

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky

Diplomová práce

Automatické vyhledávání vřícovacích bodů pro polynomiální rektifikaci

Václav VOBORA /A02179/

Plzeň, 2007

Obsah

1	Úvodní část	7
1.1	Vymezení problému	7
1.2	Současný stav řešení	7
1.3	Popis diplomové práce	8
2	Digitální obraz	9
2.1	Definice digitálního obrazu	9
2.2	Rozlišovací schopnost obrazových dat	10
2.3	Typy obrazů	10
2.4	Vzorkování a kvantizace obrazu	10
3	Detekce hran	12
3.1	Hrany	12
3.2	Vybrané metody detekce hran	13
3.2.1	Robertsův operátor	13
3.2.2	Laplaceův operátor	13
4	Detekce významných bodů	15
4.1	Detektor bodů	15
4.2	Moravcův operátor – 1	15
4.3	Moravcův operátor – 2	16
4.4	MMIO operátor	16
4.5	Zuniga–Haralick operátor	16
4.6	Trucco – Verri corner detector	17
4.7	Corner detector	17
4.8	Texture based corner detector	17
4.9	Harrisův operátor	18
4.9.1	Základní princip	18
4.10	Hessian detector	20
5	Matching	21
5.1	Teorie obrazové korelace	21
5.1.1	Technika vyhledání bodů	22
5.2	Template Matching	22
5.2.1	Míry podobnosti	23
5.3	Srovnávání se vzorem pomocí Fourierovy transformace	23
5.4	Různé techniky metody srovnávání se vzorem	23
5.5	Korelace	24
5.5.1	Lineární korelace	24
5.5.2	Cross-correlation	25
5.5.3	SDD míra	25

6	Geometrické transformace	26
6.1	Základní pojmy	26
6.2	Geometrická transformace	26
6.2.1	Homogenní souřadnice	27
6.3	Polynomiální transformace	27
6.3.1	Afinní transformace – polynomiální transformace prvního řádu	28
6.3.2	Polynomiální transformace vyšších řádů	29
6.3.3	Bilineární transformace	29
6.4	Projektivní transformace	29
6.5	Převzorkování	31
6.5.1	Sinc filtr	32
6.6	Hodnocení přesnosti transformace	33
6.6.1	RMS chyba	33
6.6.2	Rezidua a RMS chyby každého bodu	34
6.6.3	Celková RMS chyba	34
6.6.4	Tolerance	35
7	Realizace algoritmu pro hledání korespondencí	36
7.1	Formát worldfile	36
7.2	Výběr vhodných bodů	37
7.3	Nearest-neighbor strategy	39
7.3.1	Použitá okolí bodů	40
7.4	Metody ohodnocení podobnosti bodů	42
7.4.1	Porovnání jasu	42
7.4.2	Porovnání histogramu	43
7.5	Ransac algoritmus	43
7.5.1	Výpočet transformační matice	45
7.6	Vytvoření worldfilu pro převzorkovaný obraz	46
8	Zhodnocení a výsledky algoritmu	47
8.1	Testovací data	47
8.2	Výstupy algoritmu	47
8.3	Celkové zhodnocení	49
9	Závěr	53
A	Uživatelský manuál	56
A.1	Základní informace a popis aplikace	56
A.2	Ovládání aplikace	56
B	Obrazová příloha s dalšími výsledky	61
C	Struktura přiloženého CD	65
D	Evidenční list	66

Prohlášení

Předkládám k posouzení a následné obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně, výhradně s použitím citovaných pramenů.

V Plzni dne:

Podpis:

Poděkování

Na tomto místě bych chtěl především poděkovat vedoucímu diplomové práce Ing. Miloši Železnému, Ph.D. za pomoc a cenné připomínky při její realizaci. Dále děkuji všem, kteří mě podporovali a pomáhali mi při studiu.

Abstrakt

Cílem této diplomové práce je navržení algoritmu pro automatické vyhledávání vlíčovacích bodů pro polynomiální rektifikaci. Součástí teoretické části je shrnutí metod pro detekci významných bodů, metod pro popis okolí a vyhledávání bodů podle okolí. Důraz byl kladen také na popis geometrických transformací. Všechny části navrženého algoritmu jsou podobně popsány a v některých případech také formou pseudokódu. Implementace byla provedena v prostředí Matlab 6.5 (R13). Nakonec byly výsledky zhodnoceny na testovacích datech.

Klíčová slova

obrazová funkce, hrany, významné body, Moravcův operátor, Harrisův operátor, geometrické transformace, afinní transformace, RANSAC, Matlab

Abstract

The main aim of this diploma thesis is to design algorithm for automatic searching corresponding points for polynomial rectification. Include of this work is summarizing methods for detection of the interest points, methods for describing neighbourhood and searching points. Also is written about geometric transformations, it's important for image rectification. All parts of design algorithm are also described with pseudocode. Implementation was made Matlab 6.5 (R13). The results are introduced at the end on test datas.

Keywords

image function, edges, interest points, Moravec operator, Harris operator, geometric transformations, affine transformation, RANSAC, Matlab

Kapitola 1

Úvodní část

1.1 Vymezení problému

Pro práci s geografickými informačními systémy jsou důležitá data dálkového průzkumu Země. Jsou obrazové záznamy pořízené z družic, popř. z leteckých nosičů zobrazující určitou část povrchu Země v jednom časovém okamžiku. Jejich význam s časem stále stoupá. Používají se pro rychlou a efektivní aktualizaci geodatabází nebo jako podklad do GIS.

Abychom mohli s těmito daty efektivně pracovat musí být v určitém souřadnicovém systému. To znamená, že každému obrazovému bodu přísluší souřadnice v daném systému. K tomu potřebujeme transformovat pořízený obrazový záznam ze snímkového souřadnicového systému do námi požadovaného souřadnicového systému. Ten bude např. národní systém ČR S-JTSK, nebo globální systémy WGS-84 nebo UTM. Abychom mohli provést tuto transformaci, musíme znát dostatečný počet korespondujících si bodů v obou snímcích. Těmto bodům se říká *vlíčovací body*. Počet vlíčovacích bodů závisí na použité transformaci. Z těchto bodů jsme schopni vypočítat transformační matici, pomocí které přiřadíme ke každému obrazovému bodu z původního záznamu novou polohu, který již je v konkrétním, námi požadovaném souřadnicovém systému. Nové body však mají odlišnou polohu od původních a obsahují mezi sebou díry, nebo se některé dokonce překrývají. Poté je na řadě další proces, a to je *převzorkování*. Pomocí tohoto procesu se chybějící jasové hodnoty dopočítají na základě některých metod.

A tak máme původní záznam v novém souřadnicovém systému. Střed každého obrazového bodu se již reprezentuje pomocí souřadnic bod na povrchu. Ve skutečnosti každý pixel odpovídá nejčastěji čtvercové ploše o určité velikosti. Tento rektifikovaný obrazový záznam je možné použít jako jednu z mnoha vrstev do GIS. A také můžeme např. určovat souřadnice, měřit vzdálenosti, počítat plochy a objemy.

V praxi se používají systémy, které umožňují rektifikaci. Mezi nejznámější patří komerční ERDAS Imagine, ArcGIS nebo open source GRASS GIS. Ve všech těchto systémech se transformace provádí na základě vlíčovacích bodů, které musí uživatel manuálně v obou snímcích vyhledat a k sobě přiřadit. To ale nebývá vždy jednoduché a je to časově náročné.

Cílem této diplomové práce je, jak již napovídá její název, vyvinout algoritmus, který bude automaticky vyhledávat korespondující body ve vstupních snímcích. Z nich poté získá transformační matici a provede se samotná transformace. Součástí by také měla být implementace tohoto algoritmu v programovém prostředí.

1.2 Současný stav řešení

Hledání korespondujících bodů mezi dvěma překrývajícími obrazy je problém, který se řeší již dlouho na celém světě. Nejčastěji se hledají korespondence mezi dvěma obrazy, získanými např. videokamerou, ke sledování objektů na videozáznamu. Dále podobný problém se používá při kompresi videa MPEG, protože tato komprese nezakódovává jen samotné snímky, ale komprimuje

ve směru časové osy. K tomu hledá změny v následujících snímcích. Bližší informace je možné o tomto problému nalézt v [2]. Tyto aplikace jsou vzhledem k hledání korespondencí na obrazových záznamech z DPZ relativně jednoduché. Přitom jednotlivé snímky jsou až na malé výjimky velmi podobné a jsou jen o málo posunuty. Právě hledání tohoto vektoru posunu se využívá při již zmiňovaném sledování objektů. Dále se hledání korespondencí využívá ve fotogrametrii, při automatickém generování digitálního modelu terénu. Na překrývacích se snímcích se hledají stejné body a z rozdílů snímkových souřadnic a několika konstant se snadno získá vzdálenost mezi nosičem a povrchem. Tyto snímky jsou pořízeny za sebou ve velmi krátké době, za stejných podmínek. Ještě další ulehčení je, že aplikace zná přibližnou polohu korespondujících bodů, kterou odhadne z dráhy nosiče. Za použití obrazové korelace je snadné požadovaný bod nalézt. Tímto se mimo jiné celá úloha velmi ulehčí.

U záznamu z DPZ je situace odlišná. Záznamy jsou pořízeny v různých časech, často od sebe také několik let, za různých podmínek (např. oblačnost, různá hladina šumu) a různými senzory. Situace na povrchu během několika let rychle mění, zejména v obydlených oblastech. Různá je také hodnota rozlišení a existence několika spektrálních pásem. Tím vším se situace komplikuje.

Jak už bylo řečeno v předešlé části, cílem je vytvořit aplikaci, kterou bude možno prakticky využívat při rektifikaci satelitních snímků. Zatím nikde žádná taková, volně dostupná, aplikace neexistuje.

1.3 Popis diplomové práce

Celý text je rozdělen do několika kapitol. Ve druhé a třetí kapitole jsou stručně popsány základy digitálního obrazu a metody pro detekci hran. V další potom několik známých a méně známých metod pro detekci významných bodů. V páté kapitole budou popsány metody, které pomáhají nalézt korespondence a v poslední kapitole teoretické části se pojednává o geometrických transformacích (o polynomiálních prvního až třetího řádu a projektivní).

V následujících kapitolách bude na řadě vytvořený algoritmus. Nejdříve jeho popis a poté zhodnocení výsledků a to včetně obrázků, které jsou při hodnocení názornější. Na závěr vše přehledně shrneme.

Kapitola 2

Digitální obraz

V této části si uvedeme základy týkajících se digitálních obrazů. Uvedeme definice digitálního obrazu a informace o kvantování, vzorkování a rozlišení obrazů.

2.1 Definice digitálního obrazu

Obraz můžeme definovat jako průmět reálného světa na sítnici, fotografii, obrazovku, obraz na papíře. Ten můžeme nahradit matematickým modelem jako spojitou funkci f dvou nebo tří proměnných. Tuto funkci nazveme *obrazovou funkcí*. Statický obraz je popsán obrazovou funkcí dvou proměnných

$$z = f(x, y)$$

v rovině (obecněji na ploše). Obrazová funkce tří proměnných $f(x, y, t)$ se použije pro rovinné obrazy měnící se v čase, nebo jako objemové obrazy $f(x, y, z)$.

Při zpracování obrazu se pracuje s obrazovými funkcemi, které jsou reprezentovány maticemi. Souřadnice a také hodnoty jsou celočíselné. Definičním oborem obrazové funkce $f(x, y)$ je rovinná oblast R

$$R = \{(x, y), 1 \leq x \leq x_m, 1 \leq y \leq y_n\},$$

kde x_m a y_n jsou maximální souřadnice obrazu. Funkce je tedy definována na podmnožině 2-D prostoru a má omezený definiční obor.

Definic obrazu je mnoho a liší se podle jednotlivých oborů. Ta, kterou jsme teď popsali je kombinací uvedenou v [8] a [7]. Definiční obor je vždy diskrétní, a proto se používá rastr, který je složen z obrazových bodů. Rastr je matice o m řádcích a n sloupcích. Jeden bod se také někdy značí jako $f(i, j)$, kde i a j reprezentují konkrétní řádek, resp. sloupec. Hodnoty i nabývají hodnot $1, 2, \dots, m$ a j hodnot $1, 2, \dots, n$. Celkově obraz obsahuje $m \times n$ bodů. Jeden obrazový bod je průnikem jednoho sloupce a řádku. Pro obrazový bod se používá zkratka pixel (angl. pixel element). Dále budeme používat názvy obrazový bod a pixel ve stejném významu. Jejich důležitou vlastností je konečná velikost (je to dále nedělitelná jednotka). Orientace souřadnic je běžná jako kartézském souřadnicovém systému. Tedy osa x vpravo a osa y nahoru, nebo v jako maticovém počtu, pomocí řádků a sloupců.

Hodnotou obrazové funkce je jas (intenzita). Tato hodnota v objemových obrazech není jedna hodnota, ale je to vektor (např. v multispektrálních obrazech nebo na tomografu). Je to měřená veličina záznamových zařízení. Někdy se obrazová funkce reprezentuje jinými veličinami. Např. v GIS je to výška, teplota, tlak, srážky a mnoho dalších, závislých na konkrétní aplikaci.

Dvourozměrná funkce je výsledkem perspektivního zobrazení části 3D prostředí. Tento model odpovídá obrazu v dírkové komoře. Při zjednodušení se nelineární perspektivní zobrazení nahrazuje pravoúhlým (ortografickým) zobrazením. U tohoto se předpokládá, že se ohnisková vzdálenost objektivu se nachází v nekonečnu. Při převodu z 3D do 2D dochází ke ztrátě mnoha cenných informací. To, jak se při dálkovém průzkumu Země získávají data (jaká je geometrie sběru dat), k jaké dochází ztrátě informací, je možné se dočíst v [6].

2.2 Rozlišovací schopnost obrazových dat

Rozlišovací schopnost je základní vlastností každého snímacího zařízení. Je to míra schopnosti rozlišit v signálu dva body, nebo vzorky. Při pořizování tedy můžeme definovat čtyři základní typy rozlišovací schopnosti: radiometrickou, spektrální, prostorovou a časovou.

- *Radiometrická rozlišovací schopnost*
Definuje se silou signálu na detektoru. Určuje počet rozlišitelných úrovní signálu. Podrobněji popsáno v části o kvantování.
- *Spektrální rozlišovací schopnost*
Udává šířku intervalu vlnových délek elektromagnetického spektra. Tato šířka může být různě velká. U panchromatického pásma je to jen jedno široké pásmo, naopak u hyperspektrálních snímků jsou to pásma velmi úzká a je jich několik desítek nebo dokonce stovek. Výběr spektrálních pásem se určuje tak aby bylo možné co nejlépe identifikovat objekty. Landsat TM má 7 pásem, naopak Spot jen 4 pásma a běžný digitální fotoaparát 3 pásma.
- *Prostorová rozlišovací schopnost*
Určuje skutečnou velikost obrazového bodu. Právě skutečná velikost těchto bodů určuje rozlišovací schopnost obrazu (zkráceně rozlišení). Čím se bude zmenšovat skutečná velikost obrazových bodů, tím poroste prostorové rozlišení a bude možné rozpoznávat menší objekty.
- *Časová rozlišovací schopnost*
Udává jak často snímací zařízení je schopné získat data daného objektu nebo území. Snímky ze stejného území z různých časových období poskytují informace o při detekcích změn. Toto se využívá při v DPZ pro různé obory, např. při vývoji a pohybu oblačnosti v meteorologii, změny v zástavby, lesů, vodních toků, a pohybujících se objektů na Zemi.

2.3 Typy obrazů

Tady si teď uvedeme základní definice různých druhů obrazů, jak jsou uvedeny v [2]. Nejdříve šedotónový, poté multispektrální a nakonec tematický obraz.

Šedotónový obraz (angl. gray-scale image) je monochromatický digitální obraz s jednou hodnotou jasu na jeden obrazový bod.

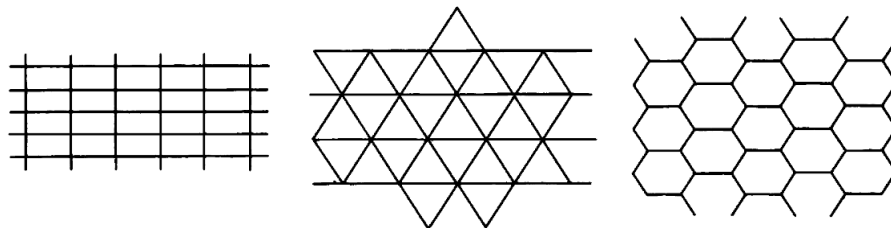
Multispektrální obraz (angl. multispectral image) je 2D obraz, který má vektor hodnot v každém bodě. Pokud je obraz ve skutečnosti barevný, tak má vektor 3 prvků.

Tematický obraz (angl. labeled image) je digitální obraz, ve kterém jsou obrazové body symboly z konečné abecedy. Hodnota symbolu bodu určuje informaci, která označuje skutečnou vlastnost obrazového bodu. Někdy se tento obraz nazývá jako pseudo-barevný obraz.

2.4 Vzorkování a kvantizace obrazu

Hodnoty obrazové funkce musí být nějakým způsobem rozmístěny v prostoru a tím se zabývá vzorkování. Také musí mít nějaký rozsah jasových hodnot a to vysvětluje kvantování.

V souvislosti se vzorkováním spojité funkce je potřeba určit vhodnou vzdálenost mezi vzorkovacími body v obraze. Tímto se zabývá *Shannonova věta*. Ta říká, že vzorkovací frekvence musí být dvakrát větší než nejvyšší frekvence ve vzorkovaném signálu. Interval vzorkování se musí volit tak, aby byl menší nebo rovný polovině rozměru nejmenšího detailu v obraze. Při zpracování obrazů se doporučuje vzorkovat alespoň 5–krát častěji než je teoretická mez daná Shannonovou větou. Dále je nutné vybrat vhodné prostorové uspořádání bodů (výběr vzorkovací mřížky). Ty bývají umístěny v některé pravidelné mřížce. Možné je vzít trojúhelníkovou, čtvercovou nebo hexagonální



Obrázek 2.1: Obdélníková, trojúhelníková a hexagonální mřížka (převzato z [4])

mřížku, viz. obrázek (2.1). Nejčastěji se používá čtvercová, a to přes problémy se vzdálenostmi a souvislostmi oblastí. Naopak umožňuje obraz uložit v počítači jako datový typ pole. Na druhou stranu hexagonální má jednu velkou výhodu. Všechny sousední body jsou ve stejné vzdálenosti.

Ve zpracování digitálního obrazu je nutné převádět hodnoty obrazové funkce do digitální formy (digitalizovat) s ohledem na jejich souřadnice (polohu). Proces kvantizace nahrazuje spojitý signál funkce $f(x, y)$ diskrétní množinou kvantizačních hladin (angl. levels). Přesnost je určena počtem kvantizačních hladin, které použijeme. Platí následující tvrzení: čím více kvantizačních hladin, tím lepší přesnost. Množina n kvantizačních úrovní se skládá z celých čísel $0, 1, 2, \dots, n - 1$. 0 až $n - 1$ jsou obvykle zobrazeny na výstupním zařízení nebo vytištěny jako černá a bílá barva, resp. jako jednotlivé odstíny šedé barvy. Úroveň kvantizace se běžně nazývá jako *stupně šedi*. Odtud máme název šedotónový obraz.

Protože se digitální obrazy zpracovávají pomocí počítače, používá se počet stupňů šedi n , jako n -tá mocnina dvou. Pak můžeme napsat

$$n = 2^b,$$

kde b je počet bitů použitých ke kvantizaci. Běžně se v DPZ používají obrazy s 256 úrovněmi jasu (8-bitů). Speciální zařízení používané v medicíně nebo v astronomii používají obrazy kvantizované 10 nebo 12 bity.

Kapitola 3

Detekce hran

V této kapitole získáme představu o tom, jak můžeme v obraze detekovat hran. Nejprve si řekneme co to hrana vlastně je. Hrana je místo v obraze, kde se prudce mění hodnota jasu (tedy jasové funkce). Při vnímání jsou právě tato místa důležitá a nesou nejvíce informací. V počítačovém vidění je cílem tyto místa zvýraznit (operací ostření - zvýrazňují se vysoké frekvence) nebo detekovat (pomocí hranových operátorů). Hrany jsou invariantní vůči osvětlení a místě pohledu. Detekce hran má mnoho použití, např. rozpoznávání obsahu obrazu, sledování objektů, 3-D rekonstrukce snímané scény a hledání korespondencí. Toto se právě týká této práce. Detekce hran obecně patří mezi lokální předzpracování.

3.1 Hrany

Hrana je dána vlastnostmi obrazového bodu a také vlastnostmi okolí. Jak už jsme se zmínili je určena tam, kde se náhle mění hodnota obrazové funkce $f(x, y)$. Matematickým nástrojem pro, jak můžeme získávat tyto informace jsou derivace. Velikost změny funkce určuje gradient $\nabla f(x, y)$ a směr gradientu dává informaci o směru největšího růstu obrazové funkce. Body s velkým gradientem se nazývají hrany. Někdy se pro bod, který reprezentuje hranu užívá výraz *edgel* (edge element), analogicky jako pixel.

V ideálním případě pro spojitou funkci dvou proměnných $f(x, y)$ platí pro gradient

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right),$$

velikost gradientu

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2},$$

a směr gradientu

$$\varphi = \arctan \frac{\left(\frac{\partial f}{\partial y} \right)}{\left(\frac{\partial f}{\partial x} \right)}.$$

Velikost gradientu je nezávislá na natočení souřadnicové soustavy. Pro jednorozměrný signál je gradient roven derivaci funkce jedné proměnné.

Derivace $f(x, y)$ ve jednom směru (u, v) se získá takto

$$(u, v) \cdot \nabla f(x, y) = \left(u \frac{\partial f}{\partial x}, v \frac{\partial f}{\partial y} \right).$$

Pro detekci hran se používají se hranové operátory. Nalezené hrany se používají pro hledání hranic jednotlivých objektů.

Místo přímého výpočtu derivace, aproximovaného pomocí diferencí, se používají konvoluční masky. Jsou to gradientní operátory, které detekují hrany a k tomu využívají konvoluci. Použití operátoru konvoluce je z hlediska doby výpočtu je výhodnější, protože v různých programových balících je tato operace efektivněji implementována. Výpočet je rychlejší než implementace pomocí diferencí jednotlivých elementů v matici.

Gradientní operátory lze rozdělit do tří skupin

- operátory aproximující pomocí diferencí, některé se počítají jen jednou maskou, další pomocí několika a pak se vybere ta, které nejlépe odpovídá derivace obrazové funkce, příkladem je Robertsův operátor
- operátory, které hledají hrany v místech, kde druhá derivace prochází nulou, příkladem je tzv. zero-crossing a Laplaceův operátor
- operátory, které lokálně aproximují obrazové funkce parametrickým modelem, např. polynommem dvou proměnných Haralickova operátoru, vztah (4.2), na podobném principu pracuje jeden z detektorů významných bodů

V dalších částech jsou uvedeny příklady některých gradientních operátorů. Podrobnější informace o detekci hran lze nalézt v [1], [2] a [4].

3.2 Vybrané metody detekce hran

3.2.1 Robertsův operátor

Nejjednodušší je Robertsův operátor. Jako jediný používá jen okolí 2×2 pixelu. Velikost gradientu vypočítáme podle vztahu

$$f = |f(i, j) - f(i + 1, j + 1)| + |f(i, j + 1) - f(i + 1, j)|. \quad (3.1)$$

Operátor se skládá ze dvou konvolučních masek

$$h_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad h_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \quad (3.2)$$

Obě tyto masky jsou na obraz aplikovány odděleně a výstupem různá míra gradientu ve dvou orientacích (h_x a h_y). Výstupy se potom kombinují, tak aby se našla absolutní velikost gradientu v každém bodě a orientace. Velikost gradientu je dána

$$|H| = \sqrt{h_x^2 + h_y^2}$$

a nebo jen aproximace (pro výpočet je rychlejší)

$$|H| = |h_x| + |h_y|.$$

Výše uvedené masky jsou navrženy tak, aby dávaly velkou odezvu na hrany, které jsou otočeny o 45 stupňů k rastrové mřížce. Tento operátor je citlivý na šum, protože použité okolí je velmi malé, ale doba výpočtu je oproti ostatním kratší.

3.2.2 Laplaceův operátor

Laplaceův gradientní operátor je také hodně používán. Aproximuje druhou derivaci, je invariantní vůči otočení a určuje velikost hrany (ale ne směr). Používají se konvoluční jádra pro 4-sousedství a pro 8-sousedství.

Laplacián funkce $f(x, y)$ je definován takto:

$$\Delta f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}.$$

Konvoluční jádro získáme tak, že sečteme dvě jádra, které se používají aproximaci druhé diference. Tyto musíme jen mírně upravit pro dvourozměrnou funkci. Budou mít velikost 3×3 a tvar

$$h_{xx} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad h_{yy} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \quad (3.3)$$

První je pro detekci druhé derivace ve směru osy x a druhá ve směru osy y (je to jen transponovaná matice odpovídající pro x). Výsledné konvoluční jádro získáme sečtením dvou předchozích jader.

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (3.4)$$

Někdy se také používá maska, která má větší váhu pro centrální pixel. Ale v tomto případě pak operátor ztrácí invariantnost vůči otočení.

$$h = \begin{pmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{pmatrix} \quad h = \begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \quad (3.5)$$

Nevýhoda jako u předchozího operátoru je citlivost na šum. Další nevýhodou jsou detekované dvě odezvy (dva pixely) reprezentující tenkou linii v obraze.

Kapitola 4

Detekce významných bodů

Abychom mohli vybírat z obrazu korespondující body, musíme nejdříve získat nějakou množinu bodů, ze které je budeme potom hledat. Není možné hledat samotné, odpovídající pixely, to by bylo neúnosné. K tomu budeme potřebovat nějaké metody pro detekci významných bodů. Metod pro tuto detekci je mnoho, ale nejvíce používaný v praxi je Harrisův operátor. Ten si popíšeme nejpodrobněji. Uvedeme si jednodušší metody jako je Moravcův operátor, poté ty složitější, jako metody založené na výpočtu Hessovy matice.

Nejdříve si musíme zadefinovat, co to vlastně roh (angl. corner) je. Je to obrazový bod ve kterém se sbíhají dvě hrany dohromady. V tomto bodě má obrazové funkce velký gradient ve směrech x a y . Výhodnější je hledat rohy právě pomocí gradientu, než z obrazu hran.

Operátory, které se nazývají Moravcovy, jsou v literatuře popsány v různých modifikacích. Každý z nich počítá odezvu jinak. Rozdíly mezi nimi jsou hlavně v počtu operací, které jsou potřeba pro výpočet odezvy z okolí. My si zde popíšeme jen některé z nich.

4.1 Detektor bodů

Úplně nejjednodušší metodou je aplikace na obraz konvoluční jádro velikosti 3×3 (4.1). Nami požadované body získáme prahováním. Výhodou rychlá implementace a malé výpočetní nároky.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (4.1)$$

4.2 Moravcův operátor – 1

Moravcův operátor je nejjednodušší možností, jak detekovat body. Tento operátor poprvé popsal Hans P. Moravec v roce 1977. Cílem je určení bodů zájmových bodů (angl. *points of interest*). Původní účel byl určit body pro nalezení odpovídajících oblastí ve sekvencích snímků.

Moravcův operátor detekuje významné body, jako body, kde se je velká změna rozptylů jasů ve všech směrech. Těmto bodům se říká rohy. Svým způsobem není tento operátor určen pro nalezení rohů, ale jen určitých oblastí v obraze po sobě jdoucích sekvencích obrazů.

Jak můžeme rozptyl (nebo varianci) určitých pozicích v obraze? H. Moravec navrhl měření rozptylů intenzity jasů položení malého, nejčastěji čtvercového tvaru. Mají většinou velikost 3×3 , 5×5 a 7×7 . Jsou centrovány v bodě (x_0, y_0) . Toto okno se posouvá o jeden pixel v osmi základních směrech, tzn. dva v horizontálním, dva ve vertikálních a ve čtyřech diagonálních. Rozptyl pro daný posun se vypočítá jako suma čtverců rozdílů odpovídajících si pixelů ve dvou oknech. Jedno je umístěné v centru v bodě (x_0, y_0) a druhého, které je posunutí o jeden pixel v jednom určitém směru.

Rozptyl intenzit v centrálním bodě je dán jako *minimum* z osmi základních směrů. Detekované body se vyberou prahováním.

4.3 Moravcův operátor – 2

Dalším ze základních detektorů druhá varianta Moravcova operátoru.

$$M(i, j) = \frac{1}{8} \sum_{u=i+1}^{i-i} \sum_{v=j+1}^{j-i} |f(i, j) - f(u, v)|$$

Tento operátor je velmi jednoduchý. Počítá jen rozdíly hodnot jasů mezi centrálním pixelem v bodě i a j ve nejbližším okolí (8-okolí). Absolutní hodnoty těchto rozdílů se sečtou a výsledná odezva se vydělí osmi. Tady se potřebuje osm odčítání a osm sčítání (devět, pokud se počítáme také centrální pixel, u toho bude rozdíl nula).

Na výsledný obraz se aplikuje vhodný práh, a poté non-maxima suppression¹. Metoda vybírá pixely s velkým kontrastem, body, které se hodně liší od ostatních. Výhodou je snadná implementace a nízká výpočetní náročnost. Operátor je velmi vhodný pro real-time aplikace. Ale ve skutečnosti se v praxi používají lepší detektory.

4.4 MMIO operátor

Tento operátor je založen na Moravcově operátoru, je to jeho modifikace (MMIO – Moravec modified interest operator). Je upraven pouze ve výpočtu funkce odezvy. Algoritmus je podrobně popsán v [12].

Vypočítají se rozptyly ve čtyřech základních směrech, v horizontálním, vertikálním a diagonálním a anti-diagonálním směru. Použije se velikost okna 11×11 . Z těchto čtyřech vypočítaných hodnot se vybere ta minimální. Takže např. pro rozptyl v horizontálním směru se použije se vztah

$$var(i, j) = \sum_{k=i-5}^{k=i+5} (f(k, j) - f(i, j))^2,$$

kde $f(i, j)$ aktivní (centrální) obrazový bod. Minimální hodnota rozptylu se použije jako míra odezvy. V [12] je nazván jako *variance image*. Poté přijde na řadu prahování, pokud je odezva menší než předem daný práh tak do výstupního obrazu se uloží nula, v opačném případě minimální rozptyl. V tomto obraze se významné body vyberou tam, kde odezva nabývá lokálního minima. Vybrané body jsou seřazeny podle velikosti odezvy. Autor uvádí ideální hodnotu prahu pro homogenní oblasti na 10000.

4.5 Zuniga–Haralick operátor

U tohoto operátoru je obrazová funkce f aproximována v okolí obrazového bodu $f(x, y)$ polynomem třetího stupně

$$f(x, y) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^3 + c_7x^3 + c_8x_2y + c_9xy^2 + c_{10}y^3 \quad (4.2)$$

Samotný operátor je dán vztahem

$$zh(x, y) = \frac{-2(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4)}{\sqrt[3]{(c_2^2 + c_3^2)^2}}.$$

Mírnou změnou tohoto operátoru je *Kitchen-Rosenfeld operator*. Ten také aproximuje obrazovou funkci polynomem třetího stupně podle vztahu (4.2). Ale odezva se vypočítá jinak, a to

$$kr(x, y) = \frac{-2(c_2^2c_6 - c_2c_3c_5 - c_3^2c_4)}{(c_2^2 + c_3^2)}.$$

Významné body se získají prahováním obrazu odezev. Prah se musí určit experimentálně (závisí na konkrétním obraze). Výpočty odezev těchto operátorů jsou výpočetně náročné. Oba operátory jsou uvedeny v [1].

¹V obraze se vybírají pouze lokální maxima obrazové funkce.

4.6 Trucco – Verri corner detector

Vypočítáme derivace obrazové funkce $f(x, y)$ v obou směrech x a y .

$$f_x = \frac{\partial f(x, y)}{\partial x} \quad f_y = \frac{\partial f(x, y)}{\partial y}$$

Určíme 3×3 okolí pro každý bod a pro každý obrazový bod spočítáme matici

$$C(i, j) = \begin{pmatrix} \sum f_x^2 & \sum f_x f_y \\ \sum f_y f_x & \sum f_y^2 \end{pmatrix}.$$

Pro každou matici $C(i, j)$ určíme vlastní čísla $\lambda_{i,j} = [\lambda_1, \lambda_2]$. Vytvoříme obraz, ve kterém bude $l(i, j) = \min [\lambda_1, \lambda_2]$. Provedeme prahování a všechno větší než práh je považováno za roh. Výsledné body jsou závislé na prahu, tzn. jsou funkcí prahu. Tento detektor je popsán v [16].

4.7 Corner detector

Tento operátor je popsán v [2]. Princip je velmi jednoduchý. Vypočítají se čtyři variance, dvě ve horizontálním a vertikálním směru a dvě v diagonálních směrech přes čtvercové okolí. Pokud si označíme w jako poloviční velikost okna, tak může počítat rozptyly ve všech směrech:

- v horizontálním směru $f(i, j - w), \dots, f(i, j + w)$
- ve vertikálním směru $f(i - w, j), \dots, f(i + w, j)$
- v diagonálním směru $f(i - w, j - w), \dots, f(i + w, j + w)$
- v diagonálním směru $f(i - w, j + w), \dots, f(i + w, j - w)$

Z těchto čtyřech hodnot vybereme minimum. Tento postup aplikujeme na všechny obrazové body (samozřejmě kromě krajů, nemají sousedy) a pokud je vybraná hodnota větší než předem určený práh, tak bod označíme jako roh. Důležitý je také výběr velikosti okna. Jako nejvýhodnější z hlediska detekce bodů a výpočetní náročnosti se jeví hodnoty 5, 7 a 9. Počet bodů zapojených do počítání variancí se rychle zvyšuje (u $w = 3$ se počítá celkem z 12 bodů, u $w = 5$ 20 a u $w = 7$ 28).

4.8 Texture based corner detector

Tento operátor využívá k detekci bodů texturu. Je to alternativa výše uvedenému operátoru a také je popsán v [2]. Nejdříve musíme spočítat velikost gradientu obrazu a to podle vztahu

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}.$$

K tomu použijeme masku buď 3×3 nebo 2×2 (symetrickou nebo nesymetrickou diferenci). Poté použijeme prahování, abychom získali binární obraz. Obrazový bod $f(x, y)$ bude pro nás zajímavý pouze v případě, že bude mít významný rozptyl ve čtyřech hlavních směrech $n \times n$ okolí (v horizontálním, vertikálním směru a v obou diagonálních) v binárním obraze. Získaný binární obraz si označíme jako B . Rozptyl ve směru $(\Delta x, \Delta y) = [(0, 1), (1, 0), (1, 1), (1, -1)]$ je součet bodů

$$B(x, y) \otimes B(x + \Delta x, y + \Delta y)$$

pro všechny body v $n \times n$ okolí se středem v $B(x, y)$, kde symbol \otimes znamená *exclusive or*. Tento operátor vrací číslo jedna, pokud jsou dva vstupy různé. Výstupní obraz získáme přiřazením minima ze čtyř rozptylů.

4.9 Harrisův operátor

Tento detektor je v praxi nejvíce používaný. Poprvé byl publikován v roce 1988 Chrisem Harrisem a Mikem Stephensem. V [11] je autory popsán. Hlavním účelem tohoto operátoru bylo vyhledání bodů v obraze hran a poté hledání korespondujících bodů v sekvenci snímků, získaných jednou kamerou. V článku je napsáno, že pro svůj účel má nejlepší výsledky a odstraňuje některé nedostatky jiných metod. V [17] je uveden docela pěkný popis Harrisova operátoru.

4.9.1 Základní princip

Máme obrazovou funkci $f(x, y)$, která nám reprezentuje obraz, ve kterém chceme hledat významné body. Potřebujeme vyjádřit minimum rozdílů čtverců mezi obrazovým bodem a posunutým okolím přes všechny směry $(\Delta x, \Delta y)$. Těch směrů je osm. Vypočítanou odezvu operátoru označíme E .

$$E(x, y) = \min \sum_{(x, y) \in N_0} \left(f(x, y) - f(x + \Delta x, y + \Delta y) \right)^2, \quad (4.3)$$

kde N_0 je okolí pixelu (x_0, y_0) . Teď budeme aproximovat $f(x + \Delta x, y + \Delta y)$ Taylorovým rozvojem. Ale pouze prvními dvěma členy. Získáme

$$f(x + \Delta x, y + \Delta y) \doteq f(x, y) + \nabla f(x, y)^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \quad (4.4)$$

Pokud aproximaci (4.4) dosadíme do (4.3) získáme

$$E(x, y) = \min \sum_{(x, y) \in N_0} [\Delta x, \Delta y] \nabla f(x, y) \nabla f(x, y)^T \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

a po úpravách

$$\begin{aligned} \min [\Delta x, \Delta y] \left(\sum_{(x, y) \in N_0} \nabla f(x, y) \nabla f(x, y)^T \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ \min \sum_{(x, y) \in N_0} [\Delta x, \Delta y] \mathbf{A} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \end{aligned}$$

Matice \mathbf{A} je symetrická, pozitivně semi-definitní o velikosti 2×2 .

$$\mathbf{A} = \sum_{(x, y) \in N_0} \nabla f(x, y) \nabla f(x, y)^T \quad (4.5)$$

$$\mathbf{A} = \sum_{(x, y) \in N_0} \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} & \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \quad (4.6)$$

Když \mathbf{A} je symetrická a to je, pak platí

$$\min [\Delta x, \Delta y] \mathbf{A} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \|(\Delta x, \Delta y)\|^2 \lambda_{min}, \quad (4.7)$$

kde λ_{min} je minimální vlastní číslo matice \mathbf{A} .

Matice \mathbf{A} má v různých případech následující vlastnosti:

- pokud je jasová funkce blízká konstantní funkci v okolí N_0 , pak gradient $\nabla f(x, y)$ a matice \mathbf{A} jsou blízké k nulovému vektoru, resp. k nulové matici a obě hodnoty vlastních čísel λ_1 a λ_2 se blíží k nule
- pokud je gradient $\nabla f(x, y)$ stejný v okolí N_0 , pak matice \mathbf{A} je singulární a $\lambda_{min} = 0$

- pokud je gradient různý ve všech směrech v okolí N_0 , matice \mathbf{A} má hodnotu dvě a $\lambda_{min} > 0$

Jak bylo napsáno v předchozí kapitole o hranách výpočet gradientu je velmi citlivý na šum. Proto je předtím nutné vždy obraz rozmazat, abychom eliminovali šum. Použijeme nějaký low-pass filtr. Typickým příkladem je provedení konvoluce s Gaussiánem. Pro připomenutí

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

kde σ je směrodatná odchylka, která kromě velikosti masky také určuje míru vyhlazení obrazové funkce.

Touto operací eliminujeme určitou část šumu, tzn. odstraníme vysoké frekvence. Gradient $\nabla f(x, y)$ můžeme získat např. tímto způsobem, použitím symetrických diferencí.

$$\nabla f(x, y) = \begin{bmatrix} f(x+1, y) - f(x-1, y) \\ f(x, y+1) - f(x, y-1) \end{bmatrix} \quad (4.8)$$

Obrazová funkce $f(x, y)$ je již po aplikaci vyhlazení. Silné vyhlazení sníží velikost gradientu. Pokud je aplikované okolí N_0 malé, zvyšuje se pravděpodobnost, že matice bude singulární a λ_{min} se blíží k nule. Velké okolí zase naopak zvýší pravděpodobnost, že ve všech směrech bude významná diference a matice bude mít plnou hodnotu. Teď využijeme Gaussián a doplníme s ním rovnici (4.5). Získáme

$$\mathbf{A} = \sum_{(x,y) \in N_0} G(x-x_0, y-y_0) \nabla f(x, y) \nabla f(x, y)^T \quad (4.9)$$

Velikost okolí N_0 a směrodatná odchylka σ jsou parametry, které ovlivňují vlastnosti a úspěšnost Harrisova operátoru. Angl. se nazývají jako *integration scale*.

Po výpočtu matice \mathbf{A} se přejde k Harrisově funkci (je to odezva po aplikaci Harrisova operátoru). Anglicky se nazývá *response function*. Harrisova odezva má následující tvar

$$R = \det(\mathbf{A}) - \kappa \operatorname{tr}^2(\mathbf{A}),$$

kde $\operatorname{tr}(\mathbf{A})$ je stopa² matice a κ je konstanta určená experimentálně. Jako nejvhodnější hodnota je 0,04 (získaná experimentálně). Pokud budeme chtít vyjádřit determinant a stopu matice v jednotlivých prvcích, tak bude platit

$$\det(\mathbf{A}) = \lambda_1 \lambda_2 = \mathbf{A}(1, 1)\mathbf{A}(2, 2) - \mathbf{A}(1, 2)^2$$

a

$$\operatorname{tr}(\mathbf{A}) = \lambda_1 + \lambda_2 = \mathbf{A}(1, 1) + \mathbf{A}(2, 2).$$

A pokud toto dáme dohromady, tak se funkce odezvy vypočítá následujícím způsobem

$$R(x, y) = \left(\mathbf{A}(1, 1)\mathbf{A}(2, 2) - \mathbf{A}(1, 2)^2 \right) - \kappa \left(\mathbf{A}(1, 1) + \mathbf{A}(2, 2) \right)^2.$$

Dále je možné použít také jiné odezvy. V [17] jsou uvedeny vhodné alternativy

$$R(x, y) = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$R(x, y) = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}.$$

Významné body se vyberou jako lokální maxima funkce $R(x, y)$. Využije se tzv. *non-maximum suppression* algoritmus.

²Stopa matice je součet diagonálních prvků

4.10 Hessian detector

Hessian detector (detektor významných bodů na základě Hessovy matice ³) je ve svém principu velmi podobný výše popsanému Harrisovu operátoru. Využívá matici druhých derivací \mathbf{H} , tzv. Hessovu matici.

$$\mathbf{H} = \begin{bmatrix} f_{xx}(x, y, \sigma) & f_{xy}(x, y, \sigma) \\ f_{xy}(x, y, \sigma) & f_{yy}(x, y, \sigma) \end{bmatrix} \quad (4.10)$$

Tuto matici musíme spočítat pro každý obrazový bod. Pro první derivace podle x a y využijeme konvoluční masky, tak jak jsou popsány v předchozí kapitole. Druhé derivace vypočítáme tak, že nepoužijeme stejné masky jako u prvních derivací, ale vezmeme Gaussián s parametrem σ . Pokud již máme pro každý obrazový bod Hessovu matici \mathbf{H} , tak musíme získat determinant této matice.

$$\det(\mathbf{H}) = \sigma^4(f_{xx}f_{yy} - f_{xy}^2) \quad (4.11)$$

To znamená, že pro každý obrazový bod vypočítáme Hessovu matici. Významné body vybereme, tak že vezmeme body, pro které má odezva větší hodnotu, než předem nastavený vhodný práh. Hodnota σ určuje, kolik bodů bude vybráno. Čím bude hodnota σ větší, tak bude vybráno více bodů. Slouží k omezení počtu vybraných bodů a závisí na konkrétních datech.

³Hessova matice je symetrická, má na diagonále druhé derivace podle jednotlivých proměnných a na mimo-diagonálních prvcích smíšené derivace

Kapitola 5

Matching

V této kapitole si popíšeme několik a možností, jak získat z dvou obrazů podobné části. Metody budou jak z počítačového vidění, dálkového průzkumu Země a také z fotogrammetrie. Dále si podrobně popíšeme metodu template matching a korelaci, pro výpočet míry shody. Uvedeme také možnost, jak použít Fourierovu transformaci u metody srovnávání se vzorem.

5.1 Teorie obrazové korelace

Cílem obrazové korelace je nalézt polohu určovaného bodu na obou snímcích stereoskopické dvojice. Tato metoda našla uplatnění při automatickém zpracování záznamů z fotogrammetrie (např. při tvorbě digitálního modelu terénu). Hledají se snímkové souřadnice bodu z obou snímků.

Pokud bychom hledali v obraze bod, který je reprezentovaný jedním pixelem, v rámci jednoho pixelu v druhém obraze, pak bychom našli mnoho stejných bodů, jejichž počet by sahal podle velikosti snímku až k hodnotám, které dosahují řádově desetitisíců až statisíců. Proto se využívá předpokladu, že každý bod má do kolem sebe do jisté míry unikátní okolí. Za pomoci tohoto okolí můžeme určit polohu hledaného bodu v druhém obraze. Platí zde přímá úměra. Čím bude okolí bodu větší, tím bude okolí unikátnější (pravděpodobnost výskytu stejného okolí bude menší), ale také se přirozeně zvýší výpočetní náročnost celého prohledávání. Ve většině případů bude toto prohledávání dávat přijatelné výsledky, ale v některých oblastech může selhat, jako např. ve velkých homogenních plochách (pole, les, vodní hladina). Princip obrazové korelace používá porovnání dvou obrazů mezi sebou a nalezení páru, který si nejvíce podobný. K určení míry podobnosti se musí zadefinovat nějaké míry (geometrické a statistické).

Obrazová korelace využívá koeficient výběrové korelace tak, jak je znám ze statistiky

$$\rho(f, g) = \frac{\text{cov}(f, g)}{\rho(f)\rho(g)}, \quad (5.1)$$

kde f a g jsou obrazy, $\text{cov}(f, g)$ je kovariance a $\rho(f)$, $\rho(g)$ jsou směrodatné odchylky. K výpočtu korelace pro dva stejně velké obrazy se budou používat hodnoty jednotlivých pixelů $f_A(i, j)$ a $f_B(i, j)$. Pro korelační koeficient získáme výraz

$$r(f, g) = \frac{C(f, g)}{\sqrt{C(f)C(g)}}, \quad (5.2)$$

kde jednotlivé výrazy jsou následující:

$$C(f, g) = \frac{1}{n^2 - 1} \sum_{i=1}^n \sum_{j=1}^n (p(f, g) - p(f))(p(f, g) - p(g)), \quad (5.3)$$

$$C(f) = \frac{1}{n^2 - 1} \sum_{i=1}^n \sum_{j=1}^n (p(f) - p(f))^2,$$

$$C(g) = \frac{1}{n^2 - 1} \sum_{i=1}^n \sum_{j=1}^n (p(g) - p(g))^2,$$

$$p(f) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n p(f),$$

a

$$p(g) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n p(g).$$

Tento vztah pro koeficient je uveden v [9].

Ve fotogrammetrii se pro snazší určování používá rámové značky snímku, réseau křížky nebo signalizované vlíčovací body v terénu.

5.1.1 Technika vyhledání bodů

Pro nalezení bodu ve snímku existují v zásadě dva způsoby.

- máme informaci jak objekt (křížek, rámová značka, signalizovaný bod) vypadá a vytvoříme vzor pro prohledávání
- máme obecný bod na jednom snímku a hledáme odpovídající bod ve druhém

Pro nalezení musíme zvolit dostatečně velké okolí. Ve známé nebo odhadnuté poloze určíme vyhledávací oblast. Vypočítáme vzájemnou korelaci (korelační koeficient) a zaznamenáme polohu středu vyhledávacího okénka. Dále posuneme okénko o jeden pixel dále a vypočítáme koeficient a zaznamenáme opět polohu. Takto prohledáme celou vyhledávací oblast. Pokud má vyhledávací okolí velikost $n \times n$ a vyhledávací oblast $m \times m$, tak získáme celkem $(m - n + 1)^2$ korelačních koeficientů. Poloha, která má maximální korelační koeficient, označíme jako střed hledaného bodu. Nalezená poloha se může ještě zpřesnit přechodem na polohu interpolovaného maxima na základě okolních hodnot. Maximum korelace v některých homogenních částech obrazu nenastává v jednom bodě, ale tvoří oblast s plochým maximumem. Námi hledanou nejpravděpodobnější polohu středu určíme interpolací jako těžiště oblasti.

5.2 Template Matching

Template matching (srovnávání se vzorem) je jednou ze základních technik segmentace obrazu. Používá se k lokalizování objektů v obraze na základě určitého vzoru. Jinými slovy máme obraz nějakého vzoru a ten chceme nalézt v původním obraze. Hledaný vzor na rozdíl od celého obrazu velmi malý. Se vzorem posouváme po obraze a hledáme místa, kde je nejlepší shoda s původním obrazem. Tato metoda segmentace je využitelná např. v dynamických obrazech (pohybující se auta, mraky). Během srovnávání se vzorem se porovnávají vlastnosti šedotónového obrazu v malých oblastech. V některých případech může být template matching invariantní k transformacím (jak jasovým, tak geometrickým).

Základní algoritmus srovnávání se vzorem je následující, jak je uvedeno v [1].

- vypočítáme míru podobnosti pro každou polohu a rotaci vzoru v obraze
- pokud lokální maximum překračuje zadaný práh, tak nalezená poloha reprezentuje hledaný vzor

5.2.1 Míry podobnosti

V této části si uvedeme několik možností, jak určit ze vzoru a části obrazu nějakou hodnotu, která reprezentuje, jak si jsou obě části podobné. Jako f si označíme vstupní obraz, h jako hledaný vzor a V jako množinu všech obrazových bodů ve zpracovávaném obraze. Následující vztahy dávají dobré kritérium pro popis vzoru h a části (oblasti) obrazu f v poloze (i, j) .

$$C_1(i, j) = \frac{1}{\max_{(i,j) \in V} |f(i+u, j+v) - h(u, v)| + 1} \quad (5.4)$$

$$C_2(i, j) = \frac{1}{\sum_{(i,j) \in V} |f(i+u, j+v) - h(u, v)| + 1} \quad (5.5)$$

$$C_3(i, j) = \frac{1}{\sum_{(i,j) \in V} (f(i+u, j+v) - h(u, v))^2 + 1} \quad (5.6)$$

5.3 Srovnávání se vzorem pomocí Fourierovy transformace

U srovnávání se vzorem je možné s výhodou využít Fourierovy transformace. Pro výpočet vzájemné korelace obrazu f a vzoru h můžeme počítat součin Fourierových obrazů F a H a poté provést zpětnou transformaci výsledku. Takto získané výsledky považují zpracovávaný obraz za periodický, a proto se budou uvažovat polohy vzoru částečně mimo obraz. Abychom mohli oba Fourierovy obrazy násobit, musí být nutně F a H stejné velikosti. Je-li vzor menší než obraz (a to pravděpodobně bude), shodné velikosti dosáhneme doplněním vzoru řádky a sloupce s nulou až do velikosti obrazu. Místo nul je také možné ukládat např. střední hodnotu jasu zpracovávaných obrazů.

Výpočet korelace vzoru s původním obrazem musí být vzor ještě otočen 90 stupňů. Provedeme násobení obrazů

$$C = F * H \quad (5.7)$$

a obraz C převedeme zpět do pomoci inverzní Fourierovy transformací. V tomto převedeném obraze získáme maximální hodnotu a podle ní určíme vhodnou hodnotu prahu. Práh by měl být někde kolem devadesáti procent.

5.4 Různé techniky metody srovnávání se vzorem

Metodou srovnávání se vzorem jsou získána všechna místa, ve kterých se nacházejí přesné kopie vzoru. Tyto kopie se musí shodovat ve velikosti, tak také v natočení. Geometrické zkruslení musí být jen velmi malé. Abychom mohli využít tuto metodu pro hledání natočených a zvětšených vzorů, museli bychom vytvořit vzor pro každou možnou velikost a každou orientací. Další možnost by bylo použít jen jeden vzor, ale ten srovnávat obraz se všemi možnými geometrickými transformacemi. Mezi těmito dvěma přístupy není velký rozdíl.

Metody srovnávání lze využít také v případě, pokud je dovoleno nekonečné množství transformací vzoru v obraze. Vzor může být složen z několika částí, které jsou navzájem spojeny. Celkový souhlas nemusí být dostatečný, ale vhodné souhlasy mohou být vhodné pro jednotlivé části. Tato nalezená umístění částí vzoru nemusí být ve správných polohách, a tak bychom museli nalezené části nějak posunout. Cílem je nalézt pro nás vhodnou kombinaci částečných souhlasů, které jsou nejlepší a jsou nejméně posunuty. Dobré je nejdříve nalézt vhodné souhlasy a poté postupně hledat vhodné kombinace. Je to jako hledat konstruovat graf, kde uzly grafu reprezentují jednotlivé části vzoru. Tento postup je uveden v [8].

Segmentace srovnávání se vzorem velmi výpočetně náročná, a to také v těch nejjednodušších případech, kdy vůbec nedochází ke geometrickým transformacím. Proces je možné urychlit vhodnou posloupností prováděných operací. Příkladem je zjištění míst obrazu, kde dochází s vysokou pravděpodobností ke souhlasu se vzorem. Potom není nutné prohledávat celou plochu. Dalšího zrychlení můžeme dosáhnout odhalením nesouhlasu dříve, než se otestují všechny obrazové body. Pokud je vzor v určitém okolí dobře korelován, pak je míra souhlasu vzoru s obrazem právě v tomto

okolí docela dobrá. Tvar průběhu hodnot míry je plochý nebo široký. Můžeme nejdříve prohledávat v obraze s menším rozlišením (méně operací a tím rychlejší) a potom vybranou lokaci upřesnit.

Další možný přístup je v každém obrazovém bodě, nebo v jeho okolí, vypočítat nějaké jednoduché vlastnosti a srovnávat nejdříve v místech, ve kterých jsou zjištěné vlastnosti podobné vlastnostem vzoru. Jako vhodná vlastnost se nabízí textura. Ta docela dobře charakterizuje okolí bodu (samozřejmě, když se nějak výrazně liší – např. významné body).

Ale výpočet určité vlastnosti nesmí být moc náročný, aby vyplatil. Když zjišťujeme míru souhlasu, tak je důležité zjistit nesouhlas co nejdříve, protože nesouhlasů je o mnoho více. Vezmeme-li míru souhlasu C_1 a C_2 v (5.5) resp. (5.6), tak musíme ukončit testování pozice hned, překročí-li hodnota (míra nesouhlasu) nějakou nastavenou mez. Je vhodné, aby míra nesouhlasu v každé pozici rostla co nejrychleji. Nejdříve jsou testovány části vzoru s vysokou pravděpodobností velkého rozdílu proti elementům obrazu. Příspěvky jsou v případě nesouhlasu větší, než kdybychom testovali míru souhlasu v náhodném pořadí.

5.5 Korelace

V této části si uvedeme několik základních měr, jakými můžeme porovnávat dvě matice. Tyto míry jsou důležité pro template matching.

Nejdříve si uvedeme definici korelace v jednorozměrném případě. V následujícím tvaru je uvedena v [13].

Korelace je algoritmus umožňující jednoduše vzájemně porovnávat dva digitalizované signály (posloupnosti vzorků). Výsledkem porovnání je opět posloupnost čísel vyjadřující podobnost dvou porovnávaných signálů a některé původně skryté vlastnosti signálů. Takto lze například zjišťovat periodicitu a periodu neznámého signálu, nalézt neznámou hodnotu zpoždění dvou stejných nebo podobných signálů, nebo objevit konkrétní hledaný signál skrytý v silném rušení (šumu).

Korelace je jedna ze základních měr mezi dvěma signály. V našem případě si uvedeme korelaci pro dvourozměrný případ. Je to jen rozšíření 1D.

5.5.1 Lineární korelace

V počítačovém vidění se často nazývá normalizovaná vzájemná korelace (angl. *normalized cross-correlation*, *NCC*). Lineární korelační koeficient $r(x, y)$ určuje, jak jsou hodnoty x_i a y_i svázány lineární funkcí. Hodnota se pohybuje v rozsahu $-1 \leq r(x, y) \leq 1$. Vodorovnou čarou se značí střední hodnota. Tady střední hodnota čísel x_1, \dots, x_n je \bar{x} a jejich variance (rozptyl) $\overline{(x - x_i)^2}$. Rozptyl je střední hodnota čtverců posunutých čísel tak, že střední hodnota je 0. Platí

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

$$\overline{(x - x_i)^2} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2, \quad \overline{(y - y_i)^2} = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

Výpočet se provede ve třech krocích.

1. Posuneme čísla x a y tak, aby měla střední hodnotu nula. Nová čísla označíme jako

$$x'_i = x_i - \bar{x}, \quad i = 1, 2, \dots, n$$

$$y'_i = y_i - \bar{y}, \quad i = 1, 2, \dots, n$$

2. Nově získaná čísla vydělíme tak, aby nová čísla měla jednotkovou varianci

$$x''_i = \frac{x'_i}{\sqrt{x'^2}}, \quad i = 1, 2, \dots, n$$

$$y_i'' = \frac{y_i'}{\sqrt{y_i'^2}}, \quad i = 1, 2, \dots, n$$

3. Čísla x_i'' a y_i'' jsou již normalizované a lineární korelační koeficient je dán vztahem

$$r(x, y) = \frac{1}{n} \sum_{i=1}^n x_i'' y_i'' \quad (5.8)$$

Tento koeficient má několik zajímavých vlastností, které si uvedeme.

1. je symetrický - $r(x, y) = r(y, x)$

2. je invariantní vůči lineární transformaci x a y . Pro čísla a_x , a_y , b_x a b_y platí

$$r(a_x x + b_x, a_y y + b_y) = r(x, y)$$

3. má rozsah $-1 \leq r(x, y) \leq 1$. Vztah pro výpočet koeficientu (5.8) můžeme brát jako skalární součin dvou vektorů z n -rozměrného lineárního prostoru. Vektory \mathbf{x}'' a \mathbf{y}'' mají kvůli jednotkové varianci a jednotkové střední chybě jednotkovou délku. To vyplývá z toho, že skalární součin dvou libovolných jednotkových vektorů leží v rozsahu -1 až 1. Bude-li $x_i = y_i$, pak $r(x, y) = 1$ a pokud bude $x_i = -y_i$, tak $r(x, y) = -1$.

5.5.2 Cross-correlation

V [2] je uveden výpočet vzájemné korelace trochu jiným způsobem. Ta se bere jako výpočet energie signálu. Ve 2D případě čtvercové matice (nebo 1D signál určen řádkami matice). Pokud budeme mít signál $a = (a_1, a_2, \dots, a_n)$, tak energie signálu je dán vztahem

$$E(a) = a_1^2 + a_2^2 + \dots + a_n^2.$$

Pokud ale budeme mít dva signály a a b , tak vzájemná energie (*cross-energy*) bude mít vztah

$$E(a, b) = a_1 b_1 + a_2 b_2 + \dots + a_n b_n.$$

Norma signálu $\|a\|$ je definována jako

$$\|a\| = \sqrt{E(a, a)}.$$

Z těchto veličin se získá normalizovaná vzájemná korelace jako

$$ncc(a, b) = \frac{E(a, b)}{\|a\| \cdot \|b\|}.$$

5.5.3 SDD míra

Další možnost, jak vypočítat podobnosti dvou obrazů je tzv. SDD míra (sum squared differences). Označíme obraz f a vzor jako h . SDD se vypočítá jako

$$sdd(x, y) = \sum_{(u,v) \in N} \left(f(x+u, y+v) - h(u, v) \right)^2,$$

kde N je dané okolí. Tam, kde je $sdd = 0$, jsou dva obrazy identické, v jiném případě je $sdd > 0$.

Kapitola 6

Geometrické transformace

V této kapitole si uvedeme geometrické transformace používané v DPZ, dále pak metody převzorkování a způsoby jak transformace ohodnotit.

6.1 Základní pojmy

Nejprve si vysvětlíme základní pojmy které souvisejí s geometrickými transformacemi a prostorovými daty. Níže uvedené definice jsou převzaty z [6].

Kartografická projekce - kartografické zobrazení, vznikající středovým promítáním bodů z referenční plochy na zobrazovací plochu.

Rektifikace - obecný proces transformace všech obrazových bodů z jednoho souřadnicového systému do jiného souřadnicového systému. Protože poloha stejného obrazového bodu je jiná, druhým krokem této transformace je převzorkování (tzv. resampling). Ale tento krok bude vysvětlen dále. Výsledkem rektifikace a převzorkování je, že můžeme dva obrazové záznamy přímo porovnávat - jsou ve stejném souřadnicovém systému a mají stejnou velikost obrazového bodu (stejně prostorové rozlišení).

Registrace (Image registration) - je to jeden ze způsobů rektifikace. Data jsou pořízena v různých soustavách a v různém rozlišení. Proto je nutné převedení do jednoho systému. Jde o proces, kterým se jedna souřadnicová soustava upraví tak, aby odpovídala souřadnicové soustavě jiného obrazového záznamu. Nemusí přitom nutně jít o souřadnou soustavu nějaké mapového zobrazení.

Georeferencování proces rektifikace během kterého jsou data transformována přímo do kartografické projekce a poloha každého obrazového bodu (pixelu) je vyjádřena v systému mapových souřadnic. Takto vznikají geokódovaná data.

Okrajově patří do geometrických transformací také *ortorektifikace*. Je to proces odstranění nepřesností, deformací a chyb v důsledku relativní změny polohy objektu, různé nadmořské výšky, vlivu reliéfu, geometrií snímání a nedokonalostí snímacích systémů. K tomu je potřeba digitální model terénu (DMT). Výsledným produktem jsou ortofotomapy. Ale to už je spíše záležitostí fotogrammetrie.

6.2 Geometrická transformace

Hlavním cílem geometrických transformací je odstranění zkreslení způsobeného nestálostí geometrických podmínek v průběhu měření nebo o mnoho častější převod dat do vhodné projekce. Do

této kategorie ještě patří tvorba nového obrazu na základě transformačních vztahů (např. ortofotomapy). Geometrickou korekci lze provádět třemi způsoby.

- transformace na základě přesně známých parametrů nosiče
- přímá geometrická transformace na základě vlíčovacích bodů nebo vektorů
- nepřímá geometrická transformace na základě vlíčovacích bodů nebo vektorů

Nejčastěji užívanou metodou je nepřímá metoda. Výhodou je, že výsledný obraz je pravoúhlý a celistvý. Postup se provádí pomocí tzv. *vlíčovacích bodů*, které můžeme jednoznačně identifikovat na snímku. Pomocí vlíčovacích bodů můžeme vypočítat koeficienty transformačních rovnic. Tyto koeficienty jednoznačně určují vztah mezi geometricky správným obrazem (georeferencovaným) a nekorigovanými daty. Výsledkem tohoto procesu jsou georeferencovaná data, někdy také nazývaná (geokódovaná). Tímto způsobem vytvořená data jsou ve stejném souřadnicovém systému a proto je lze vkládat jako vrstvy do GIS.

Ted' si zhruba popíšeme postup při geometrické transformaci. Nejdříve se definuje výstupný obraz. Tento obraz (matice) má prázdné buňky. Na základě již známých koeficientů transformačních rovnic se provádí vlastní proces výpočtu hodnot obrazové funkce. Způsob přenosu dat z původní matice do matice korigované se nazývá převzorkování (angl. resampling).

6.2.1 Homogenní souřadnice

U mnoha transformací se s výhodou využívají *homogenní souřadnice*. Homogenní souřadnice jsou speciální souřadnice bodů v prostoru. Pomocí nich můžeme zjednodušit maticové zápisy transformací. Použitím se transformace libovolného vektoru bodů se provede pouze maticovým násobením. Např. pokud budeme chtít transformovat body pomocí translace, tak v se běžně použije maticové sčítání. Po zavedení těchto souřadnic se dá toto maticové sčítání převést na maticové násobení. A to v velmi výhodné z hlediska výpočetního výkonu. Tyto souřadnice se nepoužívají jen v počítačovém vidění, ale také v počítačové grafice, teoretické mechanice a robotice.

Uspořádaná n -tice $[x_i, y_i, w]$ jsou pravoúhlé *homogenní souřadnice* bodu P v projektivním rozšíření¹, platí-li

$$x = \frac{x_i}{w}, \quad y = \frac{y_i}{w}. \quad (6.1)$$

Prvek w je tzv. homogenní složka. Převod z homogenních souřadnic do kartézských je realizován pouze dělením souřadnic homogenní složkou. Body, pro které je $w = 0$ se nazývají *nevlastní body*. Leží v nekonečnu a nelze je popsat pomocí kartézských souřadnic. Toto rozšíření může být v libovolném rozměru, ale nejčastěji v 2D a 3D.

6.3 Polynomiální transformace

Tento typ transformace se nejčastěji používá v dálkovém průzkumu Země pro georeferencování. Využití se také najde v aplikacích počítačového vidění. Koeficienty transformace se aproximují polynomem m -tého stupně. Nejvíce používané jsou prvního, druhého a třetího řádu. U transformací vyššího stupně se již dochází k nepříjemným deformacím. Speciálním případem je tzv. afinní transformace, která odpovídá transformaci prvního řádu.

Mezi nimi jsou rozdíly mimo jiné v počtu potřebných vlíčovacích bodů. K určení počtu bodů se využívá speciální vztah. Tak tedy počet vlíčovacích bodů se transformace m -tého řádu se je určen podle vztahu

$$n = \frac{(p+1)(p+2)}{2}, \quad (6.2)$$

kde p je stupeň polynomu transformace. Pro nejpoužívanější stupně polynomu jsou počty (včetně doporučených) uvedeny v tabulce (6.1).

¹Projektivní rozšíření prostoru (značí se $\mathbf{P}(E_3)$) vznikne doplněním eukleidovského prostoru o nevlastní body.

řád p	nutný počet bodů	doporučený počet bodů
1	3	4
2	6	10
3	10	15

Tabulka 6.1: Počty bodů pro polynomiální transformace

Ještě pro úplnost musíme uvést rovnice pro samotnou transformaci. Body x a y jsou vstupní body, body v původním obraze a u a v jsou v námi požadovaném souřadnicovém systému.

$$u = \sum_{i=0}^m \sum_{j=0}^{m-i} a_{ij} x^i y^j \quad (6.3)$$

$$v = \sum_{i=0}^m \sum_{j=0}^{m-i} b_{ij} x^i y^j \quad (6.4)$$

Stupeň polynomu je přirozeně m . My si zde podrobně uvedeme z polynomiálních transformací pouze afinní (1. řádu).

6.3.1 Afinní transformace – polynomiální transformace prvního řádu

Polynomiální transformace 1. řádu, nazývá se také afinní transformace (patří sem např. posun, rotace, změna měřítka, zkosení). Její hlavní a nejdůležitější vlastností je konformnost, tzn. zachovává v transformovaném obraze úhly.

Základní předpis má následující tvar

$$u = a_0 + a_1 x + a_2 y, \quad (6.5)$$

$$v = b_0 + b_1 x + b_2 y. \quad (6.6)$$

Kromě tohoto zápisu můžeme ještě použít maticový zápis. Pokud pro vstupní a výstupní souřadnice použijeme homogenní souřadnice, tak bude mít transformace následující tvar.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (6.7)$$

Ted' si ukážeme, jakým způsobem je možné vypočítat koeficienty a_0, a_1, a_2, b_0, b_1 a b_2 . Vždy je lepší použít více vřícovacích bodů než je potřeba. V tomto případě budeme mít přečtenou soustavu lineárních rovnic a k výpočtu použijeme metodu nejmenších čtverců (MNČ). Počet vřícovacích bodů si označíme jako n a vektory \mathbf{u} a \mathbf{v} urovnáme do matice o velikosti $2n \times 6$. Číslo 6 je počet hledaných koeficientů. Soustava rovnic v maticovém zápisu bude mít tvar

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{pmatrix}. \quad (6.8)$$

Využijeme maticový zápis a můžeme psát

$$\mathbf{Ax} = \mathbf{u} \quad (6.9)$$

a po úpravách

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^{-1} \mathbf{u} \quad (6.10)$$

se koeficienty, vektor \mathbf{x} , vypočítají jako

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{u}. \quad (6.11)$$

6.3.2 Polynomiální transformace vyšších řádů

Používají se jen polynomiální transformace druhého a třetího řádu. Níže uvedené vztahy jsou jen rozepsáním vztahů (6.3) a (6.4).

- Polynomiální transformace druhého řádu

$$u = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \quad (6.12)$$

$$v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \quad (6.13)$$

- Polynomiální transformace třetího řádu

$$u = a_0 + a_1 x + a_2 a_3 x^2 + a_4 xy + a_5 y^2 a + a_6 x^3 + a_7 x^2 y + a_8 xy^2 + a_9 y^3 \quad (6.14)$$

$$v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 + b_6 x^3 + b_7 x^2 y + b_8 xy^2 + b_9 y^3 \quad (6.15)$$

V praxi se používá transformace maximálně do třetího řádu. Řeší podobně metodou nejmenších čtverců jako v případě afinní transformace, včetně rozepsání do matice. Vyšší řády způsobují v obraze deformace a ty jsou nepříjemné. Polynomické transformace vyšších řádů mají tendenci transformovat body mimo území ohraničené referenčními body. Dvojice bodů je třeba vybírat co nejbližší k okrajům transformovaného obrazu a přesnost transformace ještě zpřesnit body nacházející se uvnitř území. Zvyšování řádu transformací již nepřináší další podstatnější zvýšení přesnosti.

6.3.3 Bilineární transformace

V praxi se často nahrazuje za polynomiální transformaci, je to zvláštní případ afinní transformace. Liší se posledním koeficientem u kombinace x a y souřadnice.

$$u = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy, \quad (6.16)$$

$$v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy, \quad (6.17)$$

6.4 Projektivní transformace

Mějme vlíčovací body ve dvou snímcích a naším cílem je určit transformační matici, která bude odpovídat projektivní transformaci. Body v jednom obraze (v původním) si označíme jako vektor $\mathbf{x} = [x_1, y_1]$ a vektor $\mathbf{u} = [u_1, u_1]$ v novém, transformovaném obraze. Tyto dva vektory jsou spojeny vztahem

$$\mathbf{u} = \mathbf{T} \mathbf{x}. \quad (6.18)$$

Obrazovou funkci \mathbf{f} označíme původní obraz a \mathbf{g} obrazovou funkci transformovaného obrazu. Poté platí

$$\mathbf{g}(\mathbf{u}) = \mathbf{f}(\mathbf{T}(\mathbf{u})), \quad (6.19)$$

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{T}^{-1}(\mathbf{u})). \quad (6.20)$$

Matice \mathbf{T}^{-1} je maticí inverzní transformace.

Projektivní transformace je taková, která spojuje dva souřadnicové systémy následujícím vztahy

$$u = \frac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}}, \quad (6.21)$$

$$v = \frac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}}. \quad (6.22)$$

Je možné také zapsat projektivní transformaci v maticovém vztahu.

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \quad (6.23)$$

Cílem je stejně jako v každé jiné transformaci vypočítat koeficienty a_{ii} . Čitatele těchto vztahů jsou hodně podobné afinní transformaci, ale v homogenních souřadnicích a jmenovatel je právě to známé vydělení homogenní složkou. Pokud vynásobíme rovnice jmenovatel, získáme

$$(a_{31}x + a_{32}y + a_{33})u = a_{11}x + a_{12}y + a_{13}, \quad (6.24)$$

$$(a_{31}x + a_{32}y + a_{33})v = a_{21}x + a_{22}y + a_{23}, \quad (6.25)$$

po roznásobení

$$a_{31}xu + a_{32}yu + a_{33}u = a_{11}x + a_{12}y + a_{13} \quad (6.26)$$

$$a_{31}xv + a_{32}yv + a_{33}v = a_{21}x + a_{22}y + a_{23} \quad (6.27)$$

a po odečtení členů z pravé strany získáme

$$a_{11}x + a_{12}y + a_{13} - xa_{31} - ya_{32} - ua_{33} = 0 \quad (6.28)$$

$$a_{21}x + a_{22}y + a_{23} - va_{31} - va_{32} - va_{33} = 0 \quad (6.29)$$

Tyto rovnice přepíšeme do maticového tvaru a pro n vřícovacích bodů.

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & -u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & v_1y_1 & v_1x_1 & -v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & -u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & v_2y_2 & v_2x_2 & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & u_nx_n & u_ny_n & -u_n \\ 0 & 0 & 0 & x_n & y_n & 1 & v_ny_n & v_nx_n & -v_n \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} = \mathbf{0} \quad (6.30)$$

Velikost matice je $2n \times 9$, kde číslo 9 je počet neznámých koeficientů transformace. Tuto soustavu rovnic vyřešíme. Pokud použijeme více bodů než je doporučený počet, tak se jedná o přeúčenou soustavu. Koeficienty a_{ii} označíme jako vektor $\mathbf{a} = [a_{11}, a_{12}, \dots, a_{33}]^T$ a matici vytvořenou z bodů jako \mathbf{C} .

$$\mathbf{Ca} = \mathbf{0} \quad (6.31)$$

Tuto rovnici vynásobíme zleva \mathbf{C}^T

$$\mathbf{C}^T\mathbf{Ca} = \mathbf{C}^T\mathbf{0}. \quad (6.32)$$

Vektor neznámých koeficientů získáme následujícím způsobem

$$\mathbf{a} = (\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{0}. \quad (6.33)$$

6.5 Převzorkování

Po geometrických transformacích následuje další důležitá část, a to hledání hodnoty jasu každého transformovaného bodu. Souřadnice výchozích bodů x, y jsou celočíselné hodnoty obrazové funkce $f(x, y)$. Po geometrické transformaci, nové souřadnice již nemusí být celočíselné. K dispozici máme pouze hodnoty jasů v původním obraze. Naším cílem je tedy nějakým způsobem získat nové hodnoty jasů v novém, transformovaném obraze. To se provede pomocí aproximace jasů.

V teorii aproximace se dostupné vzorky proloží polynomem dvou proměnných. Získá se analytická rovnice a po dosazení vypočítáme jasové hodnoty v transformovaném obraze. Přesnost ovlivňuje kvalitu výstupního obrazu. Pro přesnější aproximaci je nutné použít větší okolí bodu.

Aproximuje se jas ve vstupním obraze, který odpovídá jasu hledaného bodu v výstupní mřížce. Pokud budeme chtít vypočítat jas bodu v celočíselných souřadnicích (u, v) ve výstupním obraze, tak nejdříve musíme mít souřadnice bodů (x, y) v původním obraze. To provedeme použitím inverzní transformace.

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{u} \quad (6.34)$$

V obecném případě získáme reálná čísla (na obrázku (6.1) označeny čárkovanými čarami). Ty nesouhlasí s celočíselným rastrem. Jedinou informací o průběhu spojitě jasové funkce je její vzorkovaná hodnota. Abychom získali jasovou hodnotu $f_s(\Delta x, \Delta y)$ v bodě (x, y) , musíme tu hodnotu odhadnout. Výsledkem aproximace je jas $f_n(x, y)$, kde n je nějaké označení jednotlivé metody. Jas můžeme vyjádřit jako dvourozměrnou konvoluci

$$f_n(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_s(\Delta x, \Delta y) h_n(x - \Delta x, y - \Delta y) \quad (6.35)$$

Funkce h_n je konvoluční jádro. To obvykle pokrývá pouze malou část okolí bodu, kvůli výpočetním nárokům. Hodnota jádra mimo okolí je nulová.

Obvykle se používají jednoduché aproximační polynomy. Uvedeme si tři nejpoužívanější podle složitosti. Mají známější názvy.

- *Metoda nejbližšího souseda* (angl. nearest neighbour, interpolace nejbližším sousedem) určí hodnotu jasu v bodě (x, y) hodnotou nejbližšího bodu v rastrové mřížce. Na obrázku (6.1) je v pravé části uvedeno konvoluční jádro v 1D případě. Levá část ukazuje jak je přiřazena nová hodnota jasu. Čárkovaný rastr ukazuje, jak je transformovaný rastr zobrazen inverzní transformace. Plné čáry ohraničují rastr vstupního obrazu. Hodnota jasu se získá ze vztahu

$$f_1(x, y) = f_s(\text{round}(x), \text{round}(y)) \quad (6.36)$$

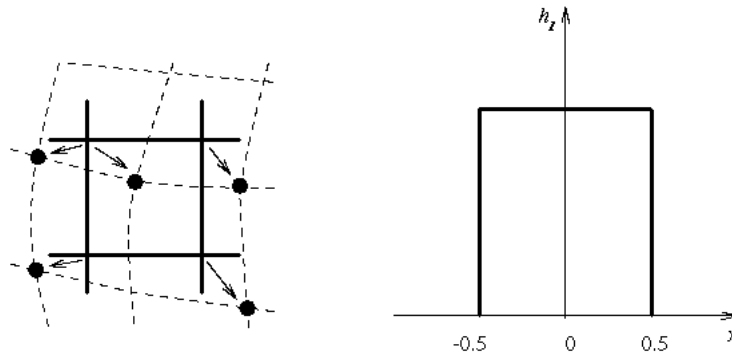
Chyba v poloze může dosáhnout maximálně hodnoty jedné poloviny obrazového bodu. Např. v obraze, kde se nacházejí nějaké rovné objekty natočené šikmo vůči rastru, je tato chyba dobře rozeznatelná. Objektu jsou schodovité.

- *Bilineární interpolace* využívá čtyřech sousedních bodů body (x, y) . Výsledný jas je kombinací těchto čtyř bodů. V levé části obrázku 6.2 je zobrazeno, jakým způsobem je se okolní body podílejí na interpolaci a pravá část interpolací jádro v 1D případě. Tato interpolace je dána vztahem

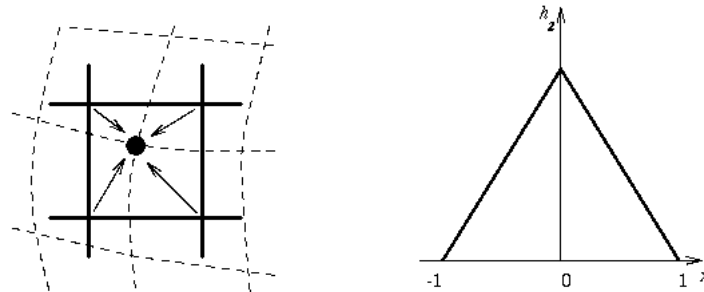
$$f_2(x, y) = (1-a)(1-b)f_s(l, k) + a(1-b)f_s(l+1, k) + b(1-a)f_s(l, k+1) + abf_s(l+1, k+1), \quad (6.37)$$

kde l, k jsou celé části hodnot x, y a $a = x - l$ a $b = y - k$.

Tato interpolace způsobuje malé snížení rozlišení kvůli tomu, že je obraz vyhlazován lineárním filtrem. Toto vyhlazení je mnohem menší než schodovitý efekt v předchozím případě. Výrazné zlepšení je znát na šikmých čarách nebo hranách.



Obrázek 6.1: Interpolace jasů metodou nejbližšího souseda (převzato z [1])



Obrázek 6.2: Interpolace jasů metodou pomocí bilineární interpolace (převzato z [1])

- *Bikubická interpolace* bere k výpočtu jasů už 16 bodů. Obrazová funkce je interpolována bikubickým polynomem a tím se zpřesňuje obrazový model funkce. Interpolační jádro by bylo ve dvourozměrném případě hodně nepřehledné a tak ho uvedeme v 1D případě.

$$f_3(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{pro } 0 \leq |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{pro } 1 \leq |x| < 2 \\ 0 & \text{jinde} \end{cases} \quad (6.38)$$

Toto jádro je vidět na obrázku (6.3). Tato interpolace zachovává detaily v obraze mnohem lépe než v předchozích případech.

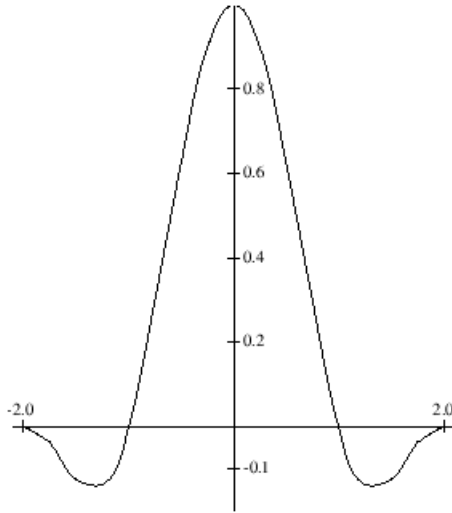
Mimo těchto tří metod existuje ještě mnoho dalších, ale my si ještě uvedeme interpolaci pomocí kubické je B-spline křivky (uvedena v [7]). Konvoluční jádro má následující tvar

$$f_4(x) = \begin{cases} \frac{1}{2}|x^3| + |x^2| + \frac{2}{3} & 0 \leq |x| < 1, \\ -\frac{1}{6}|x^3| + |x^2| - 2|x| + \frac{4}{3} & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| \end{cases} \quad (6.39)$$

Tato interpolaci se též nazývá *Parzenovo okno*.

6.5.1 Sinc filtr

Existuje způsob, jak můžeme funkci rekonstruovat přesně. A to s využitím Fourierovy transformace. Mějme funkci, která je frekvenčně omezená a obsahuje nejvyšší frekvenci f_s . Rekonstrukční ideální filtr bude mít ve Fourierově oblasti tvar obdélníku šířkou f_s rozloženými symetricky kolem



Obrázek 6.3: Interpolace jasů pomocí bikubické konvoluce (převzato z [1])

počátku. Tato funkce bude ideální nízko pásmový filtr. Odstraňuje všechny frekvence vyšší než f_s . Pro rekonstrukci potřebujeme její Fourierův vzor. A ten se jmenuje *sinc*

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad (6.40)$$

Pokud budeme rekonstruovat spojitou funkci s frekvencí f_s , tak ji musíme rekonstruovat s $\text{sinc}(f_s x)$. Tato funkce je neomezená, a proto se pro výpočet používá ze vstupní diskrétní funkce všechny její hodnoty. Pro výpočet jednoho bodu musíme použít celý obraz. Právě proto je tato metoda na výpočet velmi náročná. Nevýhodou je také to, že nabývá záporných hodnot.

Použití interpolační metody závisí na konkrétní aplikaci, protože pro některé obrazy je nutné zachovat ostré hrany, zatímco v jiných případech není rozmazání na škodu.

6.6 Hodnocení přesnosti transformace

V transformaci obrazového záznamu je nutné zjistit kvalitu provedené transformace. K tomu se využívají tzv. *RMS chyby* (angl. root mean square error).

6.6.1 RMS chyba

Je to vzdálenost mezi původní polohou vřícovacích bodů a přetransformovanou polohou těch samých bodů. Jinými slovy je to rozdíl mezi požadovaným cílovým souřadnicovým systémem a současným systémem, který je transformován pomocí transformační matice. RMS chyba se vypočítá podle vztahu

$$RMS = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2},$$

kde x_r a y_r jsou původní souřadnice a x_i a y_i jsou transformované souřadnice.

RMS chyba je vyjádřena jako vzdálenost v původním souřadnicovém systému. Pokud jsou obrazové záznamy ve snímkových souřadnicích tak je tato chyba určena v pixelech. Např. když RMS chyba je 2, tak to znamená, že původní pixel je přibližně 2 pixely od přetransformovaného.

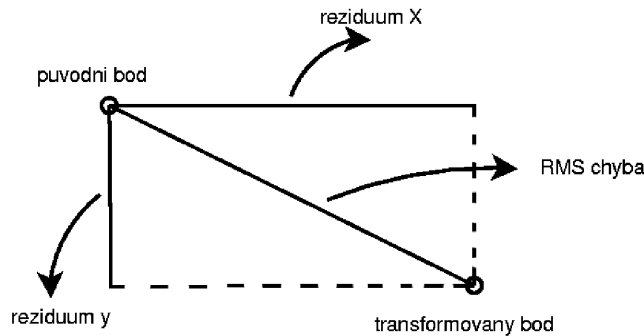
6.6.2 Rezidua a RMS chyby každého bodu

Často se používají rezidua pro x a y souřadnice. Jsou to vzdálenosti mezi původními a transformovanými souřadnicemi pouze v jednom směru. Reziduum pro x je vzdálenost mezi původní a transformovanou x souřadnicí. To samé platí také pro y souřadnici.

RMS chyba pro každý vřícovací bod pomáhá zjistit kvalitu vybraného bodu. Tuto chybu si označíme jako r_i , a vypočítá se podle vztahu

$$r_i = \sqrt{xr_i^2 + yr_i^2},$$

kde xr_i^2 a yr_i^2 jsou rezidua pro i -itý bod. Na obrázku (6.4) je vidět vztah mezi rezidui a RMS chybou pro každý bod.



Obrázek 6.4: Rezidua a RMS chyba pro každý bod

6.6.3 Celková RMS chyba

Z reziduí je možné vypočítat celkovou RMS chybu. Nejdříve si musíme říci, jak se spočítají celkové chyby pouze ve směrech x a y . Ty se vypočítají podle následujících vztahů. Celková RMS chyba ve směru x r_x

$$r_x = \sqrt{\frac{1}{n} \sum_{i=1}^n xr_i^2} \quad (6.41)$$

a celková chyba v y r_y

$$r_y = \sqrt{\frac{1}{n} \sum_{i=1}^n yr_i^2}, \quad (6.42)$$

kde n je počet vřícovacích bodů. Pro celkovou RMS chybu platí

$$T = \sqrt{r_x^2 + r_y^2}. \quad (6.43)$$

Pokud do rovnice (6.43) dosadíme vztahy (6.41) a (6.42) získáme nejpoužívanější podobu výpočtu této chyby:

$$T = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n xr_i^2 + yr_i^2 \right)}.$$

Normalizovaná hodnota, která reprezentuje RMS chybu každého bodu (je ve vztahu k RMS chybě) je také důležitá hodnota kvality transformace. Příspěvek každého bodu RMS chyby se získá

$$E_i = \frac{r_i}{T}, \quad (6.44)$$

kde r_i je chyba pro i -tý bod a T je celková RMS chyba transformace. Součet všech těchto příspěvků

$$\sum_{i=1}^n E_i = 1.$$

6.6.4 Tolerance

Hodnoty RMS chyby dosahují při výběru různých bodů různých hodnot. Záleží na konkrétní kombinaci více či méně vhodných bodů. Obecně je možné říci, že hodnota RMS chyby kolem dvou až čtyř pixelů je považována za přesnou.

Kapitola 7

Realizace algoritmu pro hledání korespondencí

Naším cílem je nalézt na dvou snímcích odpovídající body. Z těchto bodů vypočítat koeficienty transformace a nerektifikovaný snímek transformovat do požadovaného systému. Informace o souřadnicovém připojení nám poskytne tzv. *worldfile*, který je současně dodán se samotným obrazem. K výslednému převzorkovanému obrazu, který je již v souřadnicovém systému vytvoříme *worldfile* s potřebnými souřadnicemi. K tomu budeme z větší části využívat metody popsané v předchozích kapitolách.

Algoritmus se skládá z následujících částí:

- nalezeneme vhodné body v dvou obrazech – vybereme pouze významné body, ty které se nějak od svého okolí odlišují
- ke každému bodu z jednoho obrazu najdeme bod, který je k němu co nejvíce podobný – odpovídající si body
- náhodně vybereme určitý počet korespondujících bodů a vypočítáme transformační matici
- transformujeme všechny body z jednoho obrazu do systému druhého a transformaci ohodnotíme
- vybereme ty dvojice bodů, pro které je transformace nejlepší
- nakonec obraz pomocí té nejlepší transformace transformuje do cílového systému, převzorkujeme a vytvoříme *worldfile*.

Toto je jen stručný nástin, jak algoritmus bude pracovat. Jednotlivé části budou podrobně popsány v následujících částech.

Takže pokud vše shrneme, jako vstup budeme mít dva obrazy, z nichž jeden je ve snímkových souřadnicích a druhý v požadovaném souřadnicovém systému, přetransformujeme první obraz a vytvoříme *worldfile*.

V následujícím textu budeme nazývat souřadnicově nepřipojený obraz jako levý nebo první obraz a rektifikovaný obraz jako pravý (popř. druhý). Toto označení je pro snazší orientaci.

7.1 Formát *worldfile*

Worldfile je obyčejný textový ASCII soubor, který obsahuje informace souřadnicovém připojení rastrového obrazu. Extenze souboru je podle toho, jakému rastrovému formátu přísluší. Pokud máme formát TIF, tak končí jako *.tfw*, u formátu PNG *.pfw*, u JPG *.jfw* a podobně. Soubor obsahuje následujících šest od sebe oddělených řádek s vysvětlením a označením.

Řádek 1: A: velikost obrazového bodu ve směru osy x v mapových jednotkách
 Řádek 2: D: rotace kolem osy y
 Řádek 3: B: rotace kolem osy x
 Řádek 4: E: velikost obrazového bodu ve směru osy y v mapových jednotkách
 Řádek 5: C: x souřadnice středu systému horního levého rohu v mapových jednotkách
 Řádek 6: F: y souřadnice středu systému horního pravého rohu v mapových jednotkách

Těchto šest hodnot odpovídá šesti transformačním parametrů u afinní transformace. Členy B a D jsou většinou nula, ty ovlivňují rotaci. Poslední dva řádky, s koeficienty C a F , jsou to jen posun od počátku systému a A , E je měřítko v jednotlivých směrech. Afinní transformace se skládá ze tří částí, z rotace, posunu a změny měřítka.

Koeficienty A až F mají v transformaci následující umístění (včetně maticového zápisu v kartézských souřadnicích).

$$u = Ax + By + C$$

$$v = Dx + Ey + F$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} A & B \\ D & E \end{pmatrix} + \begin{pmatrix} C \\ F \end{pmatrix}$$

Tady si uvedeme vzorový worldfile ukazka.tif pro formát TIF obrazu ukazka.tif.

```
32.0
0.0
0.0
-32.0
691200.0
4576000.0
```

Dále si uvedeme jak převádět snímkové souřadnice na souřadnice daného systému, včetně inverzní transformace. Snímkové souřadnice budeme značit (i, j) , jako v matici řádky a sloupce (to ve skutečnosti odpovídá) a souřadnice systému jako (x, y) . Počátek označíme jako x_0, y_0 a velikost pixelu ve směru x a y jako px , resp. py .

transformace ze snímkových souřadnic $(i, j) \rightarrow (x, y)$

$$x = x_0 + px \cdot j$$

$$y = y_0 + py \cdot i$$

transformace do snímkových souřadnic $(x, y) \rightarrow (i, j)$

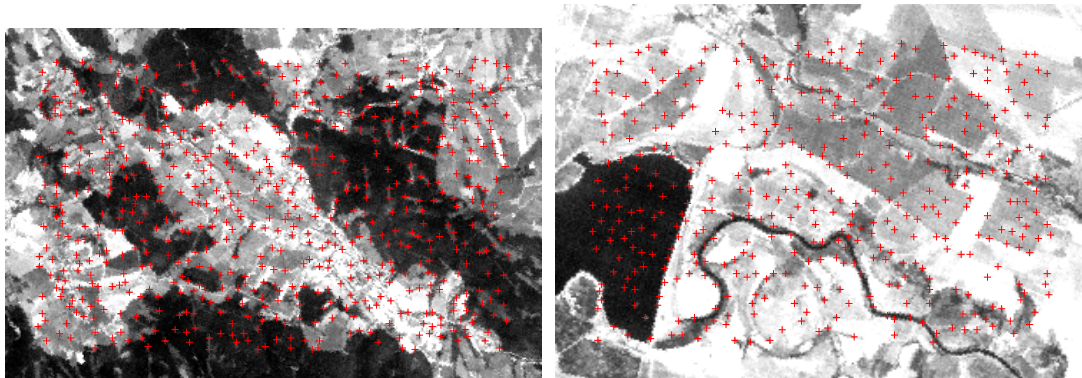
$$i = \frac{y - y_0}{py}$$

$$j = \frac{x - x_0}{px}$$

7.2 Výběr vhodných bodů

Detekce bodů je jedna ze základnějších částí realizace celého algoritmu. Právě výběr těch správných bodů je důležitý, protože vybrané body si musí odpovídat. Pokud tomu tak není, nelze nalézt správné korespondence.

Metod, jak body detekovat je mnoho. Některé z nich jsou uvedeny v kapitole o detekci významných bodů. Z praktických zkušeností byla vybrána metoda Harrisova operátoru. Vzhledem k tomu, že



Obrázek 7.1: Ukázka detekce bodů pomocí Harrisova operátoru na testovacích snímcích

okolí stejných bodů pro snímky z DPZ jsou někdy výrazně odlišné, osvědčil se právě tento detektor. Je vzhledem k ostatním relativně starý a je již dostatečně odzkoušen. Možná menší nevýhodou je větší výpočetní náročnost.

Celá metoda je podrobně popsána v již zmíněné kapitole o detekci bodů, a tak se jí už nebudeme zabývat. Za pozornost ale stojí jiný způsob odstranění bodů ležících blízko sebe (tzv. *non-maxima suppression*). Používáme k tomu statistické filtrování (*order-statistical filtering*). Jen pro připomenutí, je aplikováno až po prahování nad odezvou operátoru.

Statistické filtrování funguje na podobném principu jako jakékoliv jiné, tedy za pomoci konvoluce. Jako konvoluční jádro se použije čtvercová matice $n \times n$ (ta obsahuje jedničky a nuly, určují použité body v okolí – ekvivalent strukturního elementu v matematické morfologii) a číslo, které určuje pořadí elementu v seřazeném okolí jádra, které neobsahují nuly. Toto číslo určující pořadí se počítá od levého horního rohu doprava (v dalších řádcích počítá postupně dále) a nazývá řád (proto *order-statistics filtering*). Tak např. jednotková matice o velikosti 3×3 a řád 5 je obyčejný mediánový filtr. U stejné masky s řádem 1, je to minimální filtr a s řádem 9 maximální filtr. Matice

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

s řádem 4 nahradí v obraze maximem ve svém čtyř okolí. To se dá samozřejmě zobecnit na jakkoliv velké jádro.

Pro odstranění bodů, které leží blízko sebe (*non-maxima suppression*) se použije výše popsané filtrování. Jedním z parametrů je tzv. *radius*, ten určuje okolí jednoho určitého bodu, ve kterém se nevyskytuje žádný jiný bod. V naší implementaci se jako maska použije jednotková matice o velikosti $n = 2 \text{ radius} + 1$ a jako řád číslo n . Tedy pro *radius* 11 máme jednotkovou matici 23×23 a řád 23. Pro filtrování se použije konvoluce a při takto velké masce je výpočet dost časově náročný.

Dalším parametrem pro detekci bodů Harrisovým operátorem je práh. Po výpočtu odezvy a výsledku statistického filtrování, se použije prahování. Naprahovaný binární obraz nazveme b . Využijeme vztahu

$$b = ((odezva == \text{stat_filt}) \text{ and } (odezva > \text{prah}))$$

a tento výsledek se vynásobí (jako dvě matice, člen po členu) a teprve až z této matice se vyberou námi požadované významné body. Binární obraz obsahuje pouze jedničky a nuly. Vyberou se body obsahující v matici jedničky.

Práh nemá velký vliv na výběr bodů a proto se v konečném programu jako parametr nepoužije. Předem je daná hodnota, zjištěná experimentálně na 100. Dalším parametrem je ještě σ . Ten ovlivňuje velikost výpočet odezvy z parciálních derivací (u výpočtu druhých derivací se použije jádro a to je stejné jako u Gaussova filtru).

Vybrané body jsou samozřejmě detekovány u okrajů. Ty jsou pro nás nepřijatelné, protože dále budeme brát u každého bodu jeho okolí a tak bychom se dostali mimo obraz. Ty body u okrajů se musí odstranit minimálně do vzdálenosti poloviny maximální použité velikosti okolí. Jinak by to mohlo dále způsobovat problémy.

Z obou obrazů máme určeny souřadnice v snímkových souřadnicích, neboli v pixelech. Ty poté použijeme dále.

7.3 Nearest-neighbor strategy

Teď máme detekovány body v obou obrazech a potřebujeme ke každému každému body v levém obraze přiřadit bod, který je nějakým způsobem nejvíce podobný druhému. Podobnost budeme určovat na základě okolí. U této části budeme postupovat hrubou silou, tento postup se anglicky nazývá *nearest-neighbor strategy*.

Vytvoříme si strukturu 3D matice. Ta bude mít ve prvních dvou rozměrech nějaké okolí bodu (dále uvidíme, že čtvercové a kruhové) o velikosti w a třetí rozměr bude počet bodů. Tuto matici vytvoříme pro obě množiny vybraných bodů. Celková velikost bude matice $w \times w \times n$, kde n je počet detekovaných bodů v obraze.

Pro každý bod z levé množiny bodů vezmeme příslušné okolí a porovnáme ho s každým okolím z druhé množiny bodů. Vypočítáme parametr značící podobnost a vybereme ten bod, jehož okolí má ten parametr nejmenší a ten bod je nejpodobnější. To samé uděláme pro každý bod z první množiny. Výsledkem budou dvojice bodů, které si vzájemně odpovídají. Počet těchto bodů bude stejný jako počet bodů z první množiny. Algoritmická složitost toho hledání bude $O(n^2)$. Při detekci bodů v obrazech v řádu stovek bude celkový počet porovnání řádově desetitisíců.

Takto vybrané dvojice obsahují mnoho bodů, které jsou určeny špatně. A to z mnoha důvodů, např. šumu. Pro určité okolí bodu se může nacházet v druhém obraze více podobných bodů. Podobnost se určuje vypočítanou hodnotou z okolí. Může se stát, že správné okolí korespondujícího bodu má větší hodnotu podobnosti (dále uvidíme, že nejpodobnější okolí bude mít nejmenší hodnotu), než to chybné. Ale podle našich kritérií jsou korespondující body správné. Tyto chyby budeme muset dále odstranit. Počet chybně nalezených dvojic závisí na konkrétních datech. Pro různá data se musí různá okolí a různé metody porovnávání. Metody porovnání a různé vzdálenosti budou popsány dále. Pokud nebude v tomto hledání úspěšný a body neurčíme dobře, nebudeme schopni nalézt správné korespondence.

Celý nearest-neighbor algoritmus bude lépe srozumitelný v následujícím pseudokódu.

```
% p, q jsou dvojice bodů
% n, m - počet bodů v levém, resp. v pravém obraze
% pl, pp - body z levého a pravého, které si odpovídají

pl = p % hledají se nejpodobnější body pro body z levého obrazu

for i = 1:n

    maxdist = Inf % přiřazení nějaké velké hodnoty

    dl = vektor z okolí bodů nebo histogram z bodu levého obrazu

    for j = 1:m

        dp = vektor z okolí nebo histogram z bodu pravého obrazu

        d = eukleidovská nebo chi2 vzdálenost

        if (d < maxdist)
```

```

maxdist = d % aktualizace maximální vzdálenosti
jindx = j % index zatím nejpodobnějšího bodu

end if

end for

pp(i) = q(jindx) % do pravých bodů se přiřadí ten nejpodobnější

end for

```

7.3.1 Použitá okolí bodů

Okolí může být více druhů, ale my budeme používat pouze klasické čtvercové a také ještě kruhové. Parametrem pro okolím je jeho velikost.

Čtvercové okolí

Toto okolí má čtvercový tvar o velikosti n . Kolem bodu o souřadnicích (x, y) se vyberou obrazové body do dvou stran ve směru x a y do vzdálenosti $(1/2 w - 1)$. Počet obrazových bodů je $n \times n$. Na obrázku (7.2) je u jednoho vybraného bodu zobrazeno čtvercové okolí.



Obrázek 7.2: Čtvercové okolí bodu testovacího obrazu

Zde jsou zobrazeny konkrétní číselné hodnoty okolí o velikosti 11×11 z body z předchozího obrázku (7.2).

41	41	47	69	108	149	149	142	135	149	188
54	47	41	47	76	122	149	155	168	188	211
41	41	41	47	61	102	155	183	188	192	214

41	41	41	47	54	76	149	192	192	183	202
41	41	47	47	47	61	122	199	202	183	173
47	41	35	41	47	47	82	149	195	183	149
41	41	35	35	35	41	61	102	155	168	135
47	47	54	61	69	82	102	135	149	115	128
135	142	122	122	128	142	162	168	142	108	122
173	155	149	142	155	162	155	155	149	122	122
149	142	135	135	142	142	142	135	142	135	122

Kruhové okolí

Daleko zajímavější je kruhové okolí bodu, má také jako čtvercové okolí velikost w . Ve čtverci nejsou všechny obrazovými body, ty mají právě tvar kruhu a ostatní jsou vyplněny jinými hodnotami. Takovými, které se nemohou být v obrazové matici při běžném kódování 8 nebo 10 bity. My jsme použili hodnoty -1. Samozřejmě je možné použít jiné hodnoty.

Kruhové okolí se získá tak, že nejdříve vybere čtvercové okolí, tak jak bylo popsáno výše. Z tohoto čtvercového okolí se udělá kruhové, tzn. na správných místech se hodnoty obrazových bodů nahradí hodnotami -1. Ve čtvercové matici se určí střed (pouze relativně) jako

$$stred = floor\left(\frac{1}{2}w\right) + 1,$$

kde *floor* značí pouze celou část čísla. Pro všechny body určíme, zda se v bodě nechá hodnota obrazového bodu, nebo se nahradí číslem -1. Pro každý bod vypočítáme číslo

$$(i - stred)^2 + (j - stred)^2$$

a pokud je je číslo menší nebo rovno hodnotě w^2 , což je jen velikost okolí a zároveň poloměr kruhu (kruhového okolí), tak se hodnoty obrazové matice ponechají. V opačném případě se změni na -1. Lépe to je vidět na následujícím pseudokódu.

```
% matice je čtvercové okolí o velikosti w
% a také velikost výsledného kruhového okolí
% kruh je výsledné kruhové okolí

stred = floor(1/2*w)+1;

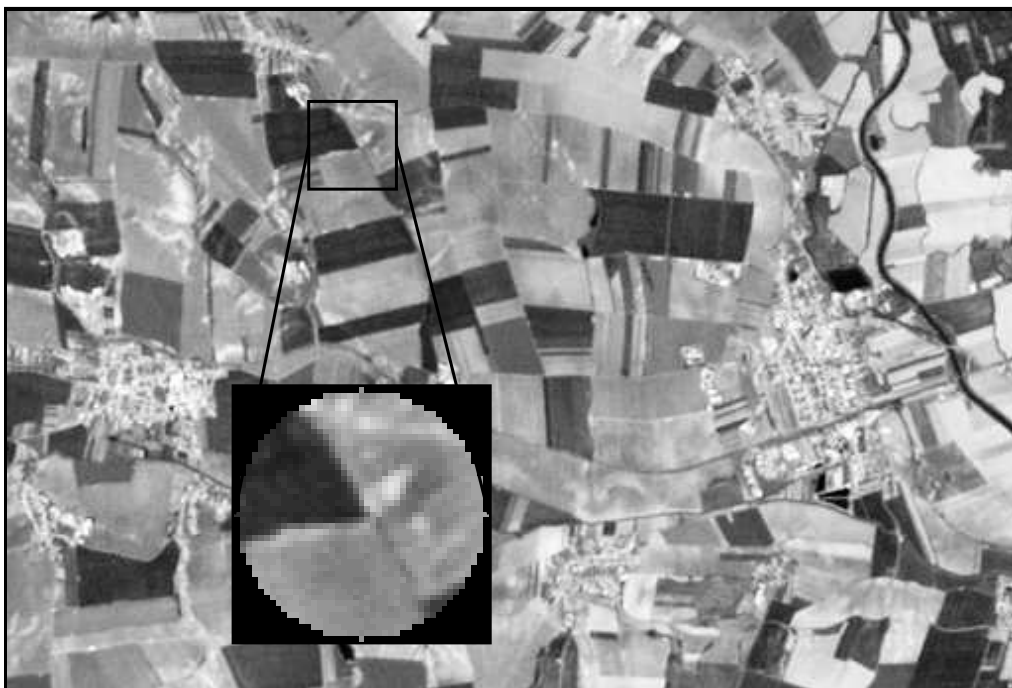
for i = 1:n
    for j = 1:n

        hodnota = (i - stred).^2 + (j - stred).^2;

        if (hodnota <= w^2)
            kruh(i,j) = matice(i,j);
        else
            kruh(i,j) = -1;
        end if

    end for
end for
```

Kruhové okolí má jednu obrovskou výhodu, a to že je invariantní vůči otočení. Tato vlastnost je pro nás mimořádně důležitá, protože budeme hledat body, které jsou mimo jiné také otočené. Na obrázku (7.3) je vidět kruhové okolí jednoho bodu na testovacím obrázku. Pozadí mimo kruh je černé, tam hodnoty nabývají -1. Konkrétní číselné hodnoty vybraného okolí o velikosti 11×11 jsou bodu z obrázku (7.3). Také v hodnotách je dobře vidět roh významného bodu.



Obrázek 7.3: Kruhové okolí bodu testovacího obrazu

-1	-1	-1	-1	-1	149	-1	-1	-1	-1	-1
-1	-1	41	47	76	122	149	155	168	-1	-1
-1	41	41	47	61	102	155	183	188	192	-1
-1	41	41	47	54	76	149	192	192	183	-1
-1	41	47	47	47	61	122	199	202	183	-1
47	41	35	41	47	47	82	149	195	183	149
-1	41	35	35	35	41	61	102	155	168	-1
-1	47	54	61	69	82	102	135	149	115	-1
-1	142	122	122	128	142	162	168	142	108	-1
-1	-1	149	142	155	162	155	155	149	-1	-1
-1	-1	-1	-1	-1	142	-1	-1	-1	-1	-1

7.4 Metody ohodnocení podobnosti bodů

Pokud máme ke každému detekovanému bodu přiřazené čtvercové nebo kruhové okolí. Musíme mít nějaký aparát, jak budeme okolí porovnávat, abychom zajistili, že vybíráme metodou nearest-nearest strategy odpovídající bod. My si zde popíšeme dvě metody, které v algoritmu použijeme. A to porovnání pouze jasů společně s výpočtem eukleidovské vzdálenosti a metodu porovnání histogramu s výpočtem χ^2 vzdálenosti.

7.4.1 Porovnání jasů

Tato metoda je jednoduchá. Ze čtvercového nebo kruhového okolí vytvoříme vektor a ten obsahuje hodnoty jasů. Velikost vektoru je $n^2 \times 1$, resp. u kruhového okolí o něco menší. Samozřejmě u kruhového okolí vynecháme záporné hodnoty mimo kruh, ty nepočítáme.

Vektor jasů bereme souřadnice bodu v n -dimenzionálním prostoru a vypočítáme eukleidovskou

vzdálenost d . Jen pro připomenutí se použije vztah

$$d(u, v) = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

kde u a v jsou vektory jasu a n je délka (velikost) vektoru. Při hledání nejpodobnějšího okolí používáme právě tuto hodnotu.

Nevýhodou tohoto postupu je, že vstupní obrazy musí mít jasové hodnoty přibližně stejné (např. aritmetické průměry jasů v obrazech by měly být blízké). V opačném případě budou vzdálenosti větší, ale podobnost by měla být zachována. Jinak je možné je vstupní obrazy normalizovat, tak aby měly průměry jasů přibližně podobné.

Místo eukleidovské vzdálenosti by bylo určitě možné použít např. šachovnicovou nebo vzdálenost v městských blocích. Ale použití těchto metrik nemá žádnou výhodou, protože se vypočítá jen jiná hodnota a tu potom porovnáváme.

7.4.2 Porovnání histogramu

Metoda porovnávání histogramu je o něco složitější. Pro obě okolí vytvoříme histogram (u kruhového vynecháme záporné hodnoty jako v předchozím případě). Histogram je vektor, jehož délka je u 8-bitového obrazu 256, určující počet jasových hodnot pro každou hodnotu 0 až 255. Takto vytvořený histogram se musí normalizovat, tzn. vydělit součtem všech hodnot v histogramu. Všechny čísla histogramu jsou menší než jedna a součet všech hodnot odpovídá číslu jedna.

Teď vypočítáme vzdálenost mezi dvěma histogramy. Ne eukleidovskou, ale χ^2 vzdálenost. Ta je definována jako

$$d(u, v) = \sqrt{\sum_{i=1}^n \frac{(u_i - v_i)^2}{u_i + v_i}}$$

kde u a v jsou vektory jasu a n je délka vektoru. Právě pro porovnávání histogramů je nevhodnější tato vzdálenost.

Pro tuto vzdálenost jsou obdobné problémy jako u eukleidovské vzdálenosti. Ty nejsou nijak závažné a dají se odstranit normalizací jasu.

7.5 Ransac algoritmus

V této části již máme dvojice korespondujících bodů v obou obrazech. Počet těchto dvojic je stejný jako počet vybraných bodů v levém obraze. Jak už jsme také psali předtím, problémem je, že ne všechny body si odpovídají. Hledali jsme pro každý bod ten, co nejpodobnější. Je jedno, v jak velkém, v jakém tvaru okolí a jakou metodou jsme ohodnocovali míru shody. V těchto dvojicích je velmi mnoho bodů, které jsou určeny chybně (nespojují tytéž body scény). Pravděpodobnost, že body budou určeny správně roste s větším okolím a přesto se tam mohou vyskytovat body určené podle některé míry jako odpovídající, ale ve skutečnosti nejsou. Pokud budeme brát eukleidovskou vzdálenost mezi okolími, tak může stát, že narazíme na špatné body s nejmenší vzdáleností.

Ransac je zkratka z angl. *Random Sample Consensus*. Je to algoritmus, který odhaduje parametry matematického modelu z měřených dat (pozorování). Tyto data obsahují odlehlé hodnoty (angl. outliers). Poprvé byl publikován Fischlerem and Bollesem v roce 1981. Základní princip je následující. Předpokládáme, že máme data, které obsahují správné hodnoty (po dosažení splňují model - inliers) a zároveň outliery. Ty nespĺňují matematický model. Data samozřejmě obsahují šum. Odlehlé hodnoty mohou vzniknout extrémní hodnotou šumu, nesprávnou hypotézou nebo nesprávnou interpretací dat. Algoritmus v některých případech může předpokládat, že při daném, malém počtu inlierů, může odhadnout parametry modelu optimálně tak, že je vyrovná.

Výhodou algoritmu je schopnost robustně odhadovat parametry (odhadne parametry s vysokým stupněm přesnosti). Nevýhodou je, že neznáme čas za který odhad parametrů proběhne.

Pokud budeme chtít odhady v nejkratším čase, tak výsledek nebude optimální (existuje lepší řešení). Nejvýznamnější aplikace jsou v počítačovém vidění, např. pro hledání parametrů přímk, kuželoseček z detekovaných bodů a také při hledání korespondencí.

My použijeme jako měřená data detekované body. Ne ty původní, ale ty korespondující, které jsme určili pomocí nearest-neighbour strategy. Dvojice bodů, které si odpovídají, ale některé jsou určeny chybně. Správné dvojice jsou v algoritmu inliery a ty nesprávné outliery. Matematickým modelem je pro nás transformační matice, protože existuje přesný vztah mezi dvěma obrazy. Naším cílem odhadnou parametry modelu, tedy parametry transformace. Kromě toho získáme korespondující body a to je náš hlavní cíl.

Celý algoritmus si popíšeme pro naše účely, speciálně určený pro hledání korespondencí. V obecném případě, jak je popsán v literatuře, je trochu nepřehledný. Náhodně vybereme z odpovídajících bodů n bodů. Tento počet určuje počet korespondencí (pro afinní transformaci jsou více než 3). Z těchto vybraných bodů vypočítáme transformační matici. Všechny detekované body z levého obrazu (toho nerektilifikovaného) transformujeme do souřadnicového systému pravého obrazu. Pokud je transformační matice správná tak námi transformované z levého obrazu se musí shodovat se souřadnicemi pravého obrazu. Tohoto ideálního stavu se nikdy nedosáhne, protože transformace obsahuje vždy nějakou chybu. Vypočítáme nějakou míru, která transformaci ohodnotí. Určíme počet bodů přetransformovaných z levého obrazu, které leží na místě korespondujících bodů v pravé obraze s nějakou tolerancí. Tolerance by se měla pohybovat podle našich nároků na přesnost někde kolem 4 až 6 obrazových bodů. Napoprvé bude pravděpodobně počet správně transformovaných bodů malý. A tak budeme postup opakovat v předem určeném počtu iterací. Budeme hledat takovou n -tici bodů a odpovídající transformační matici, u které je nejvíce správně transformovaných bodů. Takovou nalezneme, a tu prohlásíme za hledanou společně s odpovídajícími body.

Parametry, které ovlivňují chování algoritmu jsou hledaný počet korespondencí, počet iterací a tolerance. Nelze předem určit, kdy přesně nalezneme správnou transformaci. Také při opětovné aplikaci algoritmu na stejná data, dostaneme vždy jiný výsledek, tedy jinou transformační matici a jiné korespondující body.

Princip algoritmu je přehledně zobrazen na následujícím pseudokódu, společně s vhodnými parametry.

```
% pl, pp - bodu z levého a pravého obrazu, mají stejný počet

% u pl a pp doplníme body o homogenní složku (kvůli transformaci)
pl = doplnění pl o homogenní složku
pp = doplnění pp o homogenní složku

pocet = 0
tolerance = 4
max_pocet_koresp = 0
pocet_iteraci = 1000
pocet_korespondenci = 6

% RANSAC algoritmus
while (pocet <= pocet_iteraci)

    pocet = pocet + 1

%    náhodný výběr čtyř korespondujících bodů (pouze indexy)
    r = náhodný výběr čísel (indexů), maximálně však počet korespondencí

%    tady už výběr samotných bodů pro transformaci
    b_leve = náhodný výběr bodů z levého obrazu
    b_prave = náhodný výběr bodů z pravého obrazu
```

```

%   výpočet transformace
T = výpočet transformační matice z b_leve a b_prave

%   transformace všech bodů do systému v pravého obrazu
pp_transf = pl*T;
pp_transf = převod z homogenních souřadnic na kartézské

%   rms chyba pro každý pár, potřebujeme v následujícím kroku
rms = výpočet rms chyby

%   výběr páru v toleranci a jejich spočítání
soucasny_pocet_koresp = určení počtu bodu v toleranci

%   hledá transformace pro největší počet správných bodu
if (soucasny_pocet_koresp > max_pocet_koresp)

%       uložení do maximálního počtu bodů
max_pocet_koresp = soucasny_pocet_koresp

%       uložení transformační matice
A = T

%       uložení korespondujících bodů
bodyl = b_leve
bodyp = b_prave

end if

end for

```

V naší implementaci v hlavním cyklu po výběru n -korespondujících bodů, otestuje, zda vzájemně mezi body je dostatečná vzdálenost. Protože pokud je vzdálenost mezi body moc malá, tak transformace není přesná. Ideální je, když jsou body rovnoměrně rozmístěny v obraze. Minimální vzdálenost je dána jako 15 pixelů, protože při detekci bodů se zadává parametr ovlivňující minimální vzdálenost bodů (u statistického filtrování, non-maxima suppression) a jeho maximální hodnota je dána jako 15. Proto nemá menší vzdálenost smysl.

Pro bližší informace o algoritmu RANSAC je možné získat v [18], [19] a [20].

7.5.1 Výpočet transformační matice

Pro výpočet transformační matice afinní transformace byl použit postup popsáný v [2]. Tam se nazývá jako *Best 2D Affine Transformation*. Máme dvě množiny bodů $([x_i, y_i], [u_i, v_i])$, kde $i = 1, \dots, n$. Minimální počet bodů musí být tři. Transformace je svázána vztahem

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Chyba v bodech se přenese na transformaci. Bodů se používá více, a proto se pro výpočet parametrů transformace použije metody MNČ. Vezmeme funkci

$$\xi(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}) = \sum_{i=1}^n ((a_{11}x_i + a_{12}y_i + a_{13} - u_i)^2 + (a_{21}x_i + a_{22}y_i + a_{23} - v_i)^2),$$

z ní určíme šest parciálních derivací funkce podle šesti neznámých položíme výraz nule.

$$\frac{\partial \xi}{\partial a_{11}} = 0, \frac{\partial \xi}{\partial a_{12}} = 0, \frac{\partial \xi}{\partial a_{13}} = 0$$

$$\frac{\partial \xi}{\partial a_{21}} = 0, \frac{\partial \xi}{\partial a_{22}} = 0, \frac{\partial \xi}{\partial a_{23}} = 0$$

Po úpravách dostaneme šest normálních rovnic v maticové formě

$$\begin{pmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i & 0 & 0 & 0 \\ \sum x_i y_i & \sum y_i^2 & \sum y_i & 0 & 0 & 0 \\ \sum x_i & \sum y_i & \sum 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ 0 & 0 & 0 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ 0 & 0 & 0 & \sum x_i & \sum y_i & \sum 1 \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{pmatrix} \begin{pmatrix} \sum u_i x_i \\ \sum u_i y_i \\ \sum u_i \\ \sum v_i x_i \\ \sum v_i y_i \\ \sum v_i \end{pmatrix}$$

Vyčíslíme hodnoty v maticích a soustavu rovnic vyřešíme nějakou běžnou metodou pro výpočet lineárních rovnic a získáme námi požadovaných šest neznámých parametrů afinní transformace.

Právě u této transformace není vhodné použít velký počet bodů automatickém hledání. Experimentálně jsme zjistili, že není vhodný počet větší než 8. Když jsou nějaké body určeny chybně, tak vyrovnání MNC najde takovou transformaci, kde se minimalizují rezidua. Navenek to vypadá jako dobrý výsledek, ale body jsou chybně určené a převzorkovaný obraz je silně deformovaný. To je největší nevýhoda tohoto postupu. Právě z tohoto důvodu není možné v našem algoritmu vyhledávat větší počty bodů.

7.6 Vytvoření worldfilu pro převzorkovaný obraz

Pokud máme korespondující body, z nich vypočítáme transformační matice a běžným způsobem převzorkujeme vhodnou metodou. Takže další částí je obraz souřadnicově připojit, tzn. vytvořit worldfile.

K dispozici máme jak převzorkovaný obraz, tak původní rektifikovaný obraz. Na tyto dva obrazy, které si jsou hodně podobné, aplikujeme výše popsany algoritmus RANSAC pro hledání korespondujících bodů. Máme odpovídající si body, body z pravého a rektifikované obrazu. Ty pravého převedeme ze snímkových souřadnic do daného souřadnicového systému (pomocí informací worldfilu). Body si vzájemně odpovídají, z levého jsou ve snímkových souřadnicích a body z pravého v souřadnicovém systému. Ty poslední body v pravém, u nich známe obě souřadnice v obou souřadnicových systémech. Body v levém obraze jsou prakticky stejné jako v pravém (až na nějaké chyby z transformace), ale liší se snímkovými souřadnicemi (i, j) . Z nich vypočítáme, za předpokladu stejného stejné velikosti obrazových bodů, souřadnice levého horního rohu převzorkovaného obrazu. K tomu použijeme následující vztahy.

$$x_0 = x - px \ j$$

$$y_0 = y - py \ i$$

Souřadnice (x, y) a (i, j) jsou stejné body v převzorkovaném obraze, ale v jiných souřadnicových systémech. Poté máme již všechny informace o souřadnicovém připojení a nic nám nebrání zapsat je do textového souboru.

Kapitola 8

Zhodnocení a výsledky algoritmu

V této kapitole si uvedeme výsledky algoritmu na testovacích datech. A to postupně od detekce bodů, přes hledání korespondencí a převzorkování. Uvedeme obrázky s detekovanými body a vybranými korespondencemi, dále tabulku s nalezenými korespondujícími body a jejich souřadnice včetně RMS chyb. Nebude chybět ani převzorkovaný obraz.

8.1 Testovací data

Jako testovací data jsme použili severních Čech snímky z družic Landsat 4 (MSS), Landsat 5 (TM), Landsat 7 (ETM+) a Spot 5. Ty jsou volně z dostupného datasetu pro GRASS 6.0, který je k dispozici na adrese

http://gama.fsv.cvut.cz/~landa/geowikicz_data/zod/datasety/sevcech.tar.gz,

dalších dat na jako samotstatné snímky ve formátu TIF

http://gama.fsv.cvut.cz/~landa/geowikicz_data/zod/cv7/etm-jtsk.tar.gz a

http://gama.fsv.cvut.cz/~landa/geowikicz_data/zod/cv7/mss-xy.tar.gz.

Vhodné výřezy z dat byly v GRASSu 6.2 vyexportovány do formátu TIF pomocí modulu `r.out.tiff`. Potom na tyto obrázky byl aplikována afinní transformace s nejrůznějšími parametry. Pro tyto deformace byl použit Matlab. U snímků z Landsatu se vytvořily panchromatické, podle empirického vztahu pro převod barevného obrazu na šedotónový.

$$gray = 0,3 red + 0,59 green + 0,11 blue,$$

ale pokud za barevná pásma `red`, `green`, a `blue` dosadíme odpovídající pásma ze senzorů Landsatu (např. pro ETM), tak získáme

$$etm = 0,3 etm3 + 0,59 etm2 + 0,11 etm1,$$

kde `etm` je panchromatické pásmo, `etm1` je první, `etm2` druhé a `etm3` třetí pásmo senzoru ETM. Analogicky je možné použít další MSS a TM. U Spotu je použito druhé pásmo.

Pro jednoduchost byly worldfilly vytvořeny uměle, takže ve skutečnosti nejsou snímky lokalizovány ve své skutečné poloze.

8.2 Výstupy algoritmu

Jako testovací snímky pro tuto kapitolu byly použity data ze Spotu. Jedná se stejné území a na pravý obraz byla aplikována afinní transformace, přesněji rotace a různá změna měřítka pro oba směry, jak pro x , tak pro y . Ty jsou na obrázcích (8.1) a (8.2).

Na těchto dvou obrazech teď budeme aplikovat metodu detekce bodů pomocí Harrisova operátoru. Byly použity parametry $\sigma = 1,0$ a $radius = 11$. Počet detekovaných bodů v levém obraze je 195, v pravém 222. Výsledek jsou na obrázcích (8.3) a (8.4).



Obrázek 8.1: Původní nerektifikovaný snímek

Po detekci významných bodů přistoupíme k hledání korespondencí. Použijeme následující parametry:

- počet korespondencí: 6
- počet iterací: 10000
- maska: 21
- tolerance: 4
- typ okolí: čtverec
- typ prohledávání: jas

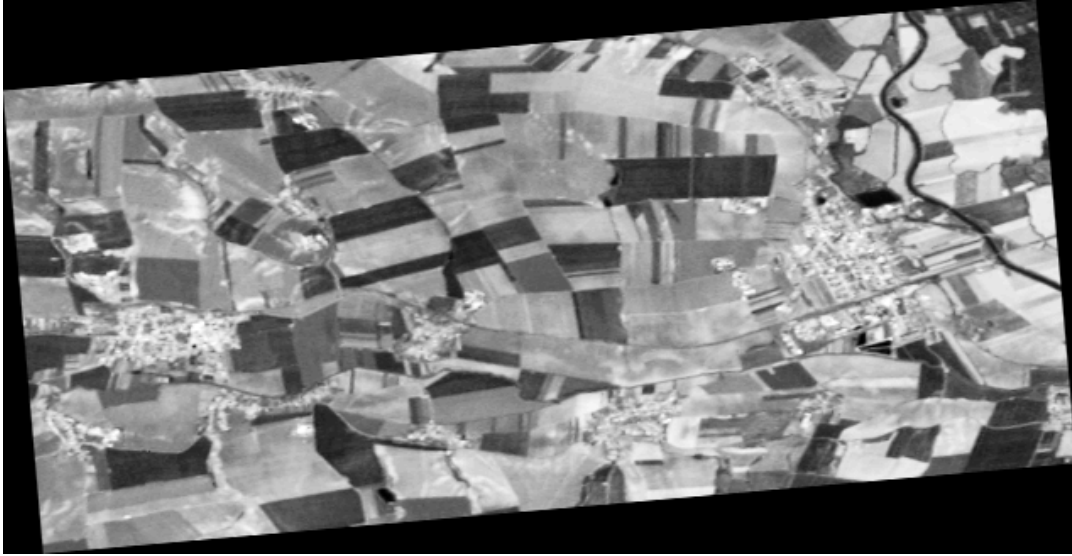
Nalezené korespondence na obou snímcích jsou na obrázcích (8.5) a (8.6).

V tabulce (8.1) tabulka nalezených odpovídajících si bodů. První dva sloupce jsou body z bodů levého obrazu, jsou ve snímkových souřadnicích a ve druhém sloupci jsou jim korespondující body z pravého obrazu. Ty jsou v tomto případě v systému S-JTSK. V další tabulce (8.2) pro výše uvedené body vypočítány RMS chyby pro každý bod ve směru osy x , y a RMS pro jednotlivý bod. Celková RMS chyba z těchto šesti nalezených bodů je 3,25, samozřejmě je v jednotkách souřadnicového systému, tedy v metrech.

Z těchto bodů byly vypočítány koeficienty transformace $a_{11}, a_{12}, \dots, a_{23}$, které byly použity pro převzorkování. Pokud se uspořádají do matice, tak bude mít tvar

$$A = \begin{pmatrix} -3,106 & 35,825 & 0 \\ 23,917 & 2,146 & 0 \\ -846242,943 & -922253,649 & 1 \end{pmatrix}.$$

Na posledním obrázku (8.7) je již převzorkovaný levý obraz, ten je systému S-JTSK. Metoda k převzorkování byla použita bikubická interpolace.



Obrázek 8.2: Původní rektifikovaný snímek

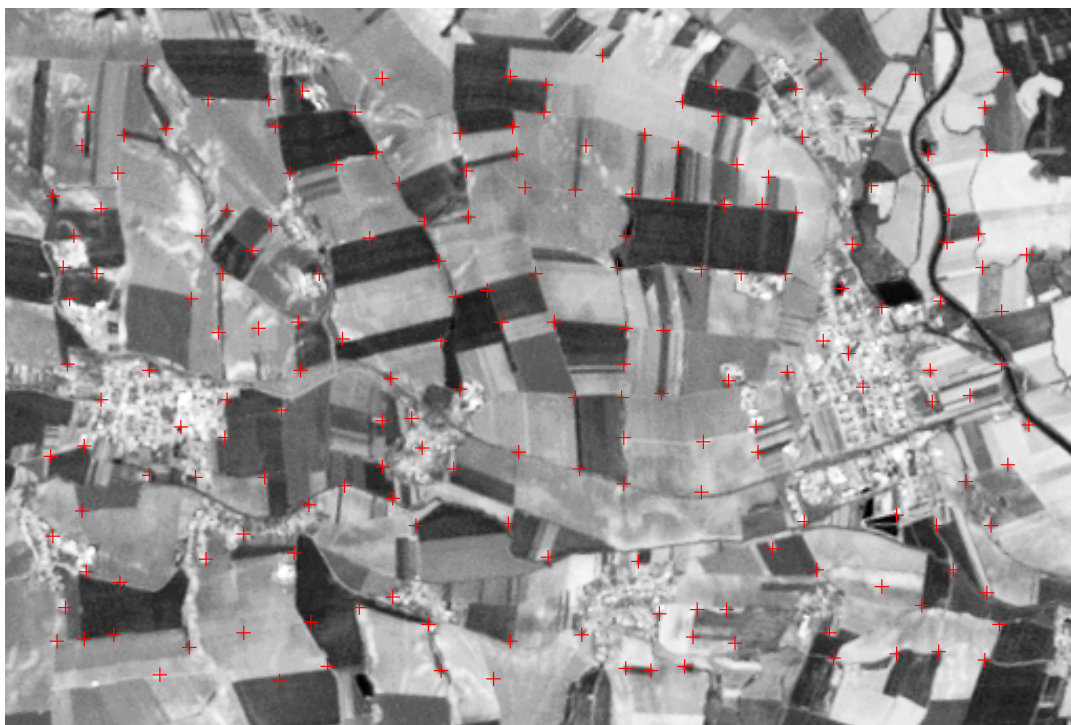
bod	i	j	x	y	x_{transf}	y_{transf}
1	344	287	-840432,86	-909319,36	-840447,35	-909314,02
2	46	187	-841903,17	-920206,73	-841913,30	-920204,40
3	402	165	-843553,52	-907489,80	-843545,45	-907498,01
4	241	302	-839772,73	-911978,48	-839768,60	-912971,78
5	177	69	-845143,85	-915767,80	-845142,52	-915764,59
6	88	305	-839232,61	-918437,16	-839221,53	-918446,53

Tabulka 8.1: Nalezené korespondence v testovacích snímcích (druhý a třetí sloupec jsou snímkové souřadnice, další dva body v souřadnicovém systému a poslední dva jsou transformované snímkové souřadnice)

8.3 Celkové zhodnocení

Podle vizuálního posouzení byly významné body detekovány celkem dobře, pokud se pozorně podíváme na obrázky (8.5) a (8.6), tak body jsou detekovány přibližně na stejném místě. A to v případě, že druhý obraz je otočen a ještě zdeformován rozdílným měřítkem v obou směrech. Jednotlivé RMS chyby pro body nepřesahují hodnotu 120 m, protože jsme jako toleranci vzali 4 obrazové body (velikost pixelu v cílovém souřadnicovém systému je 30 m). Tolerance byla volena experimentálně. Pokud bychom použili, menší hodnotu, tak není jisté, že se nám vůbec podaří nalézt nějakou vhodnou transformaci.

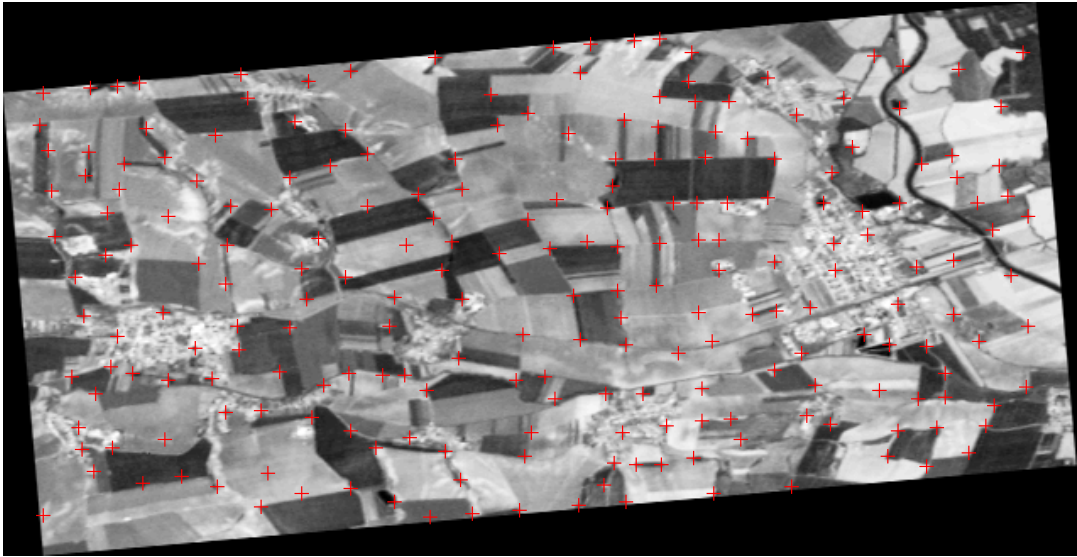
K posouzení je možné použít další testovací snímky, které jsou uvedeny příloze, a to včetně obrázků z jednotlivých kroků.



Obrázek 8.3: Detekované body na nerektifikovaném snímku

bod	r_x	r_y	RMS bodu
1	14,48	-5,33	15,43
2	10,13	-2,33	10,39
3	-8,07	8,21	11,57
4	-4,13	-6,70	7,87
5	-1,33	-3,21	3,48
6	-11,09	9,37	14,51

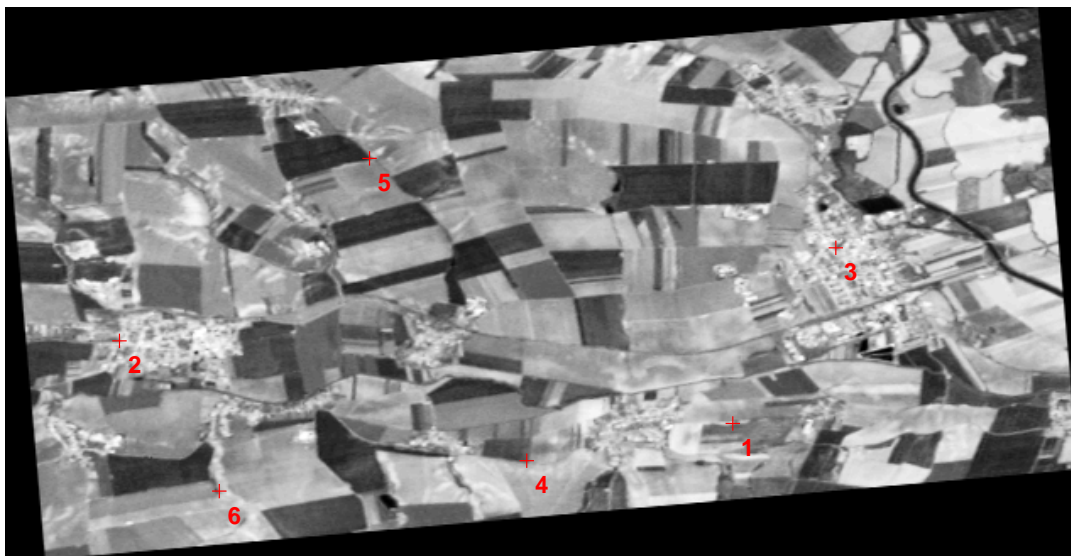
Tabulka 8.2: U nalezených korespondencích RMS chyby u jednotlivých bodů (celková chyba 3,25)



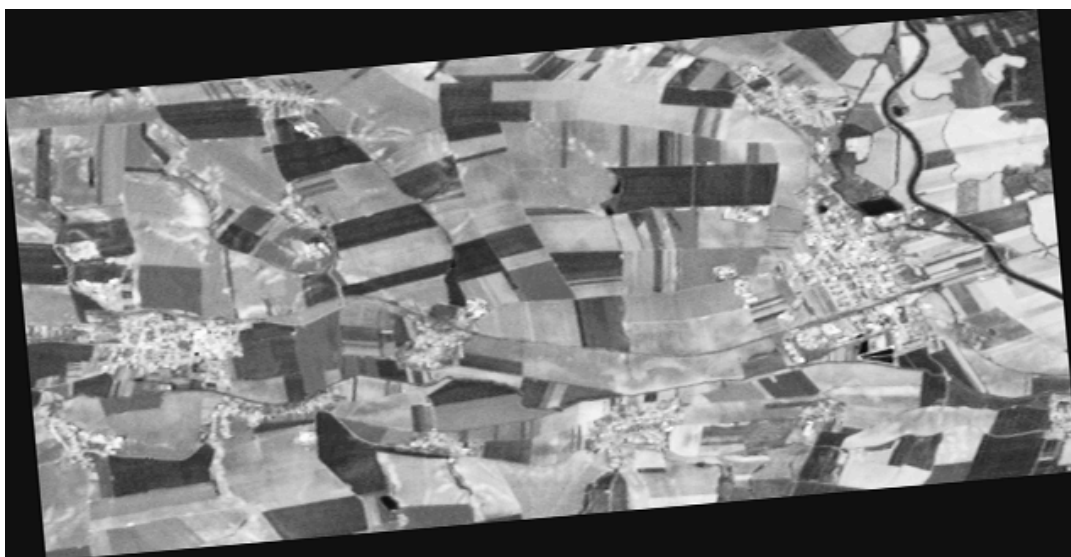
Obrázek 8.4: Detekované body na rektifikovaném snímku



Obrázek 8.5: Nalezené korespondence na nerektifikovaném na snímku



Obrázek 8.6: Nalezené korespondence na rektifikovaném snímku



Obrázek 8.7: Převzorkovaný obraz v cílovém systému

Kapitola 9

Závěr

První část práce se zabývala popisem metod, které byly při samotné realizaci používány. Jedná se především o detekci hran (pouze okrajově), detekci významných bodů, ohodnocením a porovnáváním okolí hledaných bodů a geometrickými transformacemi. V praktické části byly nejvíce používány právě detekce významných bodů a geometrické transformace.

Pro detekci bodů byla konkrétně zvolena jedna metoda, a to Harrisův operátor. Ten byl vybrán jako jeden z nejstarších a nejnámějších, který současně dává velmi dobré výsledky, mimo jiné právě na družicových datech. Detekuje s velkou pravděpodobností stejná místa také v případech, kdy jsou na obrazy aplikovány geometrické transformace (afinní a dokonce projektivní). Tento fakt byl zjištěn experimentálně.

V praktické části byl navržen algoritmus pro automatické vyhledávání vlíčovacích bodů. Ten je v jedné kapitole podrobně popsán a některé části také pomocí pseudokódu.

Implementace algoritmu byla provedena v prostředí Matlab 6.5 (R13). Bylo vytvořeno grafické uživatelské rozhraní, pomocí kterého se aplikace ovládá. Práce s pomocí rozhraní je jednoduchá a intuitivní. Načtou se vstupní obrazy (nerektifikovaný a rektifikovaný s worldfilem), poté se spustí detekce bodů. Dále následuje samotné hledání korespondencí. Pro detekci bodů a hledání korespondencí je důležité zadat správné parametry (jejich význam je popsán v uživatelském manuálu v příloze). Vybrané body a nalezené korespondence jsou v aplikaci zobrazeny. Správnost bodů je určena pomocí RMS chyb (ty je možné uložit do textového souboru). Pokud jsou korespondence nalezeny správně, tak je vypočítána transformace mezi dvěma systémy a nerektifikovaný obraz je převzorkován. Převzorkovaný obraz je uložen společně vytvořeným worldfilem, který určuje souřadnicové připojení.

Algoritmus byl testován na testovacích datech. Byly to volně dostupné snímky ČR z družic Landsat a Spot. Na takových datech jsou korespondence nalezeny celkem spolehlivě, zejména u těch, které byly pořízeny stejným senzorem (viz. kapitola o hodnocení algoritmu a příloha). Pokud nejsou odpovídající si body nalezeny správně, tak je nutné algoritmus aplikovat znovu. Nepovede-li se nalezení bodů po několika opakování, tak pravděpodobně nebude možné korespondence na těchto datech nalézt.

Zadání diplomové práce, tedy samotné automatické nalezení korespondujících si bodů a její implementace, bylo beze zbytku splněno. Body jsou nalezeny v předem dané toleranci a vysokou pravděpodobností. Další vývoj algoritmu by měl směřovat k nalezení nějaké sofistikovanější metody pro popis a pro porovnávání okolí bodů, dále určitou optimalizaci v časové oblasti a rozšíření o použití méně častých polynomiálních transformací druhého a třetího řádu.

Tato diplomová práce byla vysázena v systému L^AT_EX. Ke psaní zdrojového kódu byl použit textový editor Vi a pro vytváření a úpravu obrázků aplikace Dia a ImageMagick, vše pod operačním systémem GNU/Linux.

Literatura

- [1] Šonka M., Hlaváč V., Boyle R., *Image Processing, Analysis, and Machine Vision*, 1999, PWS/Brooks and Cole Publishing, ISBN 0-534-95393-X
- [2] Stockman G., Shapiro L. G., *Computer Vision*, 2001, Prentice Hall PTR, ISBN 0130307963
- [3] Richards J. A., Jia X., *Remote Sensing Digital Image Analysis: An Introduction*, 1999, Springer-Verlag New York, Inc., ISBN 0387548408
- [4] Ballard D. H., Brown Ch. M., *Computer vision*, 1999, Prentice Hall Inc., ISBN 0-13-165316-4
- [5] Pratt W. K., *Digital Image Processing*, 2001, John Wiley & Sons Inc., ISBN 0-471-22132-5
- [6] Dobrovolný P., *Dálkový průzkum Země, Digitální zpracování obrazu*, 1998, Masarykova univerzita v Brně, ISBN 80-210-1812-7
- [7] Žára J., Beneš B., Sochor J., Felkel P., *Moderní počítačová grafika*, 2004, Computer Press, ISBN 80-251-0454-0
- [8] Hlaváč V., Šonka M., *Počítačové vidění*, 1992, Grada, ISBN 80-85424-0-3
- [9] Pavelka K., *Fotogrammetrie*, 2003, Západočeská univerzita v Plzni, ISBN 80-7082-972-9
- [10] Jähne B., *Digital Image Processing*, 2002, Springer-Verlag Berlin, Inc., ISBN 3-540-67754-2
- [11] Harris C., Stephens M., *A Combined Corner and Edge Detection*, 1988, Proceedings of The Fourth Alvey Vision Conference, p. 147–151
- [12] U.S. Geological Survey – Landsat Project, *Extraction of GCP chips GeoCover using Modified Moravec Interest Operator (MMIO) algorithm*, 1998, http://landsat.usgs.gov/resources/files/MMIO_GeoCover_Control_White_Paper.pdf
- [13] Vojáček A., *Algoritmus korelace v digitálním zpracování signálů*, 2006, Automatizace – server z oblasti automatizace
- [14] Phillips D., *Image Processing in C*, 2000, R & B Publications <http://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/>, ISBN 0-13-104548-2
- [15] Fisher R., Perkins S., Walker A., Wolfart E., *Image Processing Learning Resources*, 2004, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>
- [16] Spletzer J., *Real-time Image Processing for Autonomous Robot Systems, Course Materials*, 2004, http://www.cse.lehigh.edu/~spletzer/cse398_Fall04/lec005.ppt
- [17] Urban M., *Harris Interest Operator*, 2003, http://cmp.felk.cvut.cz/cmp/courses/dzo/resources/lecture_harris_urban.pdf

- [18] Fischler M. A., Bolle R. C., *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, Volume 24 Issue 6 (červen 1981), str. 381–295, ACM Press, <http://doi.acm.org/10.1145/358669.358692>
- [19] Wikipedia, *RANSAC* – *Wikipedia, The Free Encyclopedia*, 2007, <http://en.wikipedia.org/w/index.php?title=RANSAC&oldid=11635818>
- [20] Václav Hlaváč, *RANSAC*, 2007, <http://cmp.felk.cvut.cz/~hlavac/Public/TeachingLectures/RANSAC.pdf>

Dodatek A

Uživatelský manuál

Tato aplikace je realizace praktické části diplomové práce Automatické vyhledávání vlíčovacích bodů pro polynomiální rektifikaci.

A.1 Základní informace a popis aplikace

Popis, jakým způsobem aplikace pracuje a jaké algoritmy a postupy používá si zde neuvedeme. Vše je zdokumentováno v samotném textu diplomové práce. Uvedeme pouze základní práci s programem, jeho spuštění, zadání a popis jednotlivých parametrů a výstupy.

Umožňuje zadání nerektifikovaného obrazu a již souřadnicově připojeného obrazu. První obraz se převzorkuje do souřadného systému druhého obrazu. A to vše automaticky bez zásahu uživatele, pouze po zadání vhodných parametrů.

Aplikace byla naprogramována v jazyce výpočetního systému Matlab. Byl použit ve verzi 6.5 (R13). Funkční by měla být také ve vyšších verzích, ale ve nižších verzích není funkčnost zaručena. Využito bylo vnitřního grafického uživatelského rozhraní Matlabu GUIDE.

Jedná se pouze jedno okno, na kterém se načtou vstupní obrazy, zadávají všechny parametry a spouštějí se jednotlivé části. Práce s ní je velmi jednoduchá a intuitivní. Problémem může být jen zadávání parametrů. Ty se musí pro každá data nejdříve vyzkoušet a budou vysvětleny v následujících částech.

Všechny skripty aplikace jsou na přiloženém CD.

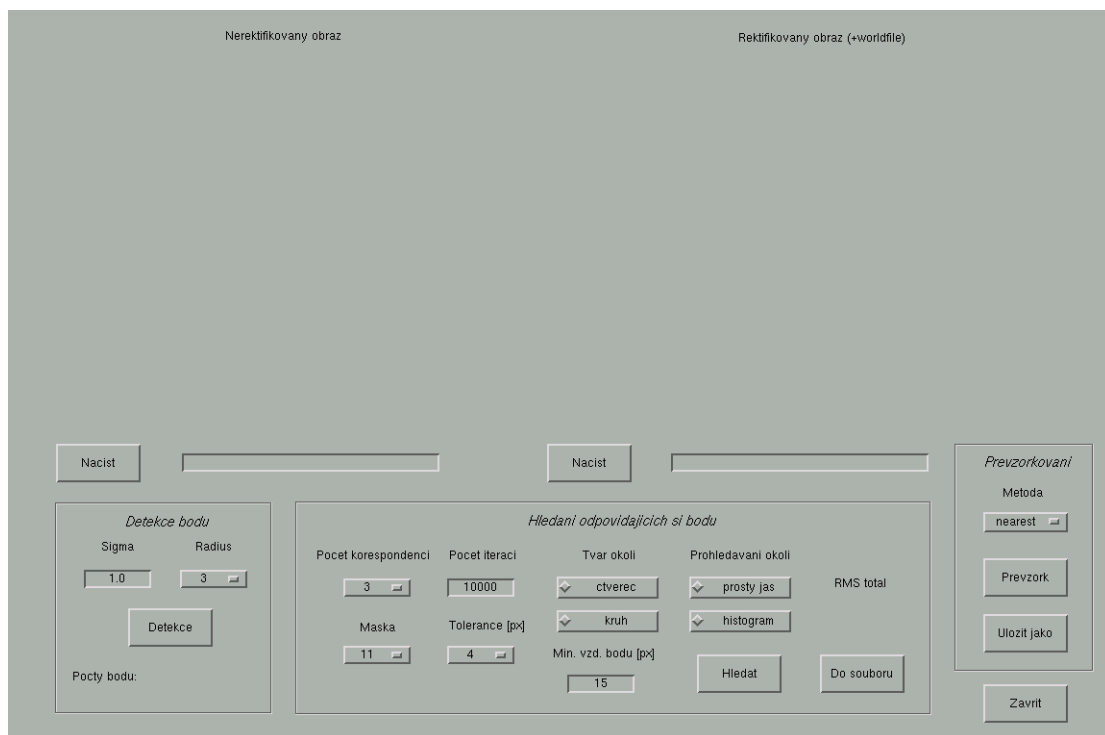
A.2 Ovládání aplikace

Aplikace se spustí na příkazové řádce Matlabu (samozřejmě po nastavení správné cesty) zadáním následujícího příkazu.

```
>> thesisgui
```

Po zadání příkazu se objeví okno aplikace na obrázku (A.1). Pomocí tlačítek **Nacist** se načtou obrazy vstupující do výpočtu. Jsou to, nerektifikovaný obraz (ten chceme transformovat do cílového systému druhého obrazu) a již rektifikovaný obraz. Ten poslední je souřadnicově připojený, tzn. je v daném systému a u něho musí být wordfile. První obraz může být v nějakém souřadnicovém systému, ale tyto informace nebudou využity. V textovém poli hned vedle tlačítek je zobrazen název načteného obrazu. V hranatých závorkách je uvedena velikost obrazu. První číslo je počet řádků a druhé sloupců.

Akceptovatelné formáty pro oba obrazy jsou Tagged Image file (TIF), Portable network graphics (PNG) a Joint Photographic Experts Group (JPG). Odpovídající worldfilly mají extenze *.tfw, *.pfw a *.jfw. Důležité je, že musí být šedotónové. V opačném případě budou na šedotónové obrazy převedeny. Zadávat obrazy s více pásmy než třemi není možné. Na dalším obrázku (A.2) je vidět aplikace s načtenými obrazy.



Obrázek A.1: Okno aplikace po spuštění

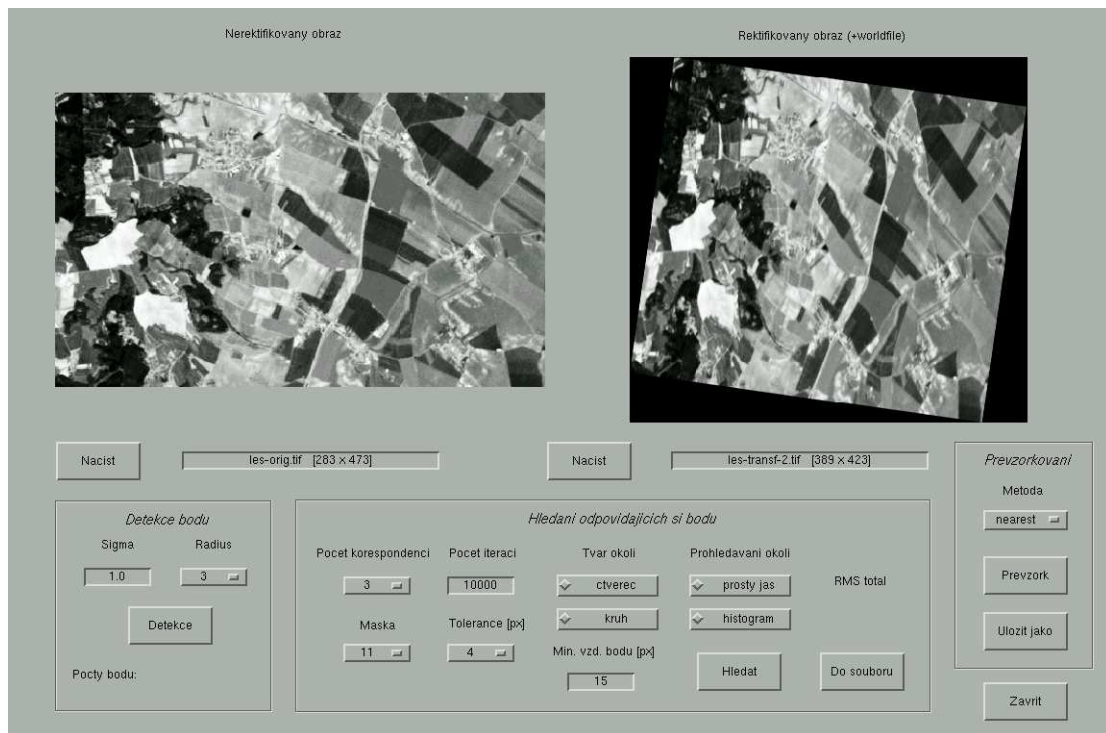
Celé vyhledávání bodů je rozděleno na tři etapy, nepočítáme-li načtení obrazů. První je detekce bodů, druhé je samotné hledání korespondujících bodů a třetí, poslední, je převzorkování. U každé etapy uvedeme parametry ovlivňující každou část.

1. Detekce bodů

- *Sigma* – ovlivňuje rozmístění bodů, při změně se posune poloha detekovaných bodů (vlastnosti vychází z Gaussova filtru a tím ovlivnění okolí), hodnoty musí být větší než 0,3 (jinak hodnota nastavena právě na 0,3)
- *Radius* – má vliv množství detekovaných bodů, hodnoty jsou od 3 do 15, čím větší bude radius, tím méně bude detekovaných bodů a v okolí s hodnotou radius nebude další bod

2. Hledání korespondencí

- *Počet korespondencí* počet nalezených korespondujících bodů
- *Počet iterací* – počet iterací v algoritmu, větší počet iterací zaručí větší pravděpodobnost kvalitnější transformace za cenu delšího výpočtu
- *Maska* – velikost okolí pro jejich porovnávání
- *Tolerance* – velikost okolí, do kterého se vejde při nejlepší transformaci největší počet bodů, větší hodnota dává menší kvalitu transformace, ale větší pravděpodobnost nalezení transformace
- *Tvar okolí* – výběr z možných dvou okolí (čtvercové a kruhové), kruhové okolí má výhodu, že je invariantní vůči otočení



Obrázek A.2: Okno aplikace s načtenými obrazy

- *Prohledávání okolí* – metoda pro prohledávání okolí, prohledávání histogramu je časově náročnější ale na určitých datech efektivnější, prostý jas (porovnaná se vzdálenost mezi vektory jasů) a histogram (porovnáva se vzdálenost mezi histogramy)
- *Min. vzd. bodů* – určuje nejmenší možnou vzdálenost mezi nalezenými body, pokud je zadána hodnota větší než celkově možná vzdálenost v obrazech (nebude možné nalézt body v takové vzdálenosti), tak se tento parametr bude ignorován (budou vybrány body s menší vzdáleností),

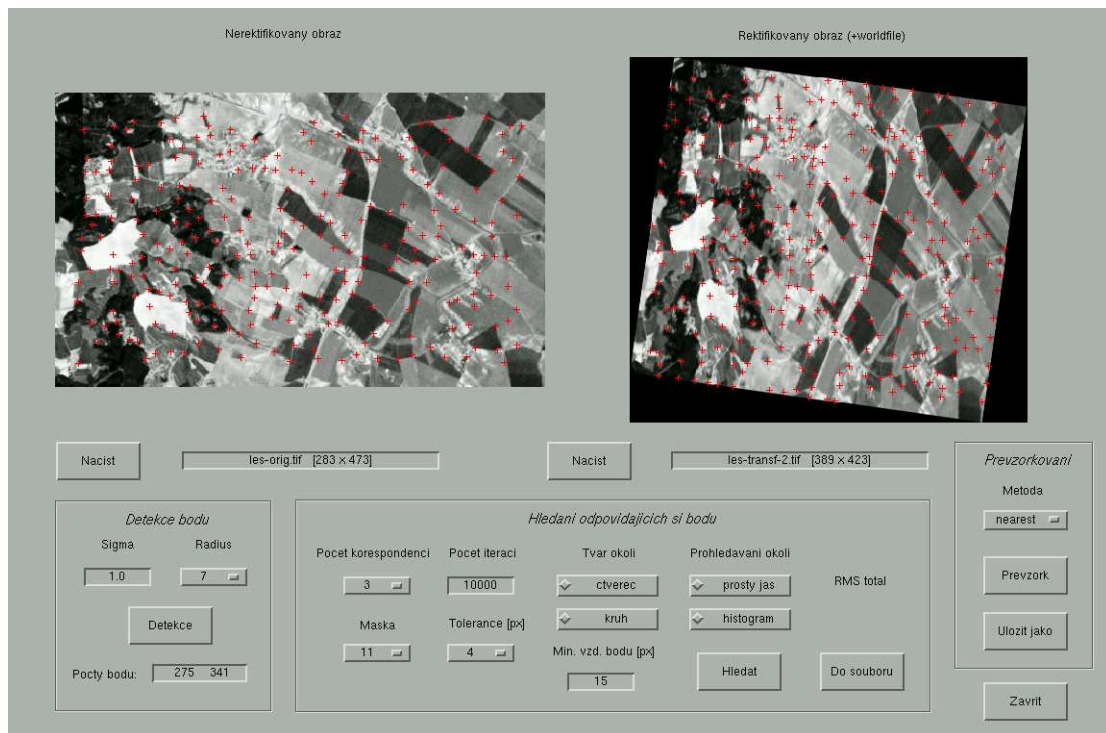
3. Převzorkování

- *Metoda* – určuje metodu převzorkování (metoda nejbližšího souseda, bilineární a bikubická)

Dále si popíšeme činnosti tlačítek. U detekce bodů je to *Detekce*, spustí detekci bodů, ty se zobrazí na obrazech v textovém poli pod tlačítkem se zobrazí počet detekovaných bodů. To je vidět na obrázku (A.3).

Tlačítko *Hledej* spustí hledání korespondujících bodů. Pokud nejsou žádné parametry zadány použijí předem nastavené. U typu okolí a prohledání okolí je to čtverec, resp. prostý jas. Po skončení výpočtu se v textovém poli RMS total zobrazí celková RMS chyba. Pod tímto polem je tlačítko *Do souboru*, to uloží do textového souboru nalezené body z levého, pravého obrazu a z transformované z levého do systému pravého obrazu včetně jednotlivých RMS chyb (pro směry x, y oba směry pro každý bodu a celková chyba, ta která je v textovém poli).

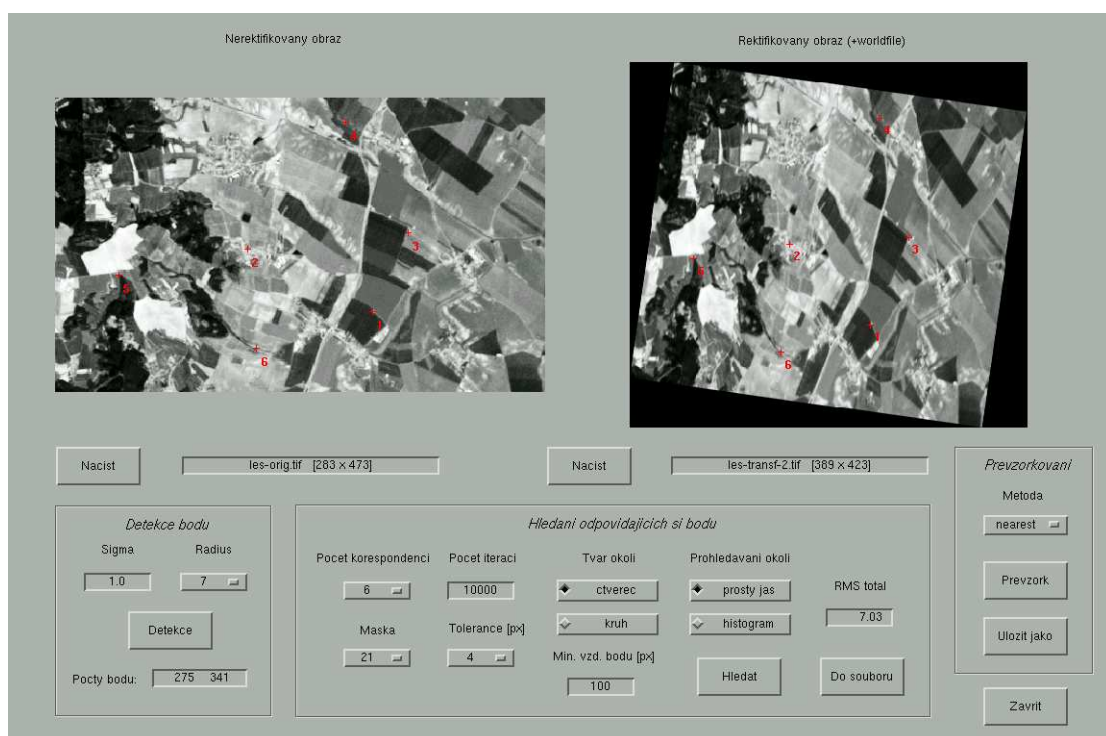
V poslední části Převzorkování je to tlačítko *Převzork*, to podle nalezených korespondencí levý obraz převzorkuje vybranou metodou. Převzorkovaný obraz se zobrazí na zvláštní okno. Tlačítko *Uložit jako* uloží převzorkovaný obraz do samostatného souboru, včetně worldfilu. Možné formáty jsou stejné jako u načtení obrazů, tedy TIF, PNG a JPG. A nakonec poslední tlačítko *Zavřít* zavře okno celé aplikace. Na obrázku (A.4) je okno aplikace s nalezenými korespondencemi.



Obrázek A.3: Okno aplikace s detekovanými body

Pokud se hledání bodů nezdaří a RMS chyba bude větší než tolerance násobená velikostí obrazového bodu, tak se objeví chybová hláška, která nedoporučuje převzorkování. V případě, že uživatel dá obraz převzorkovat, tak obraz bude deformovaný. Velikost deformace závisí na RMS chybě. Na obrazech bude vidět, že body si neodpovídají.

Vzhledem k tomu, že navržený algoritmus dává při stejných datech různé výsledky, tak je nutné v případě neúspěchu hledání korespondencí zopakovat, a to někdy vícekrát. Na nějakých datech, které jsou dost odlišné (např. stejné území časově od sebe vzdálené), nebude možné korespondence nalézt. Ale obecně platí, jsou-li data z jednoho senzoru a přibližně stejného rozlišení, že algoritmus by neměl selhat.



Obrázek A.4: Aplikace s nalezenými korespondencemi

Dodatek B

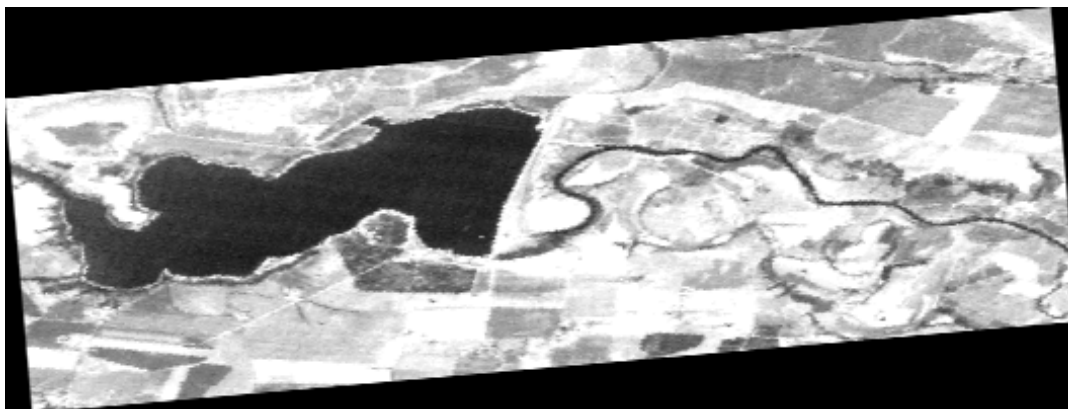
Obrazová příloha s dalšími výsledky

V této části přílohy uvedeme pouze obrazové výsledky z hledání korespondencí, bez komentáře a bez parametrů, pouze s informacemi odkud obrazy pochází. Obrázky budou ve stejné pořadí jako v kapitole o hodnocení algoritmu, a to nejdříve vstupní obrazy (nerektifikovaný a rektifikovaný), poté detekované body na obou obrazech, pak nalezené korespondence taktéž na obou obrazech a nakonec převzorkovaný obraz.

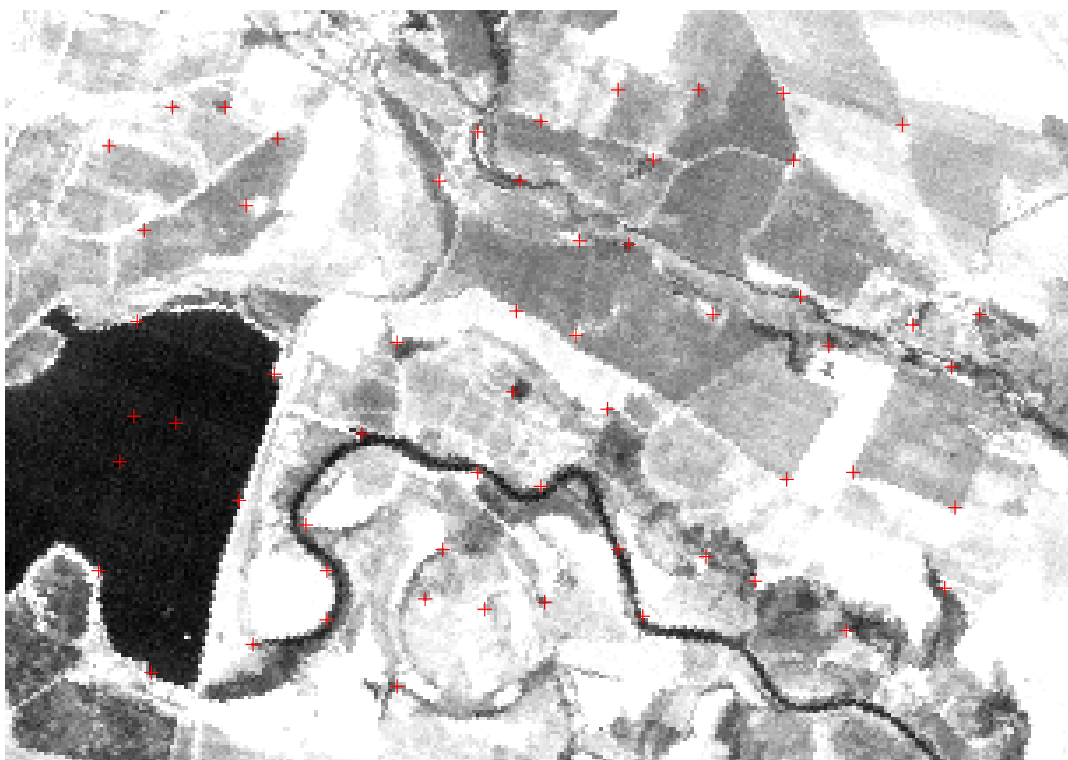
Data: družice Landsat 5 (TM) - obrázky (B.1), (B.2), (B.3), (B.4), (B.5), (B.6), (B.7)



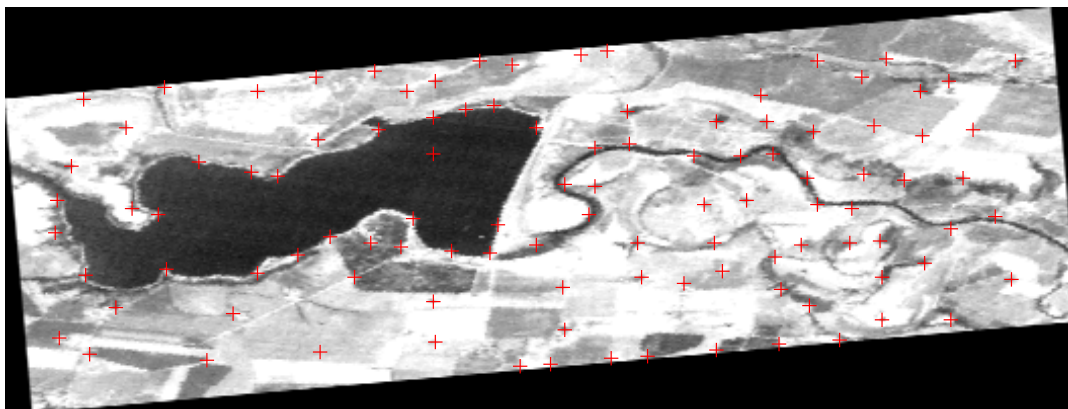
Obrázek B.1: Původní nerektifikovaný snímek



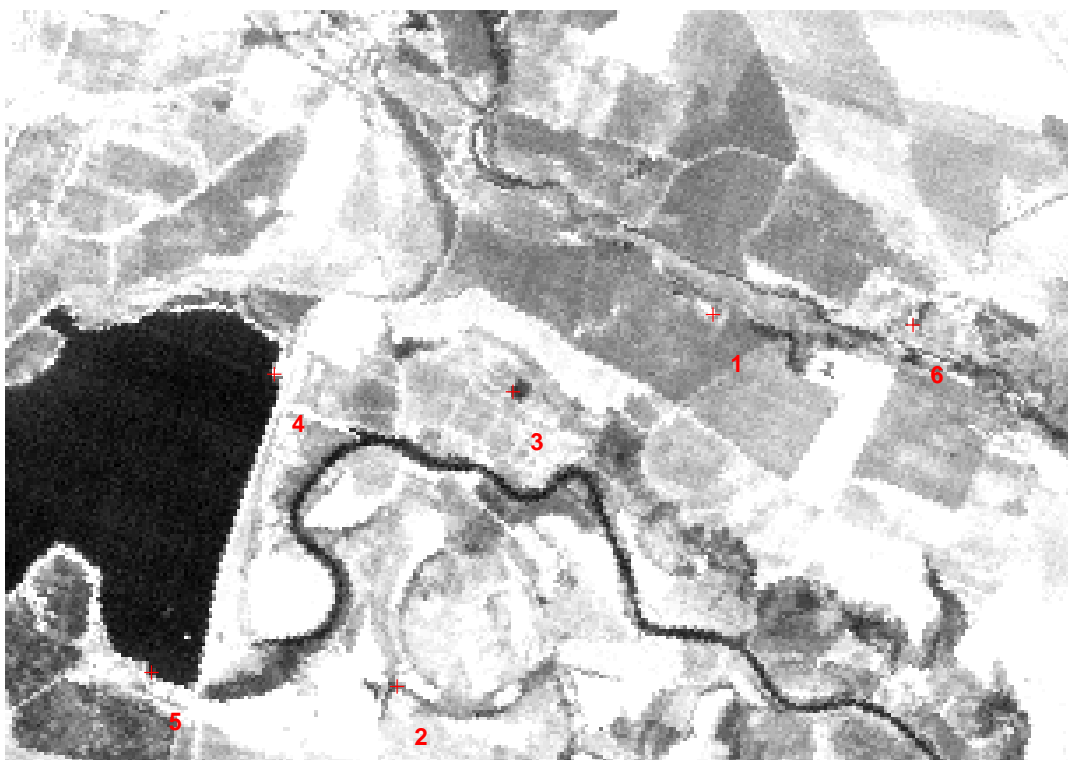
Obrázek B.2: Původní rektifikovaný snímek



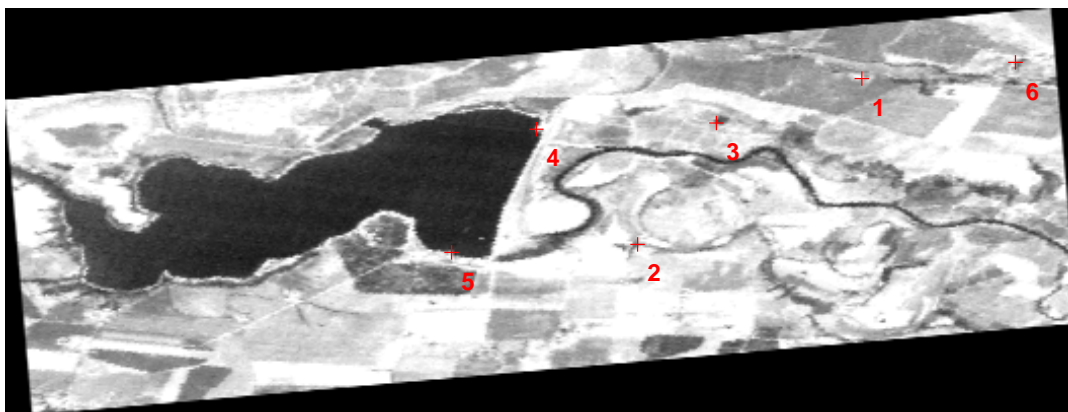
Obrázek B.3: Detekované body na nerektifikovaném snímku



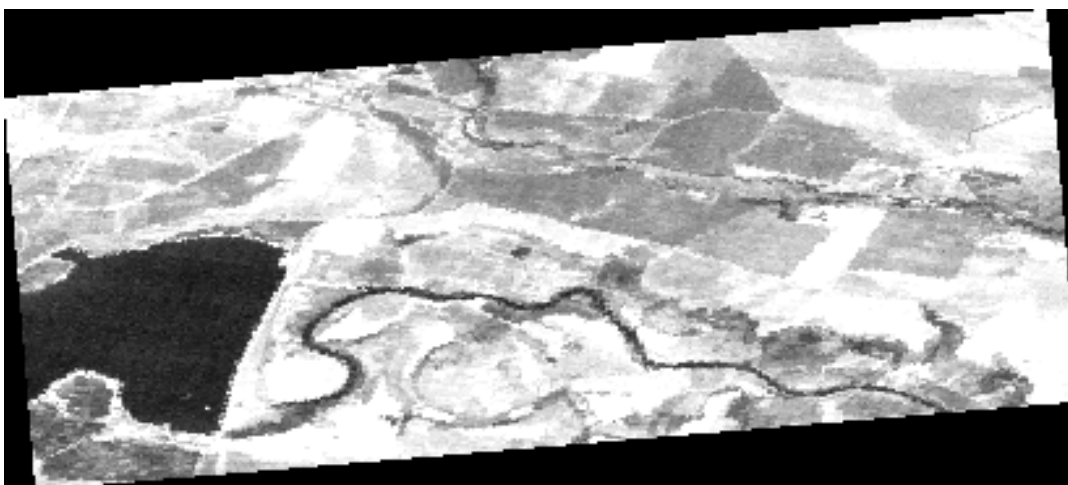
Obrázek B.4: Detekované body na rektifikovaném snímku



Obrázek B.5: Nalezené korespondence na nerektifikovaném snímku



Obrázek B.6: Nalezené korespondence na rektifikovaném snímku



Obrázek B.7: Převzorkovaný obraz v cílovém systému

Dodatek C

Struktura přiloženého CD

Obsahem přiloženého CD jsou zdrojové kódy (skripty) Matlabu pro automatickou rektifikaci družicových dat, testovací data (obrazy), obrazová příloha a tato práce ve formátech pdf a ps.

```
2016 ./aplikace/data-rect
976  ./aplikace/data-unrect
3206 ./aplikace
28770 ./dp-text
17328 ./reference
877  ./priloha/mss-zatec
624  ./priloha/spot-lp
605  ./priloha/spot-les
446  ./priloha/tm-nechranice
2557 ./priloha
93   ./zadani-dp
51957 ./
```

Dodatek D

Evidenční list

Souhlasím s tím, aby moje diplomová/bakalářská práce byla půjčována k prezenčnímu studiu v Univerzitní knihovně ZČU v Plzni.

Datum:

Podpis:

Uživatel stvrzuje svým čitelným podpisem, že tuto diplomovou/bakalářskou práci použil ke studijním účelům a prohlašuje, že ji uvede mezi použitými prameny.

Jméno	Fakulta/katedra	Datum	Podpis