvěma obrázky stačí významné body. Významným bodem může být roh, ale může se jednat například i o izolovaný bod uvnitř oblasti s konstantním jasem. Rohové body, které jsou podmnožinou významných bodů, jsou specifické tím, že určují hranice objektů nebo částí objektu. Na roh se můžeme dívat jako na spojnici dvou či více hran tj. jako na bod, který se významně liší od svého okolí. K detekování rohových bodů se používají stejné principy a algoritmy jako pro detekci významných bodů. Detekce významných bodů v obraze nebo také detekce rohů je jednou ze základních metod zpracování obrazu. Detekce významných bodů se mimo jiné používá v oblastech:

- vyhledávání korespondence ve dvou snímcích téže scény,
- rozpoznávání a detekování objektů,
- detekování a sledování pohybu,
- tvorba panoramatických fotografií,
- navigace robota.

Kvalita detektoru je posuzována podle schopnosti detekovat stejný bod ve více snímcích, které jsou podobné avšak různě posunuté. Obecně jsou na detektory kladeny tyto nároky:

- nalezení všech správných rohů,
- přesná lokalizace nalezených rohů,
- vyhnutí se falešným rohům,
- odolnost vůči šumu,
- opakovatelnost řešení,
- efektivita výpočtu.

Většina v praxi používaných rohových detektorů detekuje kromě rohů i další významné body.

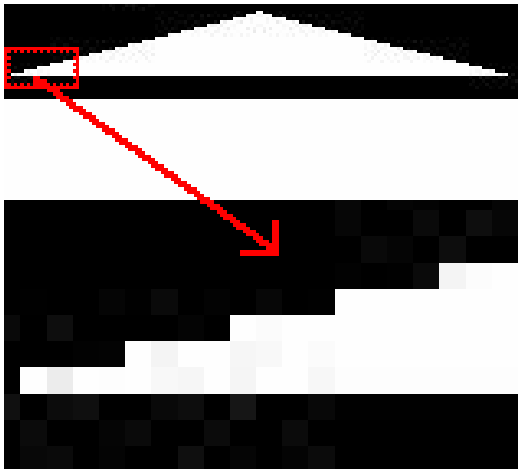
Na úvod práce jsou vysvětleny základní pojmy, které se týkají zpracování obrazu jako obrázek, pixel, gradient nebo Gaussův filtr. Dále jsou popsány některé známé metody detekce zahrnující algoritmus Moravec, SUSAN metodu, Trajkovic Hedley operátor nebo metodu Wang Brady. Následuje analýza metody Harris Stephens se zaměřením na porovnání modifikací této metody. V další části následuje vysvětlení principu nalezení významných bodů v navržené implementaci. Na závěr jsou probrány výsledky experimentování s implementací.

Protože cílem nebylo vytvořit nový propracovaný program na rozpoznávání významných bodů, ale vyzkoušet implementaci prostudovaných algoritmů, zaměřil jsem se na přehlednou implementaci algoritmu Harris Stephens, který patří mezi nejpoužívanější metody detekování významných bodů. Výsledná implementace je součástí MDSTk.

2 Teoretická část - úvod

2.1 Obraz

Rastrové obrázky (dále jen obrázky) se používají pro ukládání obrazů reálného světa, jako jsou skenované obrázky, digitální fotografie atp. Obrázek je posloupnost bodů, které jsou rozmístěné v mřížce (rastru). Obraz můžeme také chápat jako diskrétní funkci I(x,y) dvou proměnných, které nazýváme souřadnice. Funkční hodnota vyjadřuje např. jas v obraze. Všechny tyto funkční hodnoty tvoří dohromady matici čísel, pomocí níž je popsán obraz.



Obr. 2.1 Rastrový obrázek

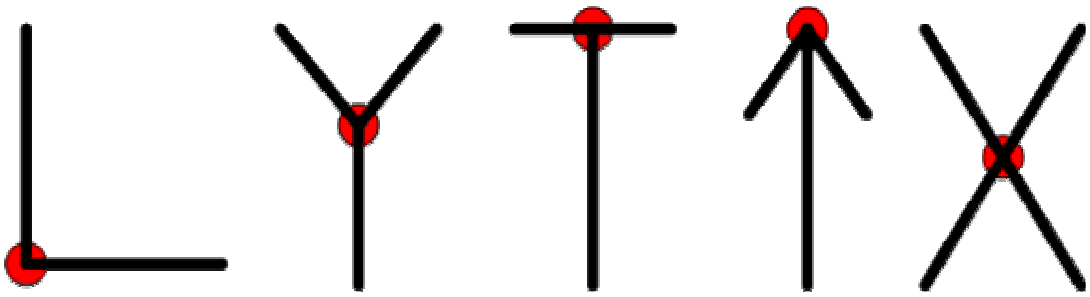
Počet bodů v rastru určuje rozměry obrázku, které zásadně ovlivňují dobu průchodu algoritmem pro vyhledání významných bodů. Každý bod obsahuje informaci o své barvě, která je ovlivněna počtem bitů, na kterých je uložena. Podle počtu barev rozdělujeme obrázky do tří kategorií:

- monochromatické – obsahují pouze bílou a černou barvu,
- ve stupních šedi,
- barevné.

Rastrová grafika sebou přináší vedle výhod (jednoduchá reprezentace dat, převod obrazových předloh do rastrové grafiky pomocí skeneru) i některé nevýhody, mezi které patří:

- velké nároky na zdroje při použití bezeztrátové komprese,
- zhoršení kvality obrázku zapříčiněné změnou jeho velikosti,
- zvětšování obrázku je možné pouze v omezené míře.

Z hlediska zpracování obrazu nás zajímá intenzita ve všech bodech obrázku pomocí níž jsme schopni určit polohu významných bodů. Obrázek 2.2 znázorňuje přehled uvažovaných významných bodů.



Obr. 2.2 L-uzel, Y-uzel, T-uzel, Šipka-uzel, X-uzel

Pixel (zkrácení anglických slov picture element, obrazový prvek) je nejmenší jednotka digitální rastrové grafiky. Představuje jeden bod obrázku zadaný svou barvou, např. ve formátu RGB či CMYK. Při implementaci detektoru významných bodů se zabýváme pixely ve stupních šedi. Body na obrazovce tvoří čtvercovou síť a každý pixel je možné jednoznačně identifikovat pomocí jeho souřadnic x, y.

2.2 Gradient, Sobelův operátor, Gaussův filtr

Pro pochopení metody Harris Stephens je potřeba vysvětlit pojem gradient. Obrazový gradient je jedním ze základních stavebních kamenů při zpracování obrazu. Matematicky gradientem rozumíme diferenciální operátor, jehož výsledkem je vektorové pole, které vyjadřuje směr a velikost největší změny skalárního pole. Složky jednotlivých vektorů, které jsou v našem případě dvourozměrné, jsou dány derivacemi intenzity v horizontálním a vertikálním směru. Poněvadž je intenzita definována pouze v diskrétních bodech, nemůžeme definovat derivaci intenzity. Musíme nejprve předpokládat, že existuje spojitá funkce intenzity, která byla navzorkovaná v jednotlivých bodech snímku. Více na [13]. Při detekování významných bodů se používá aproximace obrazových gradientů pomocí Sobelova operátoru.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Obr. 2.3 Sobelův operátor

Obr. 2.4 Lena – Sobelův operátor

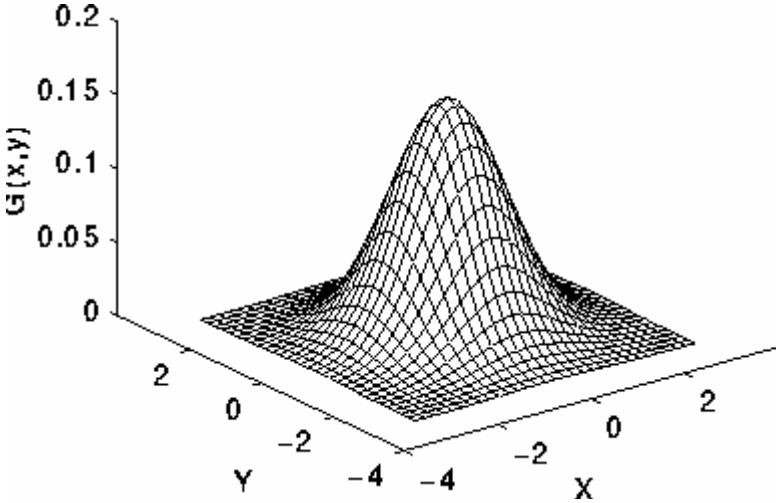
Sobelův operátor dosahuje uspokojivých výsledků na to, aby se dal využít v praxi. Používá hodnoty intenzity pouze v oblasti 3x3 kolem každého bodu snímku k vyjádření odpovídajících gradientů. Hlavní předností výpočtu gradientů pomocí Sobelova operátoru je získání izotropních výsledků. Izotropní výsledky jsou takové, kdy nezáleží na směru, ve kterém gradienty počítáme tzn. je zajištěna opakovaná detekce bodů např. pod změněným úhlem pohledu. Při detekování významných bodů za pomoci modulu, který je součástí MDSTk, byly použity dva způsoby výpočtu gradientů, Sobelův operátor a vzorec (2.1):

bodX = bod(x + 1, y) - bod(x - 1, y);
bodY = bod(x, y + 1) - bod(x, y - 1); (2.1)

Výsledné hodnoty gradientu jsou v tomto případě (2.1) určeny pomocí sousedních bodů ve svislém resp. vodorovném směru.

V implementaci je použita metoda filtrování obrázku pomocí Gaussova filtru [12], při které jsou jednotlivé pixely pokryté kernelem podle Gaussovy funkce. Gaussův filtr je použit pro vyhlazení hran ve vstupním obrázku.

$$G(x,y) = \frac{1}{2*\pi*\sigma^2} * \exp\left(-\frac{(x^2+y^2)}{2*\sigma^2}\right) \tag{2.2}$$



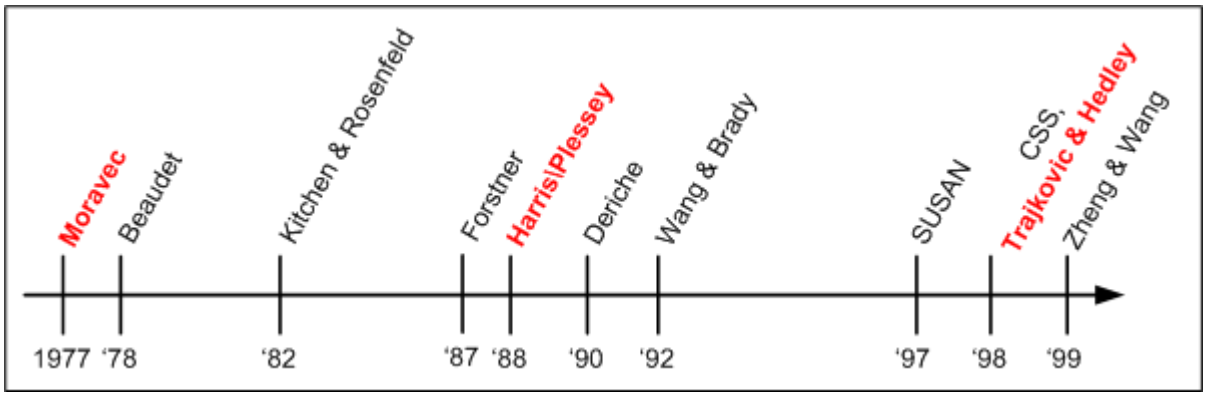
Obr. 2.5 2-D Gaussova distribuce se středem v bodě (0,0) a $\sigma = 1$.

Gaussův filtr patří mezi konvoluční filtry. Tyto filtry se aplikují postupně na všechny body obrazu a používají váženého součtu hodnot z okolí daného bodu. Pro konvoluční filtry je typické, že se ve výsledku mohou objevit hodnoty, které v původním obrázku nejsou. Vyhlazení hran pomocí Gaussova filtru je zachyceno na obrázku 4.4.

3 Přehled metod

3.1 Rozdělení metod

V současné době je známo více metod detekce významných bodů, které pracují na odlišných principech. Nejrozšířenější metodou je algoritmus Harris Stephens, který je také znám jako Plessey algoritmus. Chronologický vývoj je ukázán na obrázku 3.1



Obr. 3.1 Vývoj metod detekování významných bodů.

Principy detekování lze uměle rozdělit do několika kategorií:

- hranové metody – fungují na principu sledování změn zakřivení hran, jedná se např. o metodu Wang Brady,
- autokoleračních metody – výpočet významných bodů na základě posouvání obrázku různým směrem, zástupcem této skupiny algoritmů je např. operátor Harris Stephens,
- ostatní – použití kruhového okolí zkoumaného bodu, metody založené na principu zkoumání a porovnávání hodnot jasů v tomto okolí, do této skupiny patří např. SUSAN, Trajkovic Hedley operátor.

V následujících podkapitolách jsou podrobněji probrány některé algoritmy.

3.2 Moravec algoritmus

Jednou z prvních metod, které se zabývaly touto problematikou, je Moravec algoritmus [2], z něhož vychází dále zkoumaný algoritmus Harris Stephens. Moravec algoritmus byl objeven Hans P. Moravcem v roce 1977 kvůli výzkumu týkajícího se navigace robota. Moravec algoritmus detekuje rohy na základě podobnosti s okolními body. Moravec navrhl umístění malé čtvercové oblasti nad zkoumaný bod. Následně se pohybuje touto oblastí o jeden pixel v každém z osmi směrů

(horizontálně, vertikálně a ve směru obou diagonál). Podobnost je určena ze sumy rozdílů čtverců mezi dvěma vzájemně se překrývajícími místy. Významný bod se detekuje za předpokladu, že zkoumaný bod není podobný svému okolí tzn. jeho jas se významně mění ve všech směrech. Jako měřítko rohovosti se bere minimální hodnota jasu nalezená při posunu oblasti všemi osmi směry.

Moravec algoritmus

Vstup: obrázek ve stupních šedi, velikost okna, prahová hodnota T

Výstup: mapa ukazující pozice všech detekovaných rohů

1. *Pro každý pixel (x, y) v obrázku spočítej jasovou odlišnost při posunutí (u, v) jako:*

$$V_{u,v}(x,y) = \sum_{\forall a,b} \left(I(x+u+a, y+v+b) - I(x+a, y+b) \right)^2$$

kde uvažovaná posunutí (u,v) jsou: (1,0),(1,1),(0,1),(-1,1),(-1,0),(-1,-1),(0,-1),(1,-1)

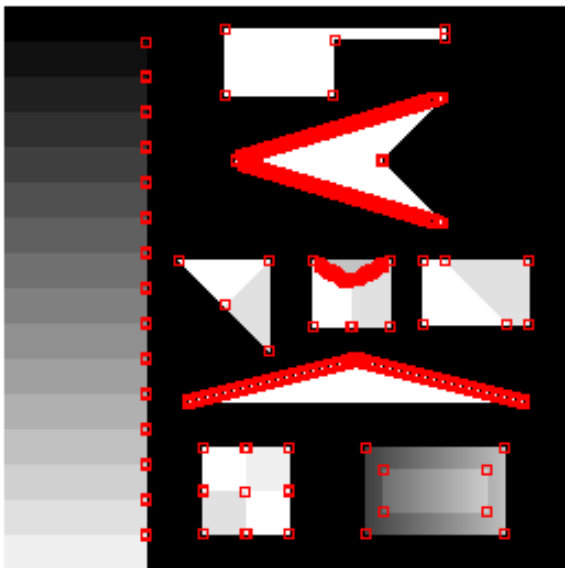
2. *Pro každý pixel (x,y) spočítej měřítko rohovosti C(x, y):*

$$C(x,y) = \min\left(V_{u,v}(x,y)\right)$$

3. *Porovnáním výsledných hodnot s prahovou hodnotou urči všechna C(x, y), která jsou menší než T a změn tyto na nulu.*
4. *Proveď výběr lokálních maxim.*

Všechny body, které mají nenulovou hodnotu jsou rohové body.

Algoritmus má problémy s detekováním hran jako významných bodů, kdy ve směru kolmém na směr přímky detekuje body na přímce jako významné.



Obr.3.2 Výsledky operátoru Moravec

3.3 SUSAN algoritmus

Na odlišném principu funguje například SUSAN (Smallest Univalue Segment Assymilating Nucleus) algoritmus [4], kde dochází k výpočtu jasu v kruhovém okolí zkoumaného bodu a porovnávání s hodnotou jasu zkoumaného bodu. K porovnání se používá funkce:

$$c(\vec{m}) = \exp\left(\frac{(I(\vec{m}) - I(\vec{m}_0))^6}{t}\right), \tag{3.1}$$

kde t určuje šířku.Exponent byl určen empiricky.

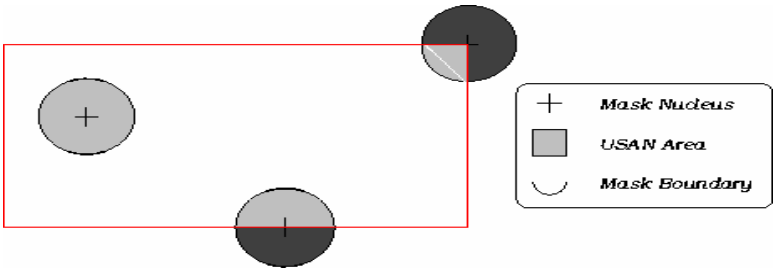
Algoritmus zahrnuje generování kruhové masky kolem daného bodu a porovnání jasových hodnot uvnitř oblasti. Tato procedura se opakuje pro všechny body v obrázku. USAN je oblast, ve které mají všechny body stejnou nebo podobnou jasovou hodnotu jako zkoumaný bod (jádro oblasti). Oblast USAN je popsána vzorcem:

$$n(M) = \sum_{\vec{m} \in M} c(\vec{m}), \tag{3.2}$$

Pokud je c rozhodovací funkce (3.2), pak n je počet pixelů v masce, které se liší svým jasem od jádra v mezích určených t. Odpověď operátoru USAN je dána následovně:

$$R(M) = g - n(M) \text{ pokud je } n(M) < g, \tag{3.3}$$

kde g je známé jako geometrická prahová hodnota. Jinak řečeno, operátor USAN dává kladnou odezvu (3.3) pouze v případě, že je oblast dostatečně malá. Nejmenší USAN oblasti mohou být určeny pomocí odstranění nemaximálních hodnot. Výsledkem je SUSAN operátor. K nalezení rohových bodů je potřeba nalezení těžišť USAN oblastí. Rohový bod bude mít těžiště daleko od jádra. Posledním nezbytným krokem je ujištění se, že všechny body na spojnici od jádra přes těžiště na okraj oblasti masky náleží do USAN. Tato technika je chráněná ve Velké Británii patentem číslo 2272285.

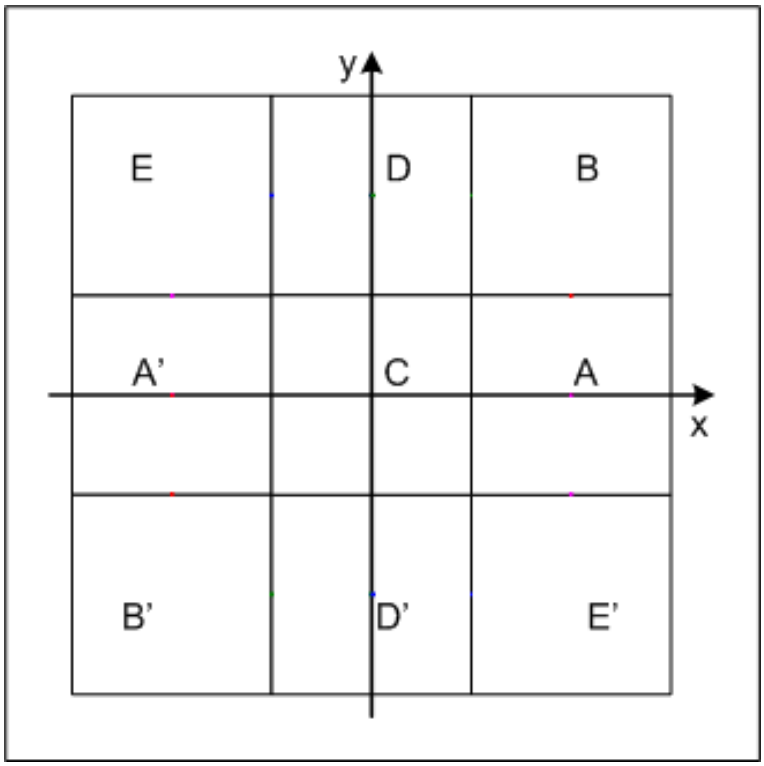


Obr. 3.3 Kruhové masky ukazující odlišné USAN oblasti

U SUSAN metody nemusí při posouvání kamery dojít k opakované detekci již jednou nalezeného bodu.

3.4 Trajkovic Hedley algoritmus

Jsou známy dvě verze operátoru Trajkovic. Rozdíl je v počtu uvažovaných okolních bodů, kde se používá pro výpočet buď čtyř nebo osmi bodů. Tyto verze jsou označovány jako Trajkovic4 a Trajkovic8 [10]. Trajkovic4 se neumí vypořádat s šumem v obraze a s detekováním falešných rohů na diagonálních hranách. Tyto nedostatky se částečně podařilo odstranit pomocí Trajkovic8, ale nadále zůstávají problémy s detekováním falešných rohů na některých diagonálních hranách.



Obr. 3.4 Trajkovic8 oblast 3x3

Trajkovic8 operátor

Vstup: obrázek ve stupních šedi, stupnice pro verzi nízkého rozlišení obrázku, prahové hodnoty T_1, T_2

Výstup: mapa ukazující pozice všech detekovaných rohů

1. Pro každý pixel (x, y) v obrázku s nízkým rozlišením spočítej jednoduché měřítko rohovosti:

$$C_{SIMPLE}(x, y) = \min(r_A, r_B, r_D, r_E)$$

kde:

$$r_A = (I_A - I_C)^2 + (I_{A'} - I_C)^2$$
$$r_B = (I_B - I_C)^2 + (I_{B'} - I_C)^2$$
$$r_D = (I_D - I_C)^2 + (I_{D'} - I_C)^2$$
$$r_E = (I_E - I_C)^2 + (I_{E'} - I_C)^2$$

Označ pixel (x,y) s $C_{SIMPLE}(x, y) \geq T_1$ *jako potenciální roh.*

2. Inicializuj mapu rohovostí M jako samé nuly.

3. Pro každý potenciální roh z bodu 1:

3a. Urči polohu pixelu v původním obrázku, (x?,y?).

3b. Spočítej $C_{SIMPLE}(x?, y?)$.

3c. Pokud je $C_{SIMPLE}(x?, y?) \leq T_2$*, pak pixel není rohem, jeho hodnotu M dej na nulu a pokračuj na bodu 3e.*

3d. Spočítej mezipixelovou aproximaci měřítka rohovosti:

Pokud je pro všechna i = 1,2,3,4 buď $B_i \geq 0$ *či* $A_i + B_i \leq 0$*, pak*

$C_{INTERPIXEL}(x, y) = C_{SIMPLE}(x, y)$.

V případě, že je pro všechna i = 1,2,3,4 buď $B_i < 0$ *či* $A_i + B_i > 0$*, pak*

$$C_{INTERPIXEL}(x, y) = \min\left(r_i - \frac{B_i^{\,2}}{A_i} \right),$$

kde:

$$r_1 = r_A, r_2 = r_B, r_3 = r_D, r_4 = r_E$$
$$B_1 = (I_B - I_A)(I_A - I_C) + (I_{B'} - I_{A'})(I_{A'} - I_C)$$
$$B_2 = (I_D - I_B)(I_B - I_C) + (I_{D'} - I_{E'})(I_{E'} - I_C)$$
$$B_3 = (I_E - I_D)(I_E - I_C) + (I_{E'} - I_{D'})(I_{E'} - I_C)$$
$$B_4 = (I_{A'} - I_E)(I_E - I_C) + (I_{A'} - I_{E'})(I_{E'} - I_C)$$
$$A_1 = r_B - r_A - 2B_1$$
$$A_2 = r_D - r_B - 2B_2$$
$$A_3 = r_E - r_D - 2B_3$$
$$A_4 = r_A - r_E - 2B_4$$

3e. Pokud je $C_{INTERPIXEL}(x?, y?) \leq T_2$ *ponech M na nule, jinak*

polož $M(x?, y?) = C_{INTERPIXEL}(x?, y?)$

4. Proveď výběr lokálních maxim M.

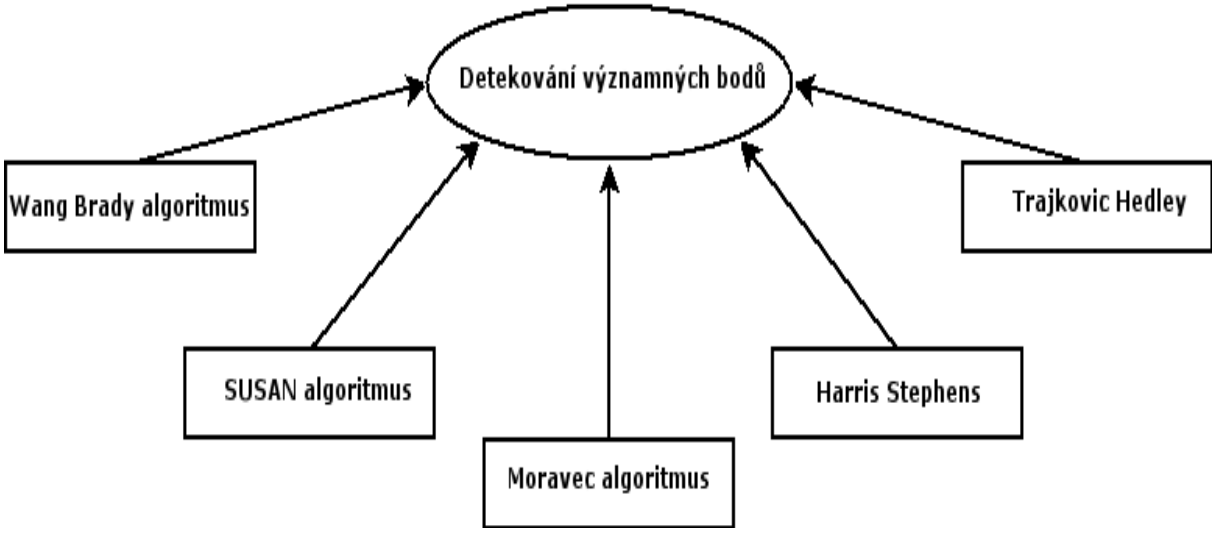
Všechny body, které mají nenulovou hodnotu jsou rohové body.

3.5 Wang Brady algoritmus

Na jiném principu funguje detektor Wang Brady [3], který zkoumá zakřivení hran a roh detekuje při náhlé změně směru přímky. Měřítko rohovosti je určeno následovně:

$$C = \nabla^2 I - c\left|\nabla I\right|^2, \tag{3.4}$$

kde c určuje citlivost na změnu směru přímky. U této metody je nezbytné použít Gaussův filtr pro snížení šumu. Vyhlazení pomocí Gaussova filtru ovšem způsobí špatné rozmístění detekovaných rohů. Algoritmus Wang Brady tento problém řeší aplikováním speciálního výrazu na detekované body, kdy po aplikování tohoto výrazu dojde k umístění rohů na odpovídající místa. Mezi nevýhody tohoto algoritmu patří špatné určování významných bodů při velmi malých a velmi velkých úhlech.



Obr. 3.5 Shrnutí algoritmů

3.6 Harris Stephens algoritmus

Harris Stephens algoritmus [1] pracuje na principu posunu zkoumaného snímku v různých směrech a výpočtu gradientu v jednotlivých bodech snímku. Problémem algoritmu Harris Stephens jsou světelné a hloubkové iluze, kdy například detekuje bod na rozhraní objektu a jeho vlastního stínu jako významný. Z tohoto algoritmu vychází množství novodobých algoritmů. Jedná se o metody Shi & Tomasi či více stupňový Harrisův detektor.

Základem metody je výpočet rozdílů čtverců jasů (SSD) ve všech bodech snímku. Princip funguje pouze tehdy, jestliže porovnávané obrázky mají stejný jas a kontrast. Citlivost na změny jasu a kontrastu je důležité omezení SSD. Na rozdíl od algoritmu Moravec uvažujeme přímo směr posunu snímku. Principem algoritmu Harris Stephens je výpočet autokorelační matice pro každý bod snímku.

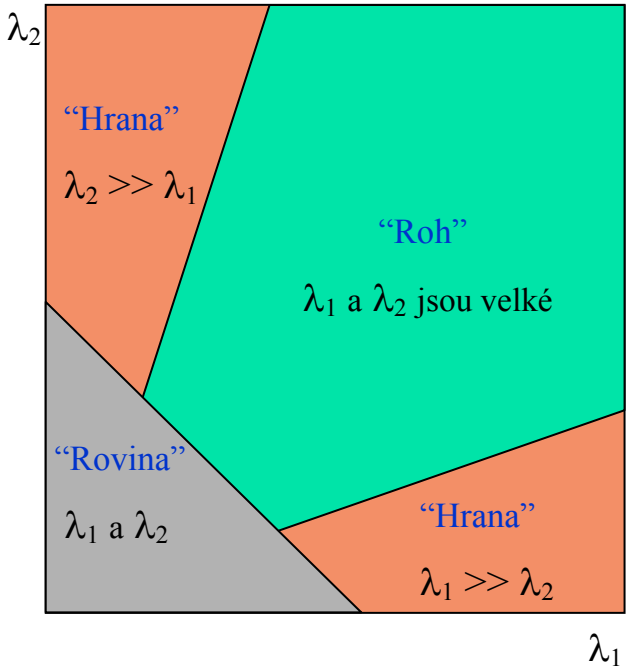
Jedná se o Hessovu matici druhých parciálních derivací. Pokud všechny druhé parciální derivace existují, má matice následující tvar.

$$A = \begin{bmatrix} \langle I^2{}_x \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I^2{}_y \rangle \end{bmatrix} \tag{3.5}$$

Z této matice (3.5) se vypočítají vlastní čísla, na základě kterých se rozhoduje, zda daný bod je či není významný.

$$\lambda_{1,2} = \frac{\langle I^2{}_x \rangle + \langle I^2{}_y \rangle \pm \sqrt{\left(\langle I^2{}_x \rangle - \langle I^2{}_y \rangle\right)^2 + 4 * \langle I_x I_y \rangle * \langle I_x I_y \rangle}}{2} \tag{3.6}$$

Aby byl bod významný z pohledu vlastních čísel, musí být tyto přibližně stejně velké a ostře větší než nula.



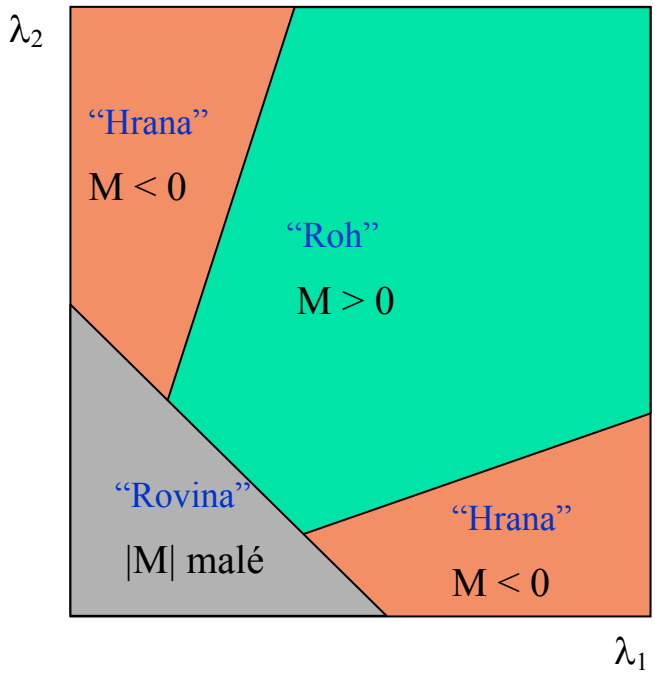
Obr. 3.6 Rozhodování na základě vlastních hodnot

Výpočet vlastních hodnot je časově náročný, poněvadž je zapotřebí použití odmocniny (3.6). Harris a Stephens namísto výpočtu vlastních hodnot navrhli vzorec (3.8), na jehož základě lze úspěšně rozhodnout, zda se v daném bodě nachází roh. Vzorec zahrnuje rozdíl determinantu matice druhých derivací a součtu prvků na hlavní diagonále umocněný na druhou. Součástí vzorce je nastavitelný parametr, který slouží pro optimální doladění detekování významných bodů.

$$M = \lambda_1 * \lambda_2 - \kappa * (\lambda_1 + \lambda_2)^2 \tag{3.7}$$

$$M = Det(A) - \kappa * Trace^2(A) \tag{3.8}$$

Hodnota κ byla určena experimentálně. Nejlepších výsledků má metoda dosahovat pro $\kappa = 0.04 \ldots 0.15$.



Obr. 3.7 Rozhodování na základě měřítka M

Pokud je tento rozdíl (3.8) větší než zadaná prahová hodnota, detekujeme významný bod. Nevýhodou metody je vysoký počet chybně určených významných bodů. Metoda má problémy se šumem v obraze, což se dá ovšem jednoduše odstranit pomocí Gaussova filtru. Dalším nedostatkem metody je spolehlivá detekce rohů pouze v L-uzlech. Největším problémem metody je neschopnost přesně detekovat rohy v T-uzlech. Stejně jako operátor Moravec dává i Harris Stephens neizotropní odpověď v případě, že se k výpočtu gradientů používají pouze horizontální a vertikální sousedé zkoumaného pixelu bez předchozí úpravy obrázku pomocí Gaussova filtru.

Harris Stephens

Vstup: obrázek ve stupních šedi,hodnota k, prahová hodnota T

Výstup: mapa ukazující pozice všech detekovaných rohů

1. *Pro každý pixel (x, y) v obrázku spočítej autokolerční matici A:*

$$A = \begin{bmatrix} \langle I^2{}_x \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I^2{}_y \rangle \end{bmatrix}$$

2. *Pro každý pixel (x,y) spočítej měřítko rohovosti C(x, y):*

$$C(x,y) = Det(A) - \kappa * Trace^2(A)$$
$$Det(A) = \lambda_1 \lambda_2$$
$$Trace(A) = \lambda_1 + \lambda_2$$

3. *Porovnáním výsledných hodnot s prahovou hodnotou urči všechna C(x, y), která jsou menší než T a změn tyto na nulu.*
4. *Proveď výběr lokálních maxim.*

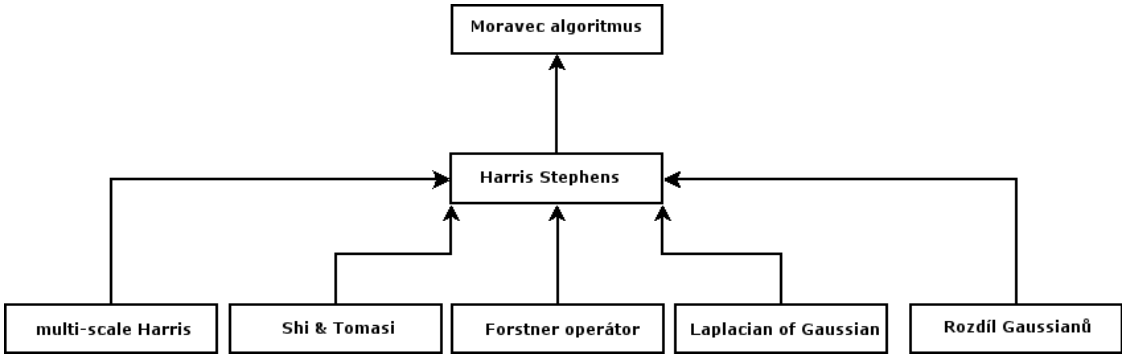
Všechny body, které mají nenulovou hodnotu jsou rohové body.

Z této metody vychází mnoho modifikací, se kterými se dá rozmanitě experimentovat. Jedná se o rozdělení snímku na oblasti s různými prahy, vyhledávání lokálních maxim, úpravu vzorce pro výpočet rohovosti atd. Mezi přednosti tohoto algoritmu patří schopnost detekovat invariantní oblasti.

OPERÁTOR	**POMĚR DOBŘE A ŠPATNĚ DETEKOVANÝCH ROHŮ**	**SPRÁVNOST ŘEŠENÍ**	**OPAKOVANÉ DETEKOVÁNÍ**	**ODOLNOST K ŠUMU**	**RYCHLOST**
Moravec	chvalitebný	dobré	chvalitebné	chvalitebné	dobrá
Harris Stephens	dobrý	dobré pro L-uzly	vynikající při afinní transformaci pro isotropní gradienty	chvalitebné	slabá
Wang Brady	dobrý	dobré	dobré	chvalitebné	dobrá
SUSAN	dobrý	špatné pro rozmazané obrázky, jinak dobré	dobré pro posunutí, špatné při afinní transformaci	vynikající	dobrá
Trajkovic8	chvalitebný	dobré	chvalitebné	dobré	vynikající

Tab. 3.1 Porovnání vybraných detektorů

3.7 Modifikace algoritmu Harris Stephens



Obr. 3.8 Modifikace metody Harris Stephens

Všechny modifikace uvedené v této podkapitole jsou součástí navržené implementace. První modifikací Harrisova algoritmu použitou při implementaci je metoda Shi & Tomasi [5], kde slouží k rozhodování ohledně rohovosti pouze menší z výše počítaných vlastních čísel.

$$\lambda = \frac{\langle I^2{}_x \rangle + \langle I^2{}_y \rangle - \sqrt{\left(\langle I^2{}_x \rangle - \langle I^2{}_y \rangle\right)^2 + 4 * \langle I_x I_y \rangle * \langle I_x I_y \rangle}}{2}$$

Pokud je tato větší než prahová hodnota, pak se jedná o významný bod. Výsledky metody Shi & Tomasi ukazují menší chybovost při detekování významných bodů než ostatní modifikace.

Další rozhodovací funkcí při detekování významných bodů, která je součástí implementace, je Förstnerův operátor, kde se rohové body určí jako lokální maxima funkce M. Förstnerův operátor nepatří přímo mezi modifikace metody Harris Stephens, pracuje ale na téměř stejném principu, pouze k porovnávání používá odlišné měřítko.

$$M = \frac{\lambda_1 * \lambda_2}{\lambda_1 + \lambda_2} = \frac{Det(A)}{Trace(A)} \tag{3.9}$$

Při použití této metody program dosahuje horších výsledků než při použití metody Harris Stephens. Výsledky jsou horší jak v případě výpočtu pomocí vlastních čísel tak i v případě, kdy používáme výslednou úpravu vzorce 3.9. Förstnerův operátor je v implementaci použit současně s výběrem významných bodů na základě porovnání hodnoty M s předem vypočítanou prahovou hodnotou. Běžně se použití prahové hodnoty vynechává, ale v našem případě se ukazuje vhodnější tuto zakomponovat do řešení.

Metoda Haralick [14] spočívá v určení váhy (w) a kruhovosti (q) jednotlivých bodů.

$$w = Det(A)$$

Váha w je známá jako Beaudetovo měřítko rohovosti a je úměrná změně gradientu v okolí zkoumaného bodu. Pokud je změna gradientu výrazná, pak jde o možný významný bod. Kruhovost q popisuje vztah hlavní a vedlejší osy elipsy v okolí zkoumaného bodu. Kruhovost je velká za předpokladu, že jsou obě osy stejně velké.

$$q = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2$$

Kruhovost se používá pro odmítnutí bodů na hranách. Aby se tomu tak stalo, musí být hlavní osa ve směru hrany a q musí být menší než jedna.

Venerova modifikace [6] spočívá v aproximaci rozhodovací funkce M.

$$M = \left(\frac{Det(A)}{\left(Trace^2(A) - 3 * Det(A)\right) * 4} \right)^2 * \left(Trace^2(A) - 2 * Det(A)\right)$$

	HARRIS.JPG	TELE.JPG	POKUS.PNG
Shi Tomasi	2.110	10.688	2.657
Forstner - eigenvalues	2.046	10.593	2.842
Forstner operator	2.016	10.500	2.663
Harris Stephens	2.078	10.657	2.625
Harris Stephens - eig.val.	2.150	10.578	2.547
Haralick	2.141	10.672	2.531
Venera's acceleration	2.016	10.563	2.546

Tab. 3.2 Rychlost zpracování v sekundách.

Tabulka ukazuje rychlost nalezení významných bodů pomocí všech použitých metod. Je vidět, že použitá metoda nemá na řešení v našem případě podstatný vliv, záleží spíše na velikosti obrázku. Nicméně je nutné zmínit fakt, že metoda Harris Stephens patří mezi pomalé metody detekování významných bodů. Při pohledu na přejatou tabulku 3.3 je zřejmé, že metoda dosahuje několikanásobně horších výsledků v porovnání s jinými v praxi používanými metodami.

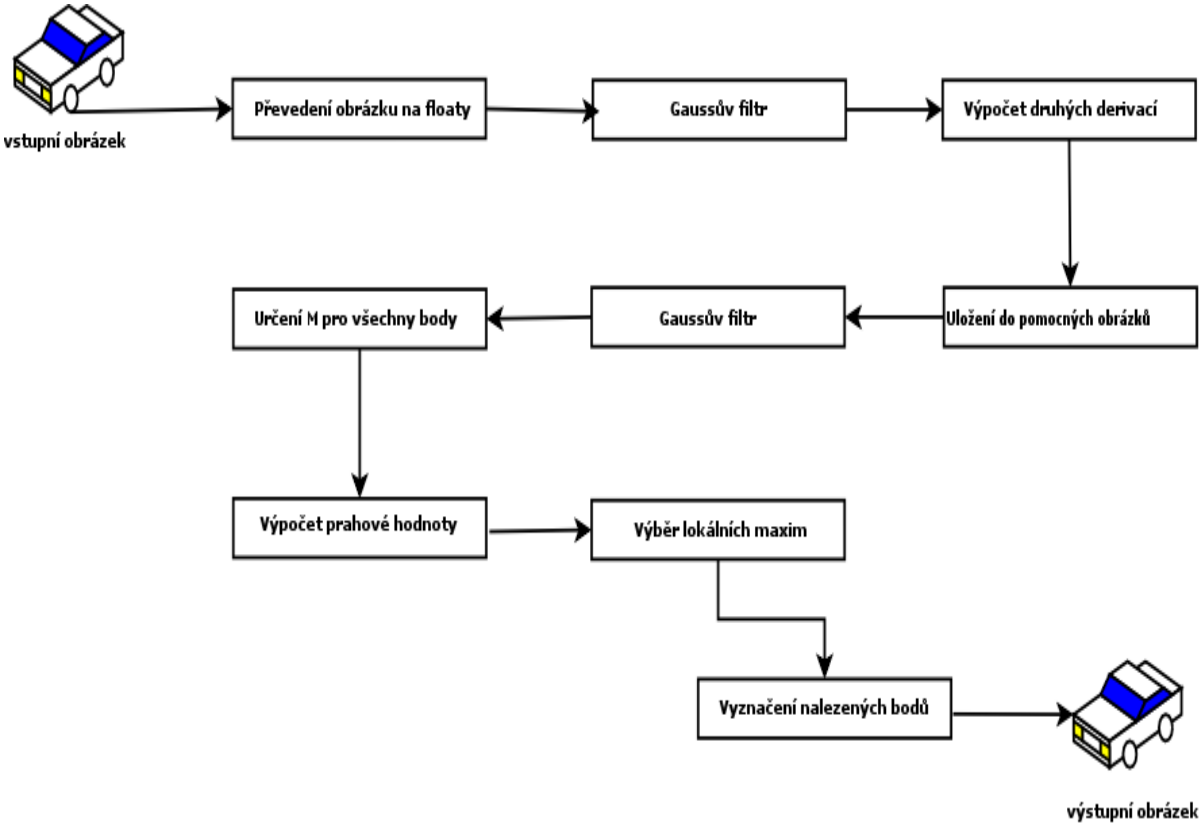
	SČÍTÁNÍ	NÁSOBENÍ	OPERACE NA PIXEL
Harris Stephens	95	22	117
SUSAN	32.25	0.75	33
Wang Brady	24.75	7.25	32

Tab. 3.3 Počet operací na jeden pixel.

Přesto je metoda Harris Stephens v praxi využívána, a to hlavně z důvodu vysoké opakovatelnosti řešení. V našem případě se navíc podařilo získat i izotropní výsledky, což byl u této metody donedávna problém.

4 Návrh implementace

Na obrázku 4.1 je znázorněn princip detekce rohových bodů použitý při implementaci algoritmu.



Obr. 4.1 Postup při detekování významných bodů v MDSTk.

Výše znázorněný postup lze shrnout do několika základních kroků:

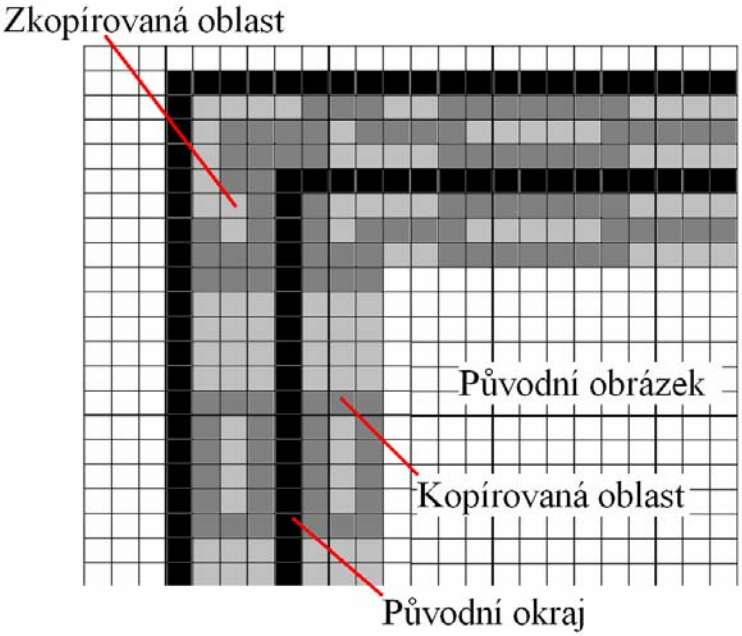
- načtení vstupního obrázku,
- úprava obrázku do vhodného formátu pro vyhledání významných bodů,
- výpočet měřítka rohovosti pro všechny body,
- výběr vyhovujících bodů,
- znázornění výsledného obrázku.

V následující části je celý postup podrobněji rozebrán.

Před vlastním výběrem možných významných bodů je potřeba upravit obrázek na vstupu algoritmu. Po načtení snímku pomocí samostatného modulu mdsLoadX, kde X reprezentuje jeden ze tří podporovaných typů obrázků, dostáváme na vstupu obrázek ve stupních šedi při 12bitové hloubce.

Obr. 4.2 Lena - původní

Abychom se vyvarovali problémů s nedostatečným rozsahem při použití souřadnic s desetinnou řádovou čárkou, převedeme si jednotlivé pixely do rozsahu $\langle -1;1 \rangle$. Nejprve se zjistí maximální a minimální intenzity, které představují hranice uzavřeného intervalu. Pomocí těchto intenzit se přepočítají intenzity zbývajících pixelů. Při další manipulaci s hodnotami pixelů se nedostaneme mimo rozsah typu float a nedojde tudíž ke ztrátě informace při jejich zaokrouhlování. Po převedení do výše zmíněného rozsahu je aplikován na obrázek Gaussův filtr. Pro zabránění výběru bodu mimo obrázek jsou duplikovány krajní body tohoto obrázku. Po tomto zkopírování krajních bodů se nemusíme obávat, že bychom při výpočtu narazili na bod, který by se nacházel mimo obrázek. Princip tohoto kopírování krajních bodů je vidět na obrázku 4.3.



Obr 4.3 Vytvoření nového okraje snímku

Obr. 4.4 Lena – Gaussův filtr

Na takto vzniklý obrázek je zavolána funkce, která provede výpočet „rohovosti“ v jednotlivých bodech. V této funkci jsou spočítány pro každý bod všechny druhé derivace, ze kterých je následně spočítána rozhodovací funkce podle zvolené modifikace metody. Těchto modifikací je v současné době sedm a jsou podrobně rozebrány v podkapitole 3.6 a 3.7. Hodnoty rozhodovací funkce jsou uloženy do pomocného obrázku.

Z pomocného obrázku je vybrána maximální hodnota, která je použita při určení prahové hodnoty, na základě které se rozhoduje, zda by se mohlo jednat o významný bod či nikoli. Výpočet prahové hodnoty spočívá ve vynásobení maximální hodnoty vhodnou konstantou. Poté je proveden výběr hodnot, které jsou větší než vypočítaná prahová hodnota, z pomocného obrázku. Prahová hodnota je použita pro zefektivnění časové náročnosti algoritmu.

Obr. 4.5 Lena - nonmaximal

Následuje výběr lokálních maxim, kde hranice oblastí, které prohledáváme, jsou vypočítány pomocí hodnoty σ použité při aplikaci Gaussova filtru.

Obr. 4.6 Lena – rohové body

Takto vybrané body představují hledané významné body. Pro jejich zvýraznění a lepší čitelnost výstupního obrázku je použita funkce, která na místě nalezeného významného bodu vykreslí kontrastní kříž. Pro zpracování výstupu algoritmu je použit samostatný modul mdsSliceView.

Obr. 4.7 Lena – zvýrazněné rohové body

U všech výstupních obrázků, které jsou výsledkem vyhledávání, je patrná diagonální přímka. Je to dáno způsobem vykreslování obrázků v MDSTk. Nejde tedy o detekování nadbytečných bodů, jak by se na první pohled mohlo zdát.

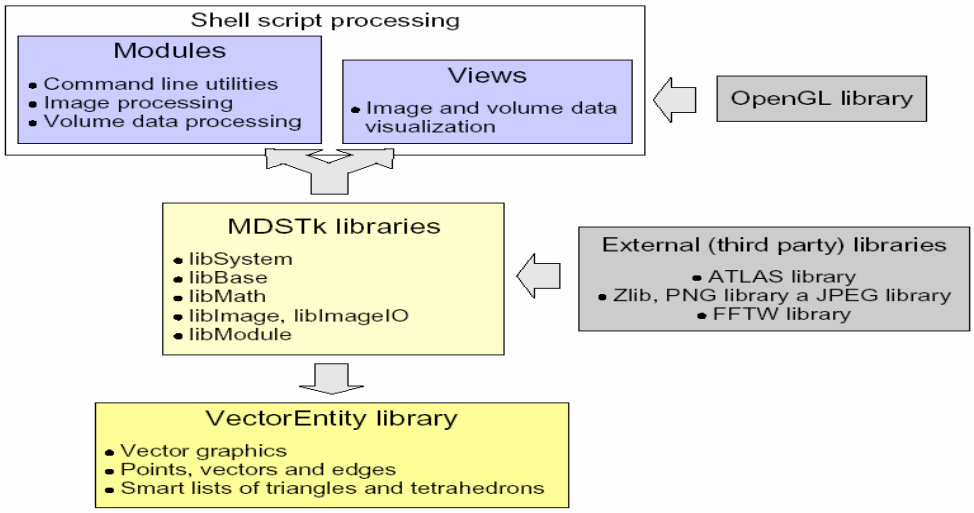
5 Implementace

Program byl implementován v jazyce C/C++ s využitím knihoven toolkitu MDSTk. Bylo použito vývojové prostředí MS Visual Studio 2005.

Vlastní implementace

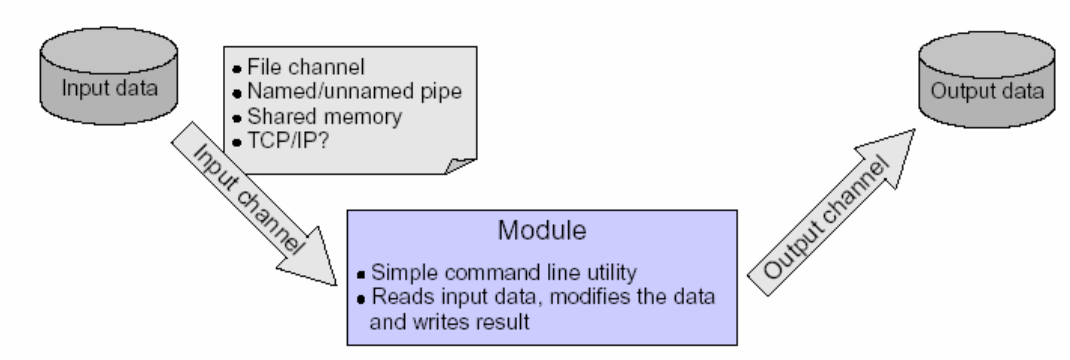
V úvodu implementace je nejprve zapotřebí předzpracovat načtený vstupní obrázek. Nejprve si pomocí předdefinované funkce Convert převedeme obrázek do souřadnic s desetinnou řádovou čárkou a poté aplikujeme Gaussův filtr. Zavoláme funkci HarrisCorners, která nám jako výsledek vrátí pole hodnot, které nám poslouží jako rozhodovací měřítko rohovosti. Toto měřítko se odvíjí od zvolené modifikace metody. Pole představuje vstup funkce hysteresis, která nejprve tyto hodnoty porovná s vypočítanou prahovou hodnotou T, určí rozsah oblasti pro výběr lokálních maxim a následně vybere lokální maxima v oblastech, kde je prahová hodnota větší než T. Ostatním bodům přiřadí nejmenší možnou hodnotu. Pro body, kde se nachází lokální maxima, zavoláme funkci drawCross, která vykreslí kříž na určených souřadnicích. Výpočet lokálních maxim je v použité implementaci časově náročný a neefektivní, protože se jedná o čtyři vnořené for cykly. Existuje efektivnější výpočet lokálních maxim, kde se používá pomocná funkce, která určí lokální maximum na základě operátoru ||. Tento druhý způsob je nesporně časově úspornější, ale zápis kódu je v tomto případě nepřehledný a navíc jde o formát nevhodný pro experimentování z důvodu obtížné modifikovatelnosti. Poněvadž je cílem implementace přehledný a čitelný program, je použit více názornější způsob se čtyřmi vnořenými for cykly.

Protože je výsledná implementace součástí MDSTk, byly při řešení využívány možnosti, které tento toolkit nabízí. Na obrázku 5.1 je znázorněna struktura toolkitu.



Obr. 5.1 Struktura MDSTk

Důležitou součástí MDSTk jsou moduly, pomocí nichž lze snadno manipulovat se vstupním obrázkem. Následující obrázek ukazuje princip fungování modulu.



Obr. 5.2 Fungování modulu

Pro načtení vstupního obrázku se v MDSTk nachází samostatné moduly mdsLoadDicom, mdsLoadJPEG a mdsLoadPNG. Z názvů modulů je patrné, že MDSTk podporuje tři následující formáty obrázků:

- DICOM (Digital Imaging and Communications in Medicine) [11] – formát obrázku používaný knihovnou MDSTk pro manipulaci s medicínskými obrázky. Hodnoty pixelů jsou omezeny na 0…4095. Jedná se o nevhodný formát při detekování významných bodů, poněvadž dochází k detekování velkého počtu falešných bodů,
- JPEG (Joint Photographic Experts Group) – metoda ztrátové komprese používaná pro ukládání obrázku ve foto realistické kvalitě. Pro detekování významných bodů se nejedná o vhodný formát, poněvadž obrázky JPEG obsahují viditelné a rušivé artefakty, které mohou zapříčinit chybné určení bodů jako významných. Použití správné prahové hodnoty výsledek zlepší, ale nedosahuje se takových výsledků jako u formátu PNG,
- PNG (Portable Network Graphics) [9]– metoda bezeztrátové komprese rastrové grafiky. Bitová hloubka jednotlivých pixelů se pohybuje v rozmezí 1 až 48. Z hlediska detekování významných bodů jde o nejlepší z podporovaných formátů, poněvadž se v obrázku nachází pouze ostré hrany. Je možné použít stupně šedi spolu s alpha kanálem (úroveň průhlednosti každého pixelu).

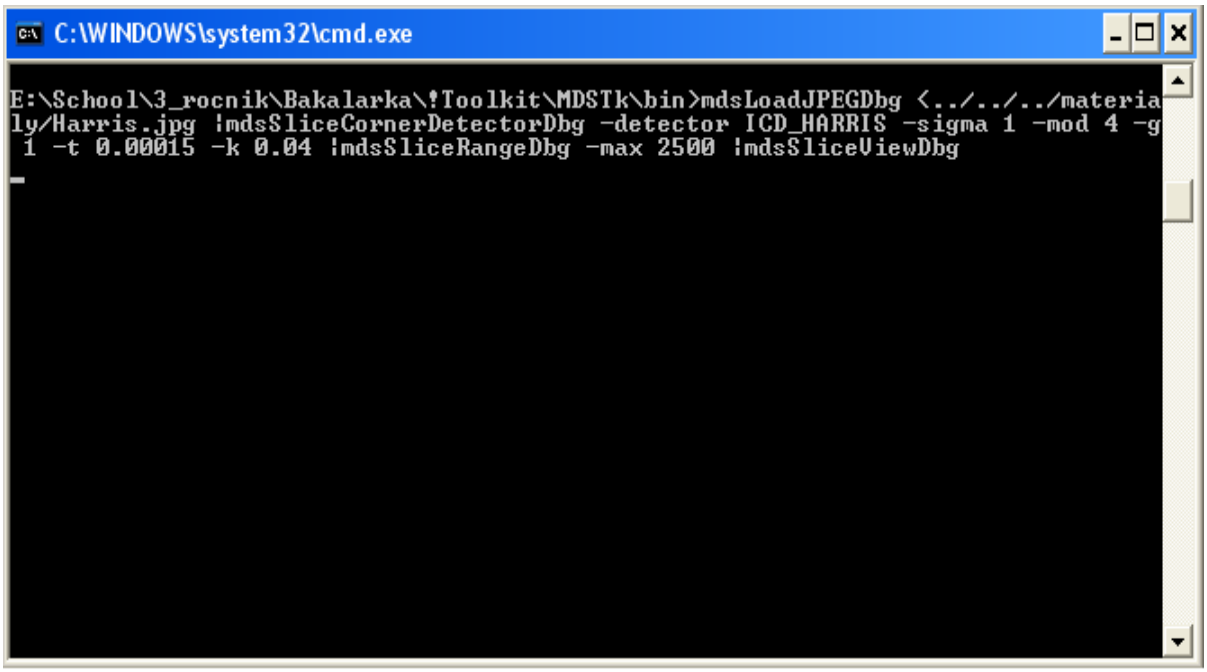
Spuštění

K manipulaci se snímky slouží příkazový řádek, kde pomocí jednotlivých modulů dochází k načtení zkoumaného snímku, nalezení významných bodů podle zvolené metody a vykreslení snímku na standardní výstup. Vstupem do modulu je obrázek, který je v modulu zpracován a následně jsou získané významné body zapsány do výstupního obrázku. K přechodu mezi jednotlivými moduly se používá bezejmenná pipe. Na následujícím příkazu si ukážeme jednotlivé složky navrženého modulu

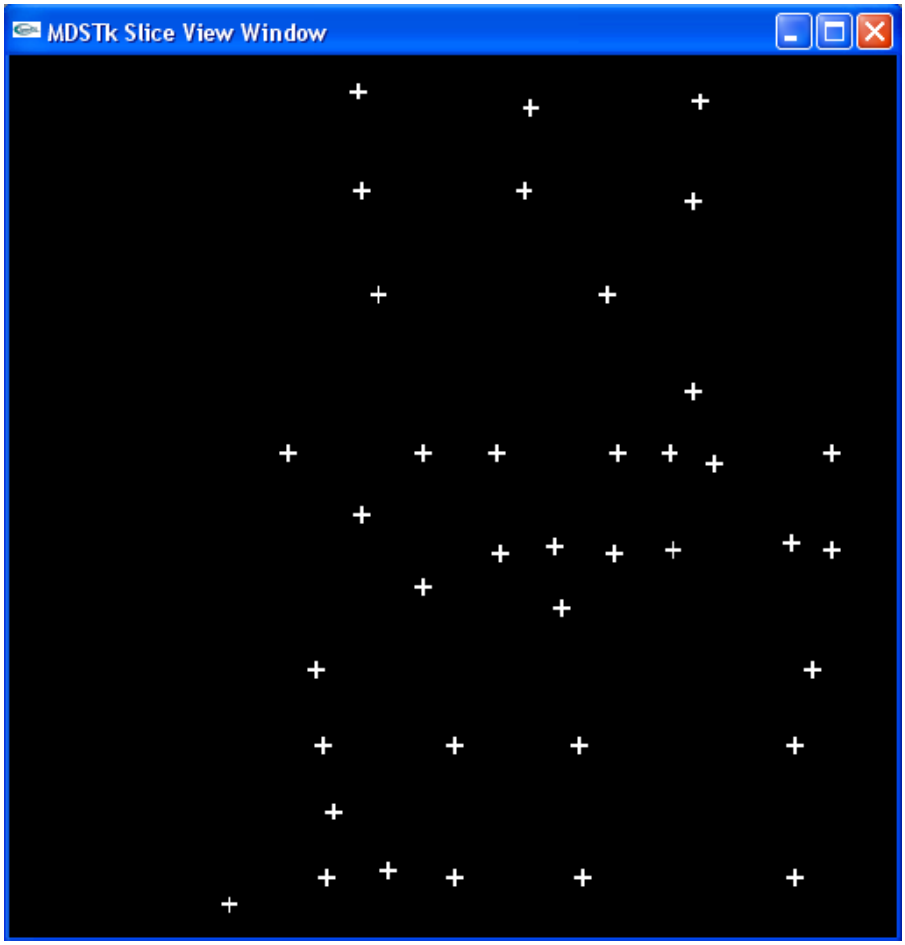
mdsSliceCornerDetector. Na obrázku 5.3 můžeme také pozorovat zpracovávání obrázku v sekvenci příkazů.

Obr. 5.3 Průběh zpracování obrázku

Projdeme si všechny parametry modulu mdsSliceCornerDetectorDbg. Koncovka Dbg v názvu modulu znamená, že byl projekt ve Visual Studiu přeložen v módu Debug. Za parametrem *detector* následuje vybraná metoda detekce významných bodů, v současné době je funkční pouze metoda Harris Stephens. Parametr *sigma* nám říká, jaká je odchylka pro průchod Gaussovým filtrem. Zároveň se z tohoto parametru odvozuje i velikost oblasti pro výběr lokálních maxim, tradičně se používá šestinásobku až osminásobku σ. Parametr *mod* označuje vybranou modifikaci algoritmu Harris Stephens. Za parametrem *g* se nachází zvolená metoda při výpočtu gradientů. Za parametrem *t* je zadaná prahová hodnota. Posledním parametrem je *k*, které určuje citlivost rozhodovací funkce u nemodifikované metody Harris Stephens. Příkaz může být brán jako příklad spuštění programu, za předpokladu, že je nastavena správná cesta ke spustitelným modulům a je zadaná správná cesta ke vstupnímu obrázku. Nejprve je ovšem nutné nainstalovat toolkit MDSTk. Podrobný návod, jak provést instalaci toolkitu, je v příručce k MDSTk [7]. Pokud jsou zadané parametry mimo rozsah, dojde k ukončení programu a na obrazovce se objeví výzva k využití nápovědy, kde jsou uvedeny aktuální rozsahy všech používaných parametrů. Nápovědu lze spustit příkazem *mdsSliceCornerDetectorDbg –h* .



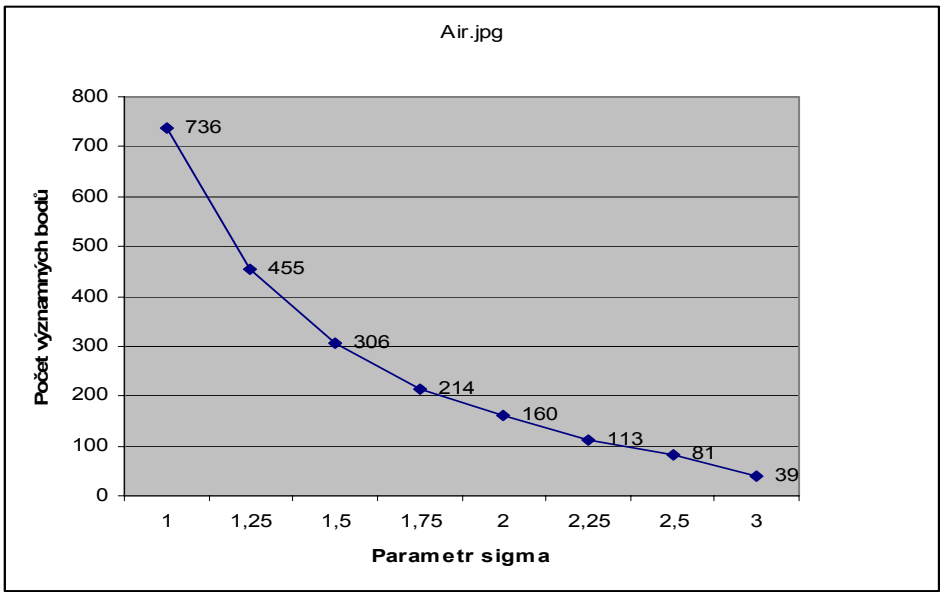
Obr. 5.4 Příklad spuštění



Obr. 5.5 Výstupní obrázek

6 Výsledky

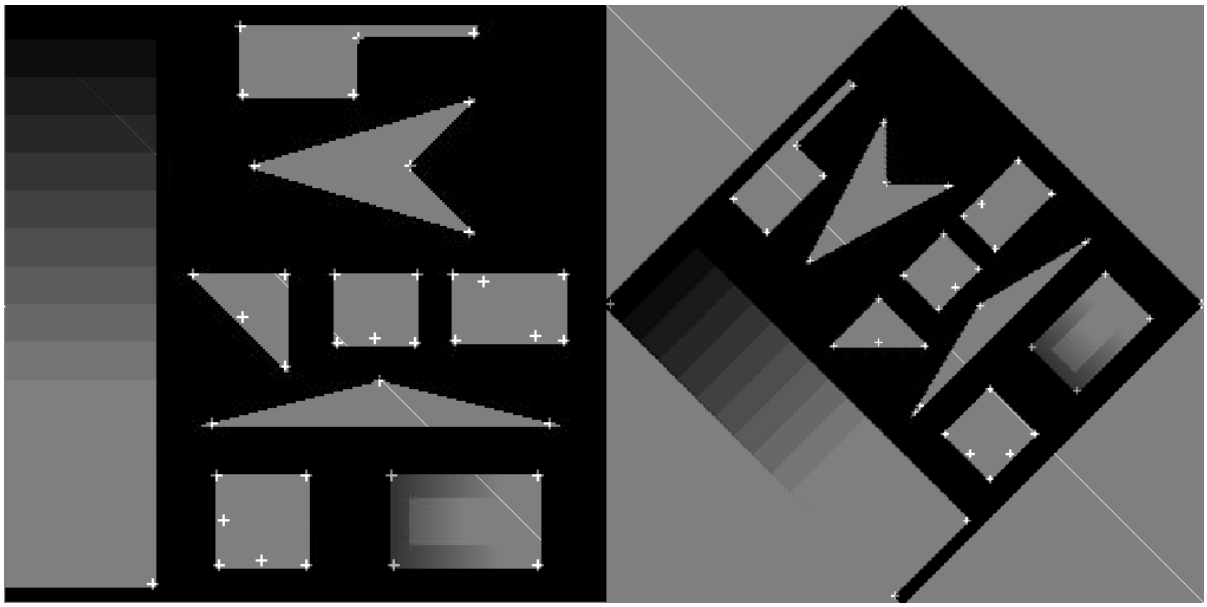
Algoritmus Harris Stephens nabízí široký prostor pro experimentování. Je to patrné již při pohledu na příkazový řádek, kde vidíme spoustu parametrů, které ovlivňují zásadním způsobem podobu výsledného řešení. Prvním s parametrů, které slouží k optimálnímu nastavení výstupu je *sigma*. Tato hodnota ovlivňuje výsledné řešení nejlépe při hodnotě jedna. Čím je hodnota *sigma* větší, tím je efekt rozmazání obrázku viditelnější. S rostoucí hodnotou *sigma* se také snižuje počet detekovaných významných bodů, protože se zvětšuje oblast, ve které dochází k výběru lokálních maxim. Úbytek bodů můžeme pozorovat v grafu 6.1. Výsledná optimální hodnota byla ověřena, stejně jako všechny ostatní, důkladným zkoumáním hodnot v celém jejím rozsahu.



Graf 6.1 Závislost řešení na parametru sigma

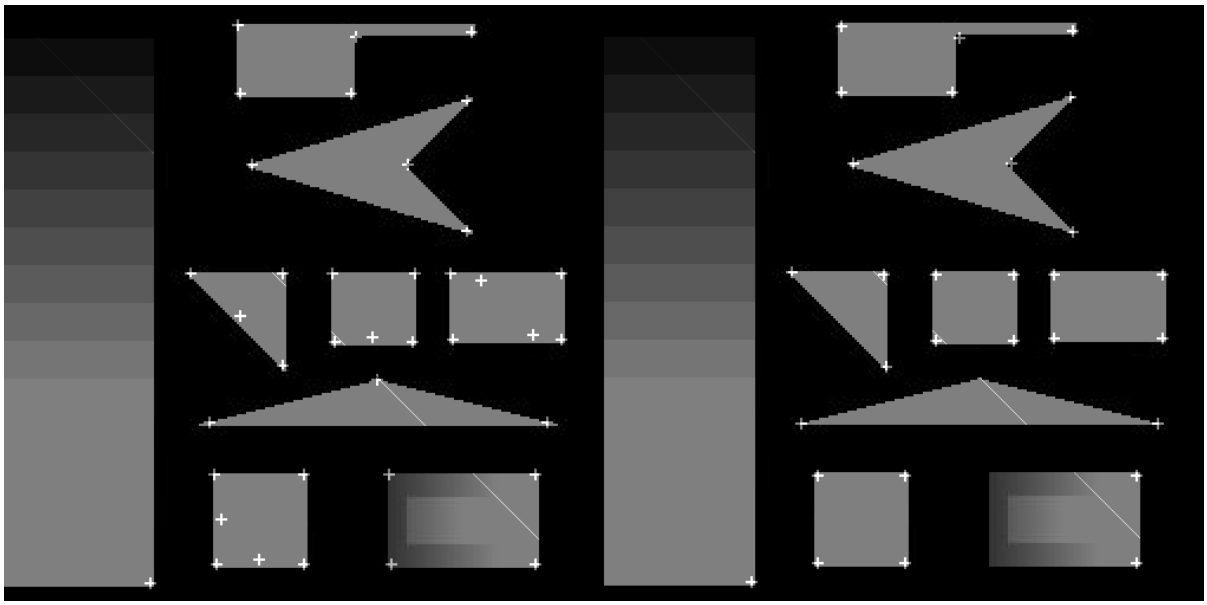
Dalším parametrem je *mod*. Pro obrázky v menším barevném spektru bylo nejlepších výsledků dosaženo pro *mod* = 4, což je výše popsaná metoda Harris Stephens bez modifikací. Při experimentování s digitálními fotografiemi se osvědčili všechny implementované metody. Nejhorších výsledků dosahuje modifikace Haralick, kde dochází k nepřesnému detekování vybraných rohových bodech v L-uzlech.

Experimenty s parametrem *g* proběhly rychle, poněvadž tento parametr může nabývat pouze hodnot jedna resp. dva. Obě naimplementované metody vykazují podobně kvalitní výsledky. Z důvodu zařazení dalšího možného způsobu výpočtu gradientů je vymezen pro hodnoty *g* větší rozsah. Pokud by však byla zadaná jiná hodnota než jedna či dvě, došlo by buď k ukončení programu anebo k použití standardní hodnoty výpočtu gradientů tj. *g* = 1. Na obrázku 6.1 můžeme vidět izotropní vlastnosti použitých metod pro určování gradientů při současném použití Gaussova filtru.



Obr. 6.1 Významné body při rotaci o 45°

Postupným obměňováním hodnot u parametru k bylo zjištěno, že se nejlepších výsledků dosahuje pro hodnotu k = 0.04. Podle dostupných materiálů by měla být hodnota parametru $k = 0.04 \ldots 0.15$. V případě navržené implementace se jeví být optimální hodnoty v rozmezí $k = 0.04 \ldots 0.10$. Pro vyšší hodnoty k dochází k neúplnému detekování správných významných bodů. Na obrázku 6.2 můžeme pozorovat úbytek detekovaných významných bodů při zvyšování hodnoty k.



Obr. 6.2 Výstup pro hodnoty $k = 0.04$ a $k = 0.15$

Posledním nastavitelným parametrem je prahová hodnota t. Experimenty bylo zjištěno, že se prahová hodnota liší s modifikací algoritmu Harris Stephens použitou při výpočtu. V tabulce 6.1 je znázorněn přehled doporučených hodnot parametru t pro jednotlivé modifikace algoritmu.

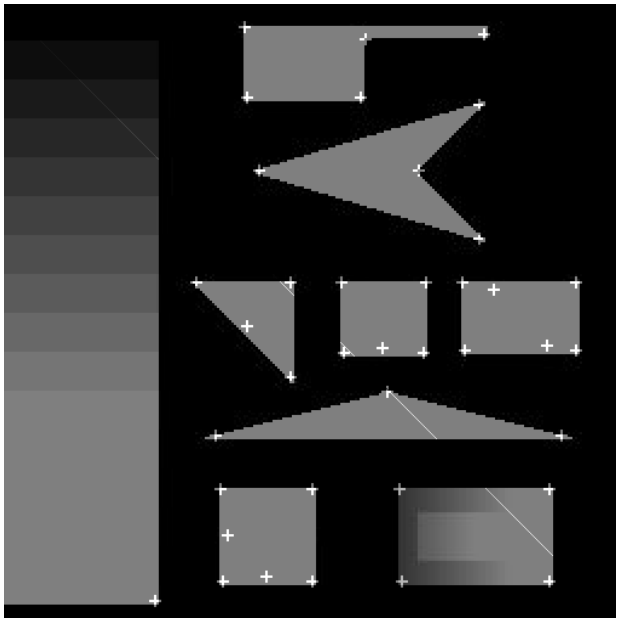
MOD	METHOD	MIN THRESHOLD	OPT THRESHOLD	MAX THRESHOLD
1	Shi and Thomasi	0.01	0.013	0.2
2	Forstner - eigenvalues	0.001	0.0107	0.2
3	Forstner operator	0.001	0.0107	0.2
4	Harris Stephens	0.0001	0.00015	0.2
5	Harris Stephens - eig.val.	0.0001	0.00015	0.2
6	Haralick	0.0001	0.0012	0.2
7	Venera's acceleration	0.00001	0.000015	0.02

Tab. 6.1 Prahové hodnoty.

Při zadání prahových hodnot, které jsou mimo rozsah, dojde k použití výchozích hodnot. Tyto hodnoty jsou vyznačeny v tabulce 6.1 ve sloupci OPT THRESHOLD. Standardní nastavení parametrů vypadá následovně: $\sigma = 1, \mathrm{mod} = 4, g = 1, t = 0.00015, k = 0.04$.

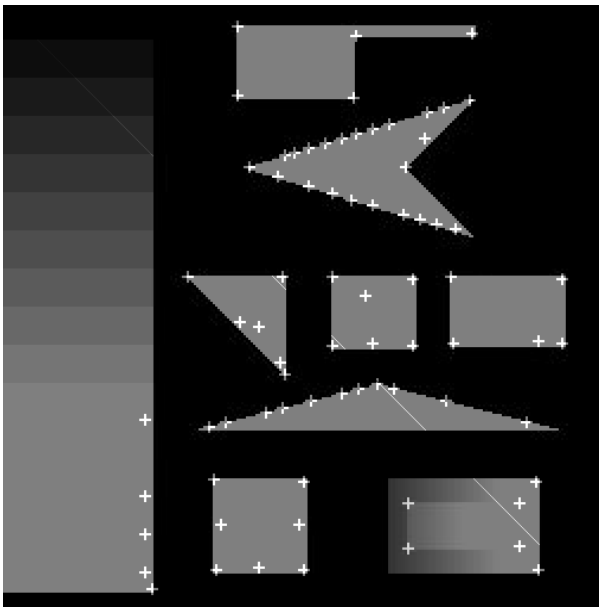
6.1 Úloha Gaussova filtru

Při experimentování s výslednou implementací se podařilo odhalit citlivost algoritmu na použití Gaussova filtru. Pokud vstupní obrázek projde Gaussovým filtrem, pak řešení vypadá jako na obrázku 6.3.



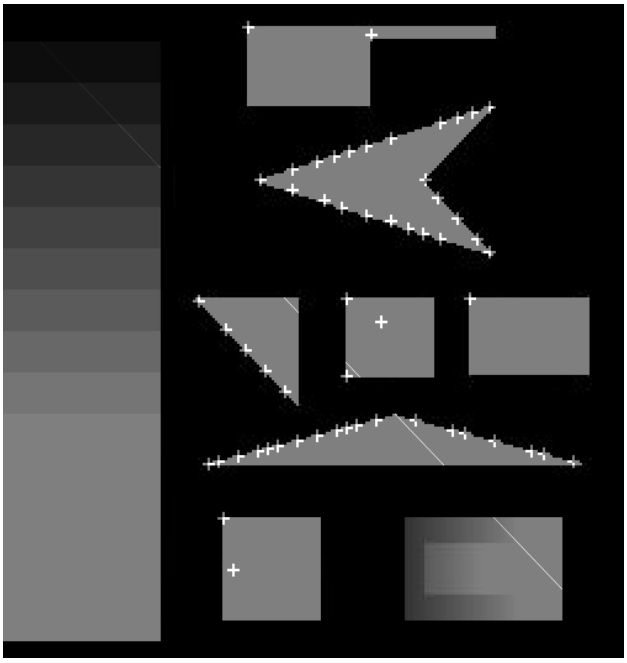
Obr. 6.3 Standardní výsledky.

Pokud program spustíme se stejnými parametry a na vstupní obrázek Gaussův filtr neaplikujeme, vypadá řešení následovně.



Obr. 6.4 Bez použití Gaussova filtru.

Můžeme pozorovat nárůst falešných rohových bodů, které program detekuje. Pokud navíc použijeme ještě standardní vzorec pro výpočet gradientů, kde k výpočtu používáme použijeme zkoumaný bod a jeho levého souseda, pak řešení odpovídá obrázku 6.5.



Obr. 6.5 Použití sousedních bodů pro výpočet gradientů.

Vidíme, že při použití Gaussova filtru vykazuje program nesrovnatelně lepší výsledky než při jeho vynechání. Použitím Gaussova filtru se podařilo odstranit detekování falešných rohů na hranách. Gaussův filtr má navíc pozitivní vliv i na získávání izotropních výsledků.

7 Závěr

Práce se zabývá detekováním významných bodů v obraze. Její podstatná část je věnována algoritmu Harris Stephens, který je v dnešní době jedním z nejpoužívanějších algoritmů. Součástí práce je také implementace tohoto algoritmu a zakomponování metody do MDSTk.

Cílem mého snažení bylo vytvořit program, který bude použit v knihovně MDSTk jako modul pro detekování významných bodů. Dle mého názoru se podařilo vyvinout fungující aplikaci splňující všechna základní kritéria. Program plní svou funkci, dosažené výsledky jsou odpovídající. Podařilo se odhalit závislost výsledků programu na použití Gaussova filtru. Pokud je tento aplikován, dochází k výraznému zlepšení výsledků metody Harris Stephens. Algoritmus se stává více konkurenceschopným. Implementaci je potřeba zoptimalizovat z hlediska časové náročnosti. Dále by bylo vhodné zautomatizovat nalezení optimální prahové hodnoty v závislosti na vstupním obrázku. Zajímavým rozšířením by bylo převedení algoritmu do 3D, což by umožnilo detekci významných bodů v objemových datech. Protože existuje vícero principů, na jejichž základě dochází k detekci významných bodů, bylo by vhodné obohatit stávající modul pro detekování významných bodů o další algoritmy. V započaté práci bych rád pokračoval a chtěl bych rozšířit možnosti MDSTk zakomponováním Trajkovic Hedley algoritmu a algoritmu Wang Brady do modulu pro detekování významných bodů. S větším počtem implementovaných metod, které pracují na vzájemně odlišných principech, vzroste zajisté i úspěšnost detekování významných bodů na správných pozicích.

Literatura

[1] Harris, C., Stephens, M. A combined corner and edge detektor. Proc. of the 4th ALVEY vision conference, M. M. Matthews, Ed., University of Manchester, England, Septemeber 1988, pp. 147–151.
Dokument dostupný na URL
http://www.csse.uwa.edu.au/~pk/research/matlabfns/Spatial/Docs/Harris/A_Combined_Corner_and_Edge_Detector.pdf

[2] Moravec, H. Obstacle avoidance and navigation in the real world by a seeing robot rover. Carnegie-Mellon University, Robotics Institute, Tech. Rep. CMU-RI-TR-3, September 1980.
Dokument dostupný na URL
http://www.ri.cmu.edu/pub_files/pub4/moravec_hans_1980_1/moravec_hans_1980_1.pdf

[3] Shen, F., Wang, H. Real time gray level corner detector. School of Electrical & Electronic Engineering Nanyang Technological University, p.6.
Dokument dostupný na URL http://www.ntu.edu.sg/home/hw/ShenWangIcarcv2000.pdf

[4] Smith, S., M., Brady, J.m M. A New Approach to Low Level Image Processing. Int. J. of Comp. Vision, 23(1), 45-78, 1997.
Dokument dostupný na URL http://citeseer.ist.psu.edu/smith95susan.html

[5] Shi, J., Tomasi, C. Good Features to Track. 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer, June 1994.
Dokument dostupný na URL http://citeseer.ist.psu.edu/shi94good.html

[6] Venera, J. Metody detekce rohových bodů vhodné pro HW akceleraci.
Dokument dostupný na URL http://cpg.felk.cvut.cz/tiki-download_file.php?fileId=6

[7] Španěl, M. MDSTk_A_Brief_Guide. Faculty of Information Technology, Brno, November 2006.
Dokument dostupný na URL
http://www.fit.vutbr.cz/~spanel/mdstk/download/guide/MDSTk_A_Brief_Guide_v2.zip

[8] Corner detection. (2007, January 27). In *Wikipedia, The Free Encyclopedia*. Retrieved April 20, 2007, from http://en.wikipedia.org/w/index.php?title=Corner_detection&oldid=103500259

[9] *Wikipedie: Otevřená encyklopedie: Portable Network Graphics* [online]. c2007 [citováno 20. 04. 2007]. Dostupný z WWW:
http://cs.wikipedia.org/w/index.php?title=Portable_Network_Graphics&oldid=1359939

[10] Parks, D., Gravel, J. *Corner Detectors* [online]. 2000 [cit. 2007-04-20]. Dostupný z WWW: http://www.cim.mcgill.ca/~dparks/CornerDetector/comparison.htm

[11] Rorden, Ch. *Dicom introduction* [online]. 2005 [cit. 2007-04-20]. Dostupný z WWW: http://www.sph.sc.edu/comd/rorden/dicom.html

[12] Fisher, B., et al. *Digital Filters : Spatial filters* [online]. 1994 [cit. 2007-04-20]. Dostupný z WWW: http://www.cee.hw.ac.uk/hipr/html/gsmooth.html

[13] *Wikipedie: Otevřená encyklopedie: Gradient* [online]. c2007 [citováno 20. 04. 2007]. Dostupný z WWW: http://cs.wikipedia.org/w/index.php?title=Gradient&oldid=1157325

[14] Dias, P. *Haralick corner detector* [online]. 05/11/2003 [cit. 2007-04-20]. Dostupný z WWW: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS2

Seznam zkratek a symbolů

MDSTk - Medical data segmentation toolkit

DICOM - Digital Imaging and Communications in Medicine

JPEG - Joint Photographic Experts Group

PNG - Portable Network Graphics

SSD - sum of squared differences

C++

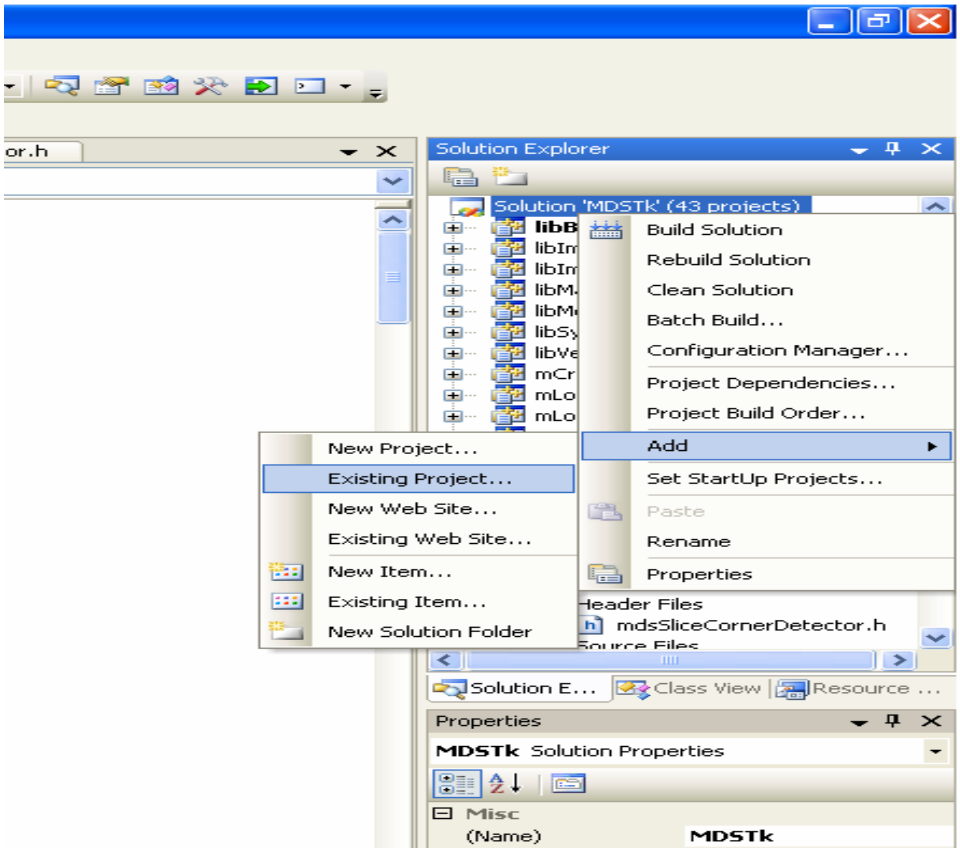
Seznam příloh

Příloha 1. Instalace MDSTk a přidání modulu pro detekování významných bodů

Instalace probíhá ve dvou krocích. Jedná se o instalaci MDSTk podle příručky MDSTk_A_Brief_Guide [7] a zakomponování souborů do MDSTk. Po úspěšné instalaci MDSTk je potřeba rozmístit soubory potřebné pro správné fungování modulu do odpovídajících adresářů. Rozmístění souborů zahrnuje následující body:

- adresář ImageCornerDetectors je nutné zkopírovat do adresáře ..\MDSTk\include\MDSTk\Image\
- soubory mdsImageCornerDetector.h a mdsImageCornerDetectors.h musí být nakopírovány do adresáře ..\MDSTk\include\MDSTk\Image\
- adresář SliceCornerDetector je potřeba zkopírovat do adresáře ..\MDSTk\src\modules\

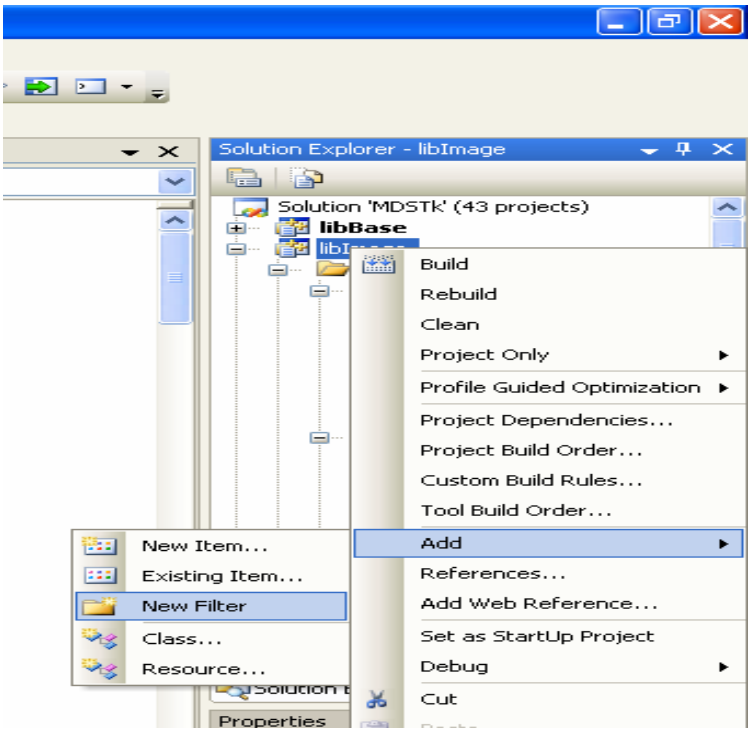
Po rozmístění souborů je nutné zakomponovat tyto do řešení MDSTk. Postupujeme podle následujících obrázků. Přidáme modul SliceCornerDetector z adresáře ..\MDSTk\src\modules\.



Obr. 1 Přidání modulu

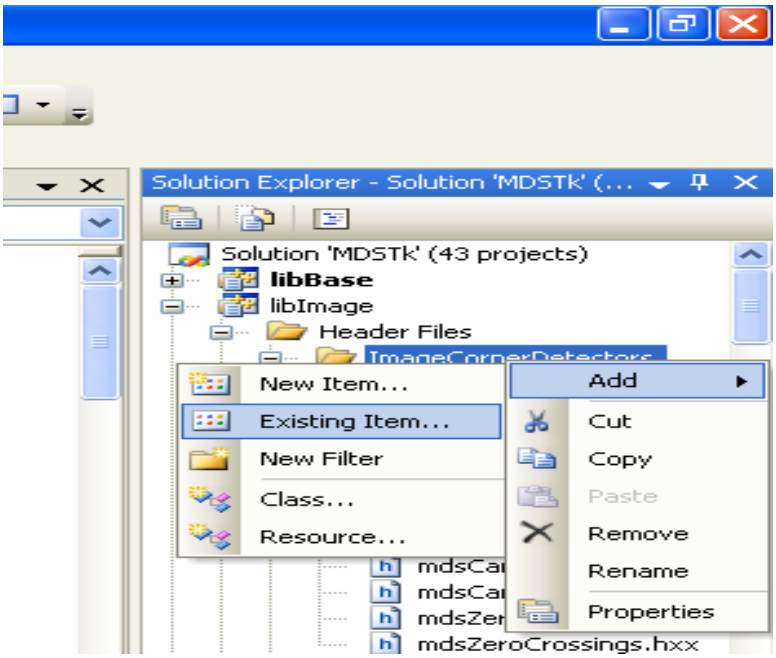
V této fázi můžeme přeložit právě vytvořený modul mSliceCornerDetector a následně jej používat. Pro editaci souborů je nutné vytvoření adresáře ImageCornerDetectors pro následné umístění

souborů. Do knihovny libImage umístíme soubory mdsImageCornerDetector.h a mdsImageCornerDetectors.h z adresáře ..\MDSTk\include\MDSTk\Image\.



Obr. 2 Vytvoření adresáře ImageCornerDetectors

Z adresáře ..\MDSTk\include\MDSTk\Image\ importujeme také soubory mdsHarris.h a mdsHarris.hxx do adresáře ImageCornerDetectors.



Obr. 3 Přidání mdsHarris.h a mdsHarris.hxx

V této fázi je modul přeložitelný i editovatelný.

Příloha 2. Spuštění programu a implementace vyhledávání lokálních maxim

Příklad spuštění programu s vyznačením nalezených bodů na původním obrázku

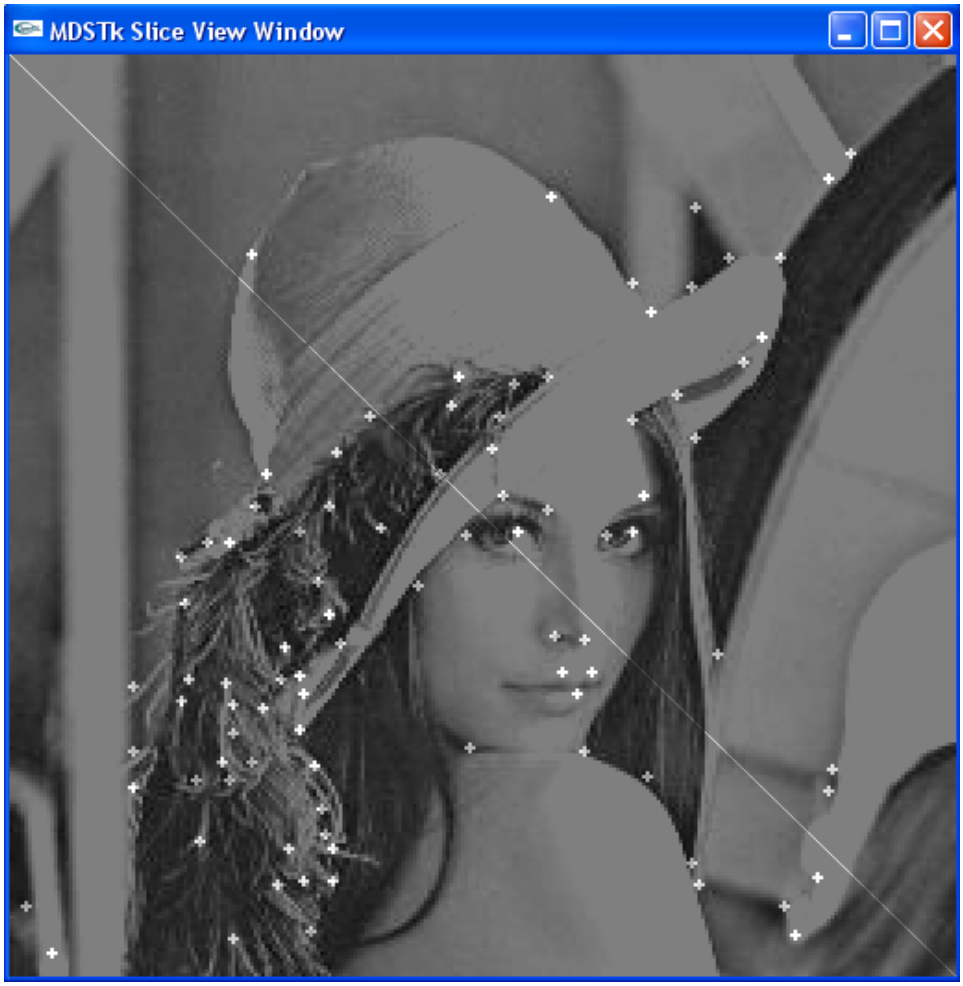
```
C:\WINDOWS\system32\cmd.exe

E:\School\3_rocnik\Bakalarka\!Toolkit\MDSTk\bin>mdsLoadJPEGDbg <../../../materia
ly/Lena.jpg |mdsSliceCornerDetectorDbg -detector ICD_HARRIS >Lena.slc

E:\School\3_rocnik\Bakalarka\!Toolkit\MDSTk\bin>mdsLoadJPEGDbg <../../../materia
ly/Lena.jpg  | mdsSliceRangeDbg -max 2500 >Lena.rslc

E:\School\3_rocnik\Bakalarka\!Toolkit\MDSTk\bin>mdsSliceRangeDbg -max 2500 <Lena
.slc |mdsSliceViewDbg -i stdio::file:Lena.rslc

E:\School\3_rocnik\Bakalarka\!Toolkit\MDSTk\bin>_
```



Implementace vyhledávání lokálních maxim

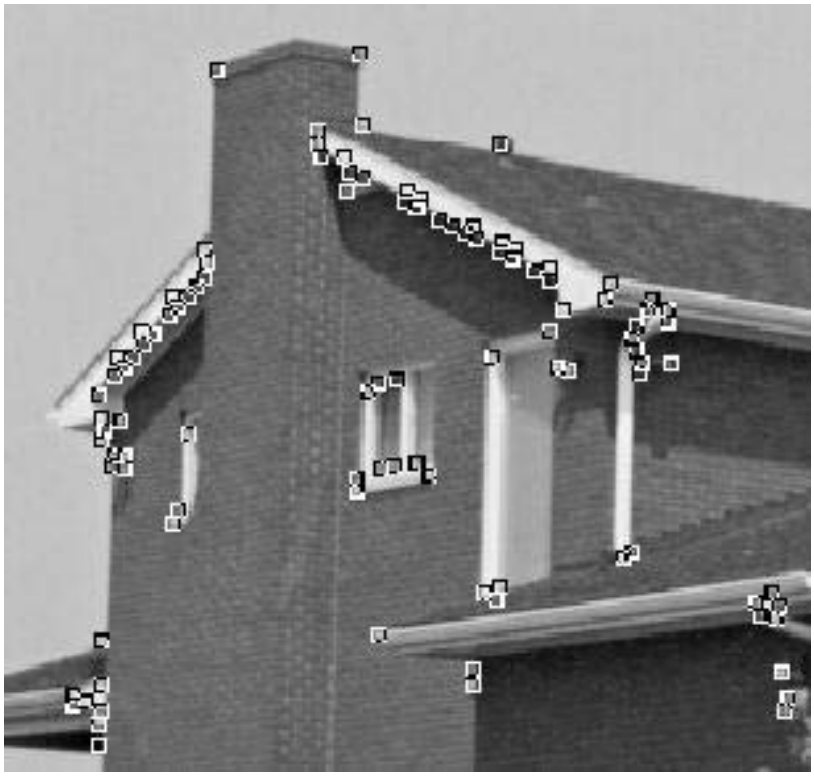
První, časově náročnější, ale přehlednější způsob.

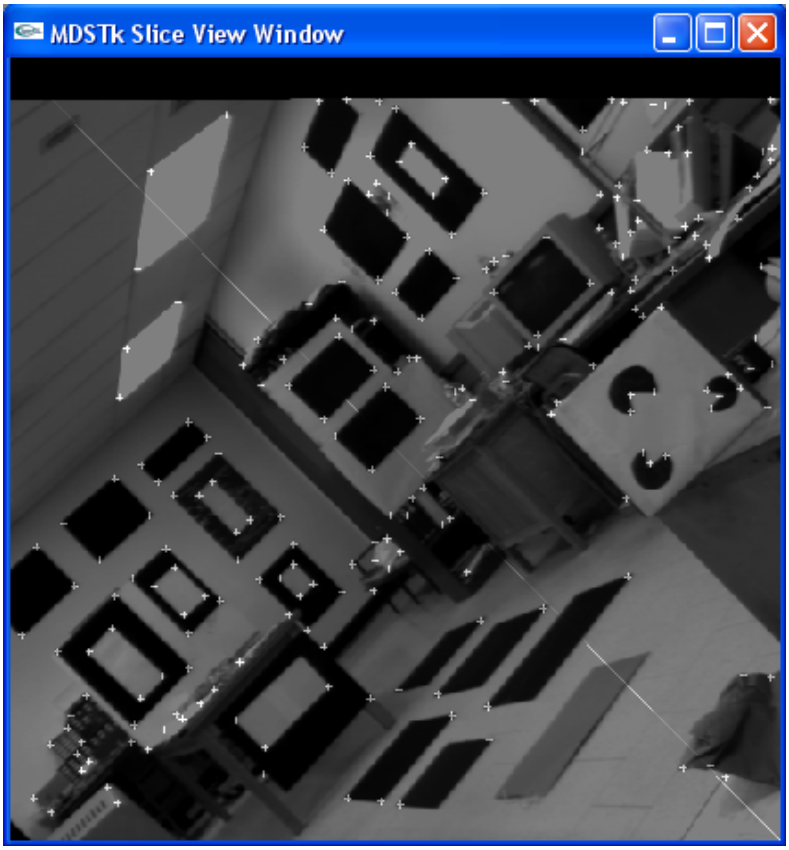
```
// Check the neighbours
for( tSize y = 0; y < YSize; ++y )
{
   for( tSize x = 0; x < XSize; ++x )
   {
for (l = -radius; l < radius; l++) {
         for (r = -radius; r < radius; r++) {
if ((l!=0 || r!=0) && MagnitudeImage.checkPosition(x, y) &&
(MagnitudeImage.get(x + r, y + l) >= MagnitudeImage.get(x, y)))
            CornerImage(x, y) = Minimum;
         }
      }
   }
}
```

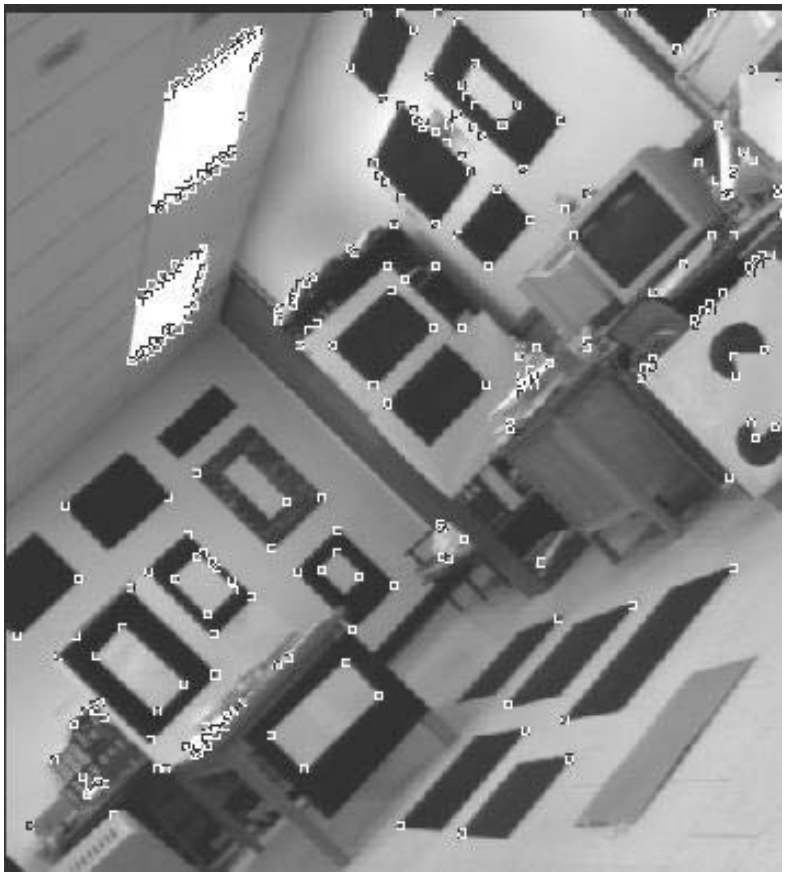
Druhý způsob, který je rychlý, ale implementace je méně přehledná.

```
bool CImageCornerDetector<I, ICD_HARRIS>::checkNeighbours(tImage& Image,
                                                          tSize x,
                                                          tSize y,
                                                          tPixel T
                                                          )
{
    // Volume properties
    tSize XOffset = Image.getXOffset();
    tSize YOffset = Image.getYOffset();
    // Pointer to the pixel
    tPixel *p = Image.getPtr(x, y);
    // Check the neighbours (fixed distance, for testing unsuitable)
    return (   *(p + YOffset) >= T
            || *(p - YOffset) >= T
            || *(p + XOffset) >= T
            || *(p + XOffset + YOffset) >= T
            || *(p + XOffset - YOffset) >= T
            || *(p - XOffset) >= T
            || *(p - XOffset + YOffset) >= T
            || *(p - XOffset - YOffset) >= T
            || *(p + 2*XOffset) >= T
            || *(p + 2*XOffset + YOffset) >= T
            || *(p + 2*XOffset + 2*YOffset) >= T
            || *(p + 2*XOffset - YOffset) >= T
            || *(p + 2*XOffset – 2*YOffset) >= T
            || *(p - 2*XOffset) >= T
            || *(p - 2*XOffset + YOffset) >= T
            || *(p - 2*XOffset + 2*YOffset) >= T
            || *(p - 2*XOffset - YOffset) >= T
            || *(p - 2*XOffset – 2*YOffset) >= T
            || *(p + 2*YOffset) >= T
            || *(p + 2*YOffset + XOffset) >= T
            || *(p + 2*YOffset - XOffset) >= T
            || *(p - 2*YOffset) >= T
            || *(p - 2*YOffset + XOffset) >= T
            || *(p - 2*YOffset - XOffset) >= T
            );
}
```

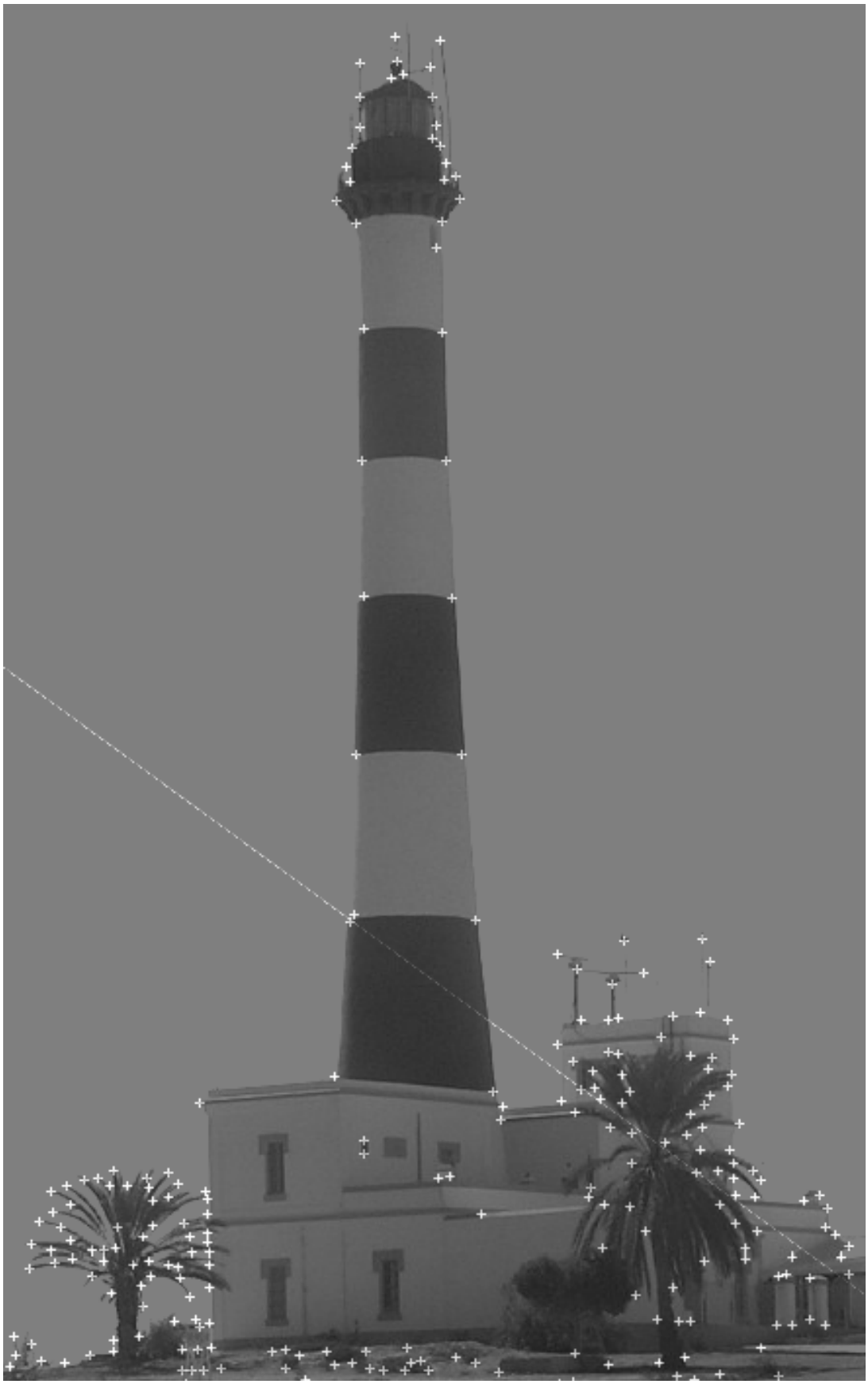
Ukázka vylepšení dosažených výsledků oproti původnímu algoritmu Harris Stephens

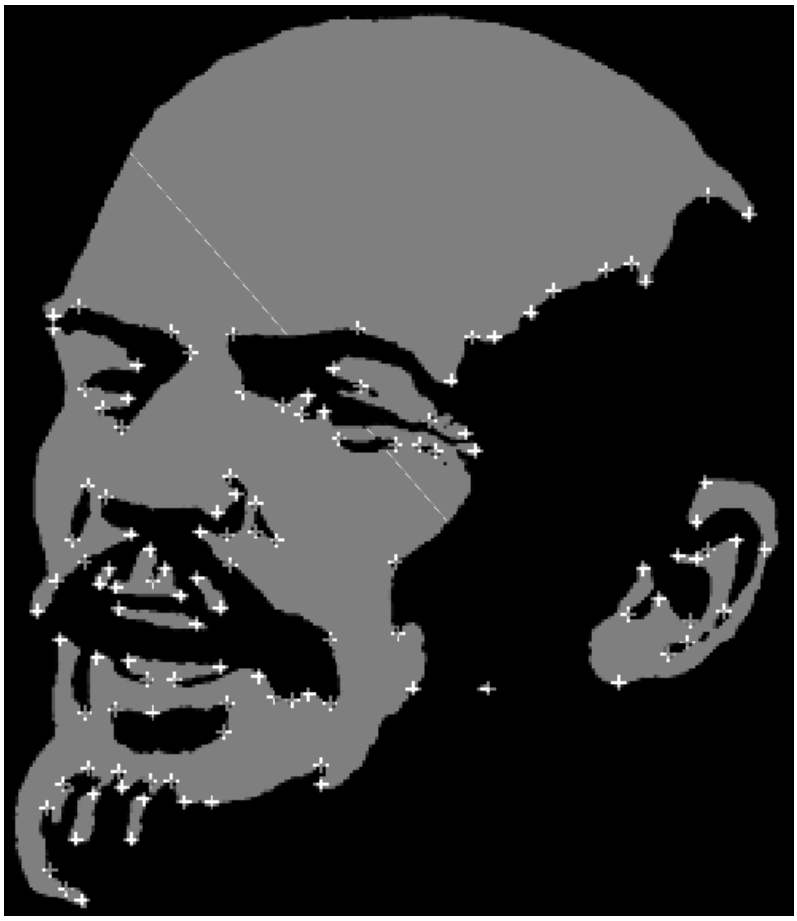


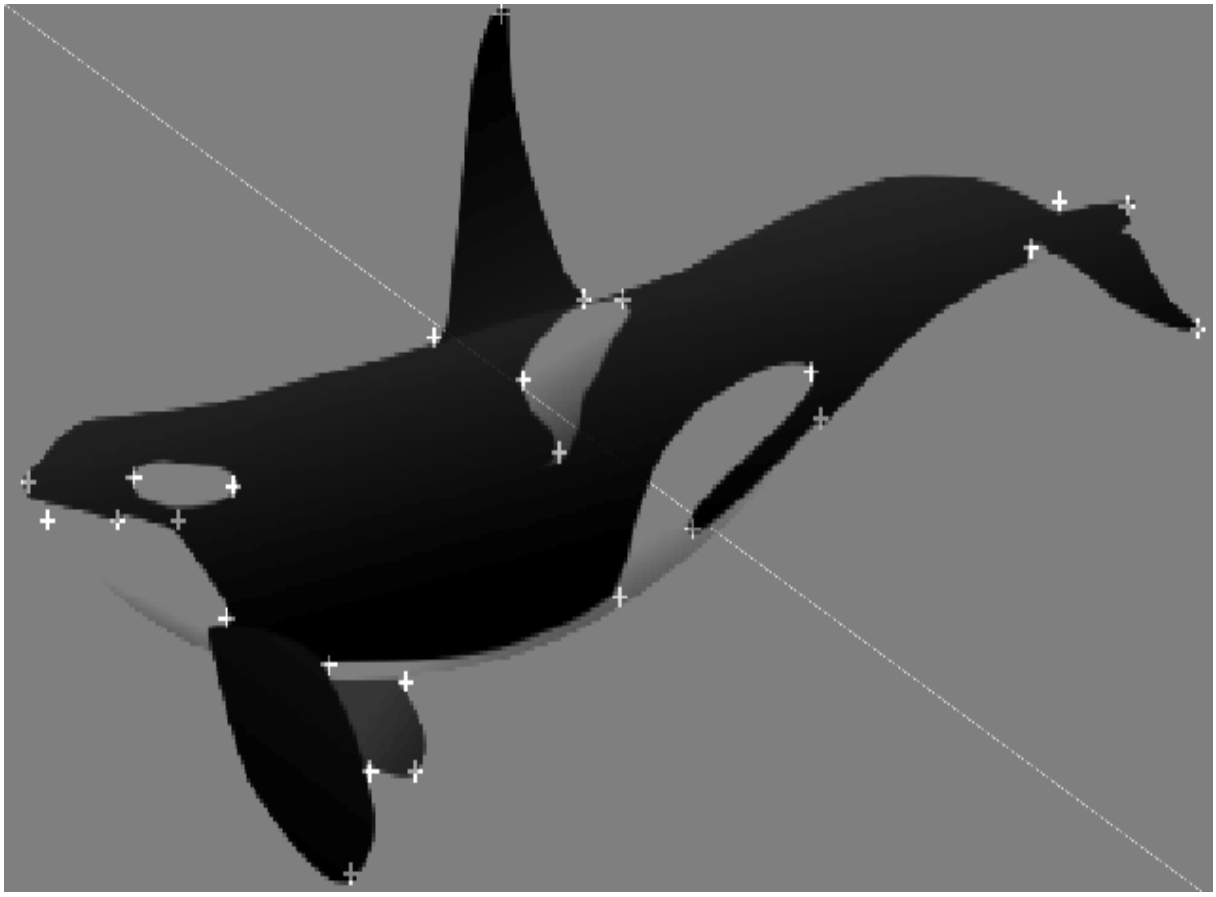

MDSTk Slice View Window

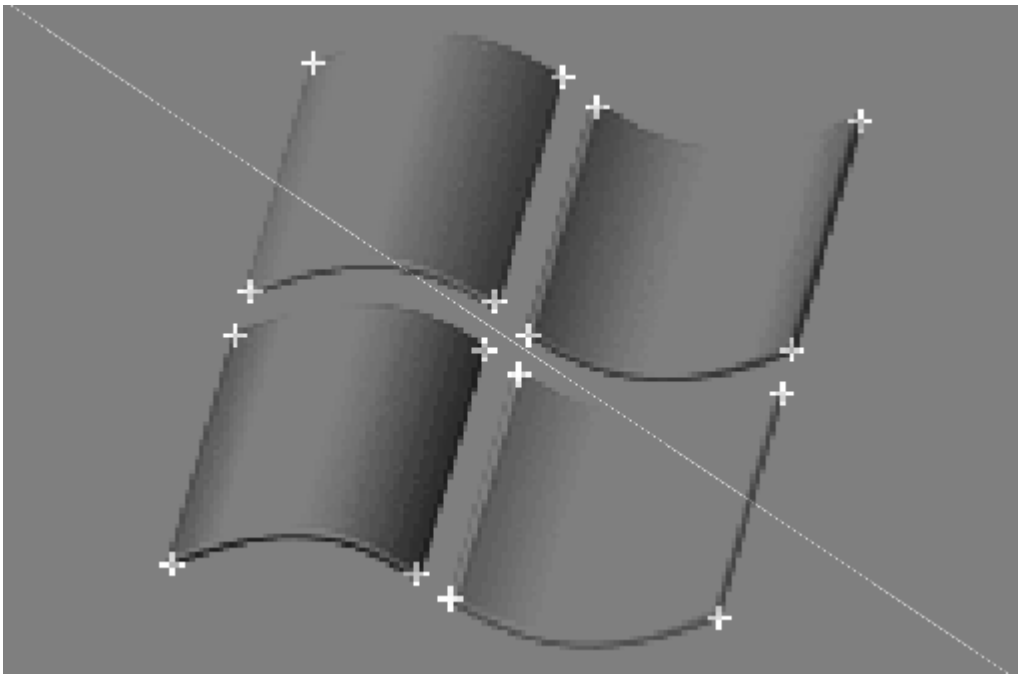


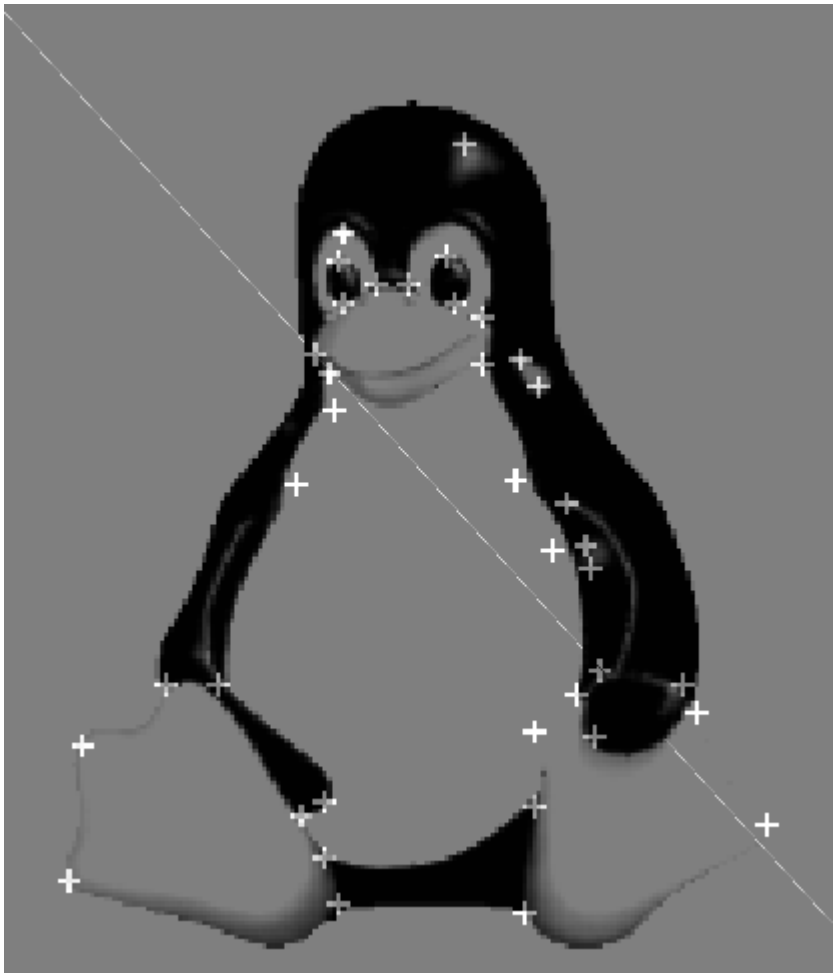
Příloha 3. Výstupní obrázky

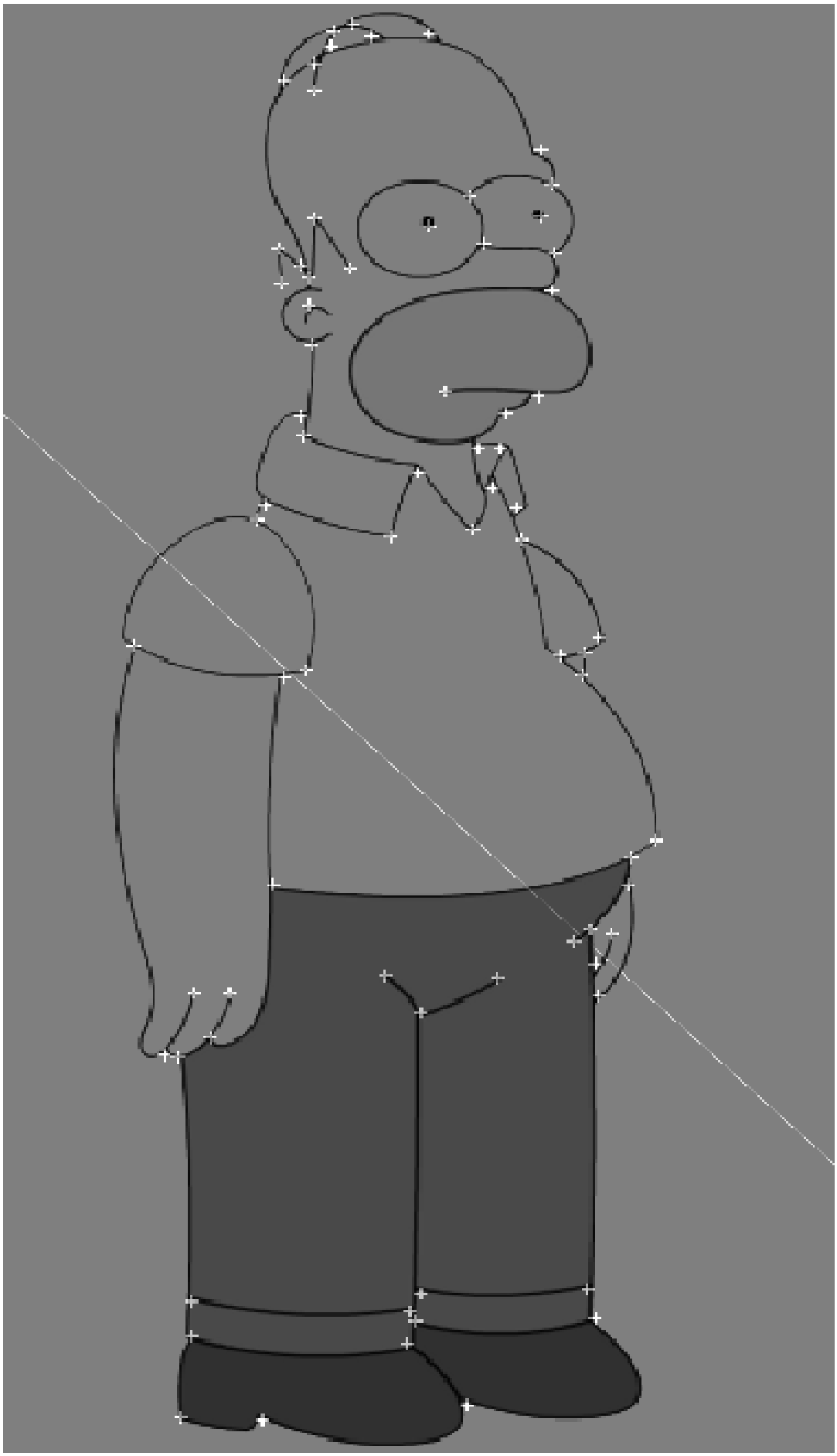


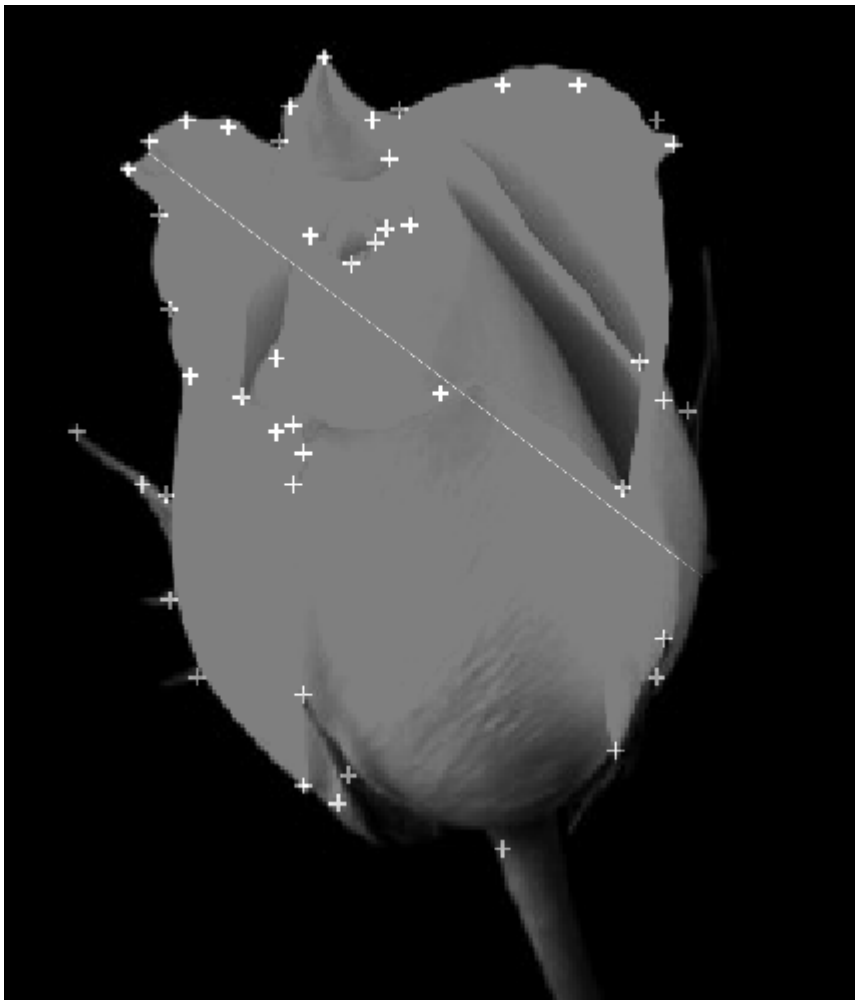


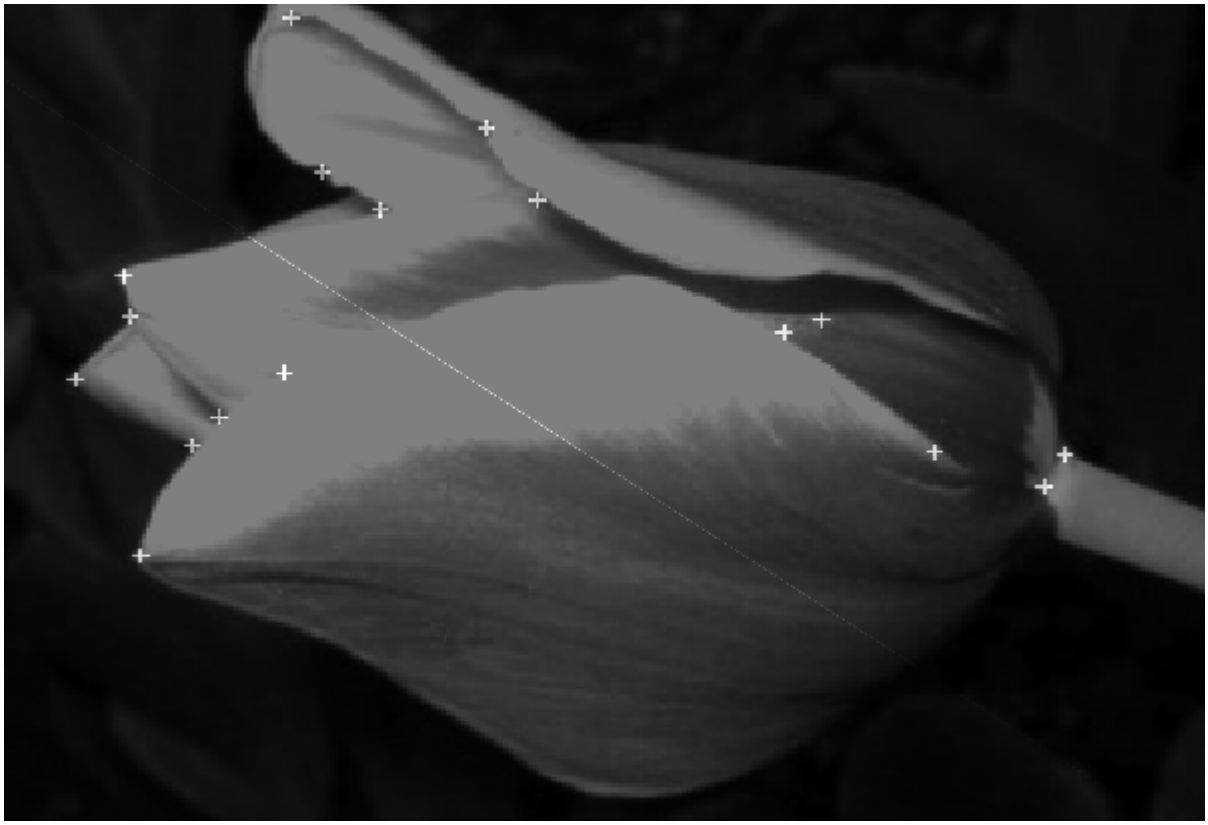


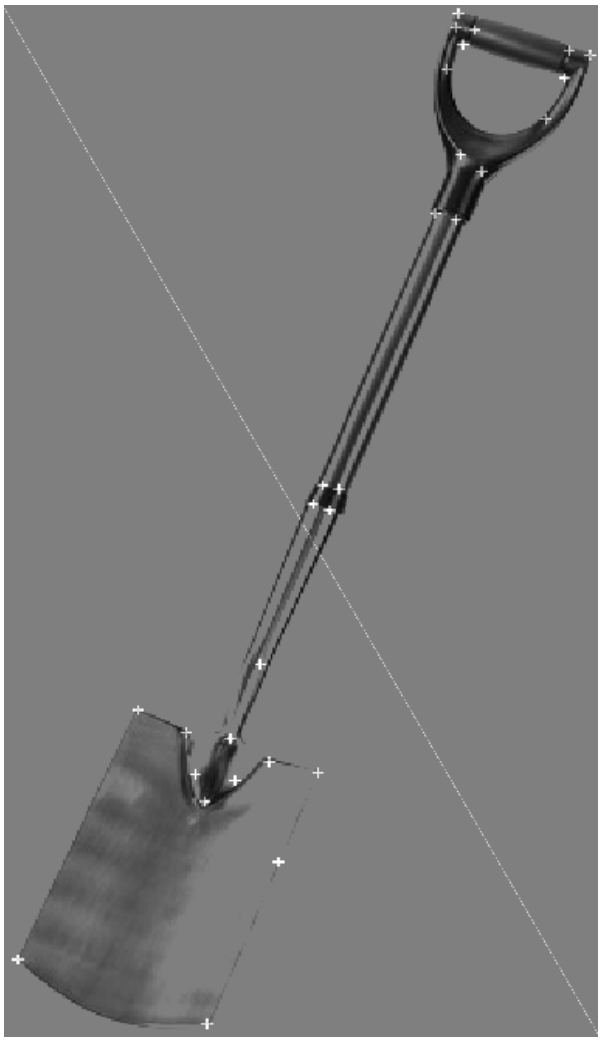


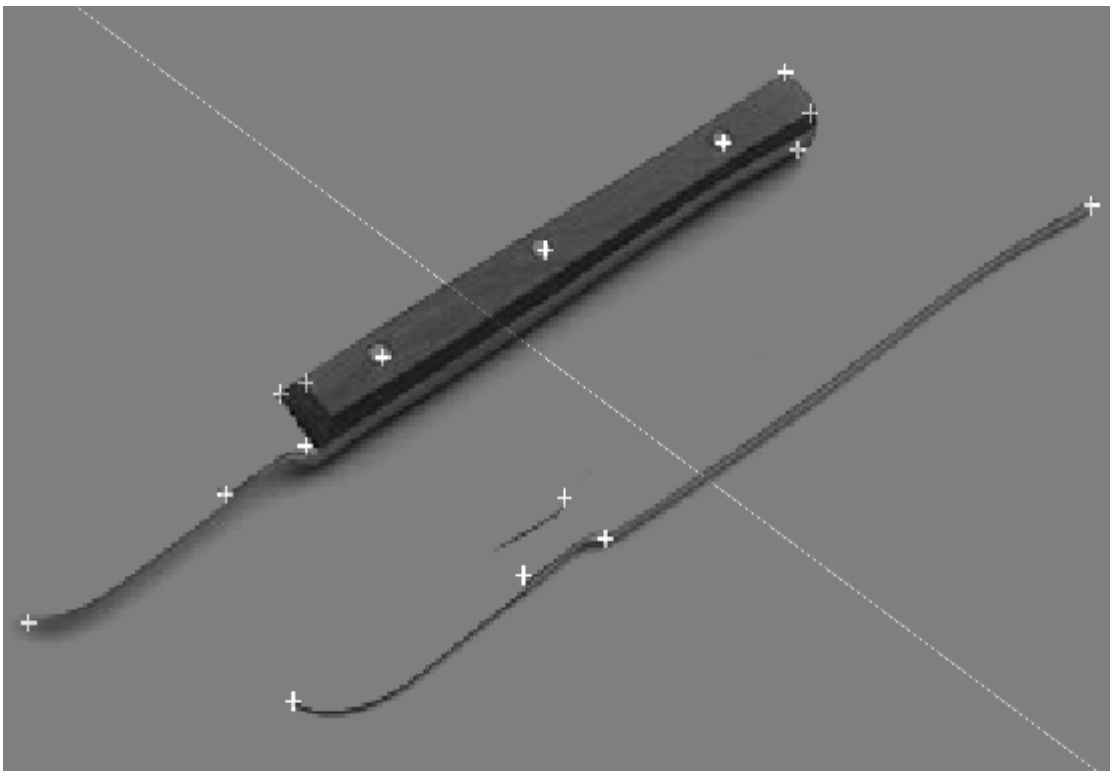












Příloha 4. CD

V kořenovém adresáři přiloženého disku je uložen text bakalářské práce bcprace.pdf. V adresáři Zdroj se nachází zdrojové kódy programu, zdrojový text bakalářské práce a toolkit MDSTk. V adresáři Output se nachází ukázka výstupních obrázků. V adresáři Input jsou obrázky, na kterých probíhalo testování.