

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE VÝZNAČNÝCH BODŮ V OBRAZE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. TOMÁŠ DUDA

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE VÝZNAČNÝCH BODŮ V OBRAZE

INTEREST POINT DETECTION IN IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ DUDA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAL ŠPANĚL

BRNO 2009

Abstrakt

Tato diplomová práce se zabývá detekcí význačných bodů v obraze. Primárním cílem je vytvořit aplikaci pro vytváření panoramatických fotografií, která bude na této detekci založena. Aplikace bude využívat detektor SIFT pro nalezení klíčových bodů obrazů. Následně algoritmem RANSAC zjistí geometrickou konzistenci bodů obrazů a podle této geometrické konzistence pak transformuje obrázky tak, aby vytvářeli panorama. Nakonec budou použity techniky pro vyhlazení přechodů mezi fotografiemi.

Abstract

This master's thesis deals with the interest point detection in images. The main goal is to create an application for making panoramic photos, which is based on this detection. The application uses the SIFT detector for finding keypoints of the image. Afterwards the geometrical consistence of image points using the RANSAC algorithm is finds out and images are transform into panorama in accordance with this geometrical consistence. Finally techniques for blending of transition between photos are used.

Klíčová slova

Detekce významných bodů, SIFT, hledání korespondence, RANSAC, panoramatická fotografie, skládání snímků

Keywords

Interest point detection in images, SIFT, finding correspondence, RANSAC, panoramic photos, image stitching

Citace

Duda Tomáš: Detekce význačných bodů v obraze, diplomová práce, Brno, FIT VUT v Brně, 2009

Detekce význačných bodů v obraze

Prohlášení

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

.....
Tomáš Duda
26. 5. 2009

Poděkování

Tímto bych chtěl poděkovat mému vedoucímu diplomové práce Ing. Michalu Španělovi za cenné rady a příkladné vedení při vypracovávání této studie. Dále bych chtěl poděkovat svým rodičům a přítelkyni za nekonečnou podporu nejenom při vypracovávání diplomové práce, ale po celou dobu studia.

© Tomáš Duda, 2009

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

Obsah

Obsah	1
1 Úvod.....	2
2 Detekce bodu v obraze.....	3
2.1 Detekce rohu v obraze	3
2.2 Detekce významných bodů.....	5
3 SIFT	6
3.1 Popis metody	6
3.2 Sestrojení scale-space	6
3.3 Lokalizace významných bodů	8
3.4 Odstranění bodů na hranách	8
3.5 Přiřazování orientací.....	9
3.6 Tvorba deskriptorů.....	9
4 RANSAC	11
4.1 Určení modelu	11
4.2 Popis metody	12
4.3 Výhody a nevýhody.....	13
4.4 RANSAC na korespondenčním problému.....	13
5 Panoramatická fotografie	15
6 Návrh řešení	19
6.1 Rozbor navrženého řešení.....	19
6.2 Detekce významných bodů.....	20
6.3 Vyhledání korespondence.....	21
6.4 Transformace obrázků a jejich vykreslení	24
6.5 Vyhlazení přechodů.....	27
6.6 Srovnání úrovní jasu.....	28
7 Implementace	29
8 Testování a zhodnocení výsledků	30
8.1 Interpolace transformovaného obrazu	30
8.2 Vyhlazení přechodu.....	30
8.3 Průměrování na hraně obrázku	31
8.4 Srovnání úrovní jasu	32
8.5 Testování	33
9 Závěr	37

1 Úvod

V dnešním světě techniky je oblast počítačového vidění velmi rozsáhlá oblast, která si zaslouží velkou pozornost. Tato věda se zabývá prací s obrazem a zjištění co nejvíce informací o tomto obraze. Důležitým úkonem ve zjišťování informací o obraze je detekce určitých prvků (bodů, křivek, útvarů). Díky těmto detekcím je pak možné v obraze rozpoznat např. SPZ značku auta stojícího na parkovišti, vadný výrobek na běžícím páse továrny nebo počet lidí čekající ve frontě v bance.

Významnou technikou při zpracování obrazu je detekce významných bodů v obraze. Od samého počátku vzniku počítačového vidění se používalo mnoho technik na zjištění významných bodů. Některé metody se zaměřily spíše na detekci rohů, některé spíše na detekci větších oblastí, některé vyloženě na detekci bodů jako takových. Použití detekce významných bodů má široké použití. Používá se např. k vyhledání známých objektů v obraze, 3D rekonstrukci objektu nebo spojování obrazů na základě shodných bodů.

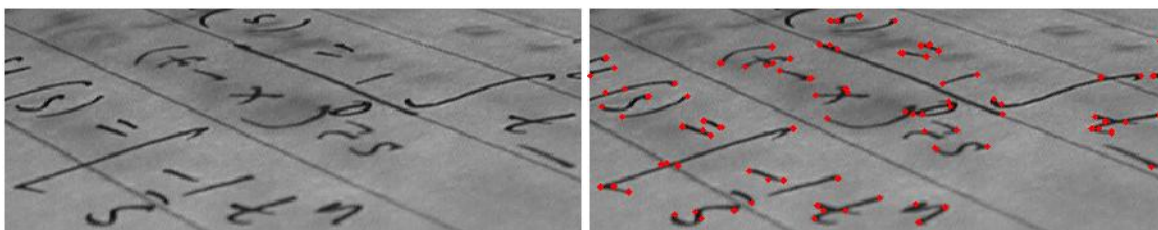
Tato technická zpráva rozebírá nejdůležitější algoritmy s různým přístupem k detekci. Nejvíce je však zaměřena na metodu SIFT, která detekuje významné body jako takové. Výsledkem diplomové práce měla být aplikace, která využívá detekci významných bodů v praxi. Jelikož rád fotografii a aplikování počítačového vidění do běžného života (čili i mého) mi přijde velice zajímavé a praktické, vybral jsem si proto aplikaci pro spojování snímků a následného vytváření tzv. panoramatických fotografií. V této technické zprávě je vysvětleno „co to je panoramatická fotografie“, jsou probrány techniky, které souvisí s vytvářením panoramatických snímků, a je rozebrán návrh a implementace aplikace pro tvorbu těchto panoramat.

Struktura technické zprávy je rozdělena do 9 částí. Algoritmy detekující významné body jsou probírány v kapitole 2, algoritmus SIFT pak v kapitole 3. V kapitole 4 je rozebrán algoritmus RANSAC, který bude sloužit k vyhledání korespondence významných bodů samotných fotografií. Kapitola 5 rozebírá pojem „panoramatická fotografie“, jaké druhy těchto fotografií existují a k čemu se používají. Návrh řešení, jakých technik využiji při tvorbě aplikace pro výrobu panoramatických fotografií, je řešen v kapitole 6. V kapitole 7 pak popis samotné implementace aplikace. Výsledky, jakých se výsledná aplikace dopracovala a jejich zhodnocení, jsou popsány v kapitole 8. V poslední kapitole nazvané Závěr jsou popsány možné optimalizace a vylepšení aplikace a je zhodnocen celkový přínos diplomové práce.

2 Detekce bodu v obraze

Detekce bodu v obraze je pro počítačové vidění jedním z nejdůležitějších oborů. Díky nalezeným bodům ve scéně může počítač určit celou řadu důležitých vlastností objektu (velikost, tvar, polohu, atd.) a podle nich je schopen určit, jaký objekt se ve scéně vlastně nachází. Proto je detekce těchto bodů velmi důležitá. Počítač musí pomocí algoritmů rozeznat, které body jsou ty významné. Tyto body musí splňovat určité vlastnosti:

- musí být jasně matematicky popsány,
- musí mít jasnou definici v prostoru,
- okolí tohoto bodu musí mít bohatou škálu vlastností pro pozdější jednoduché zpracování vizuálním systémem,
- je stabilní vůči lokálním a globálním deformacím obrazu, změně perspektivy (afinním transformací obrazu, změny měřítka a rotace) a změny osvětlení/jasu tak, aby byl s vysokým stupněm opakovatelnosti opět nalezen.



Obrázek 2.1: Detekce významných bodů z obrázku nalevo – výsledek je na obrázku vpravo

Pojem významného bodu sahá až ke staršímu pojmu detekce rohu, kdy rohové rysy byly zjištěny ve studii s cílem získání robustní, stabilní a dobře popsané funkce pro sledování a rozpoznávání objektů ve 3D prostoru z obrázků v 2D prostoru.

2.1 Detekce rohu v obraze

Rohy, které snímají detektory rohu lze chápat dvěma způsoby – buď jako průsečík dvou linií (hran) nebo jako lokální oblasti, které mají vysoký stupeň změny jasu ve všech směrech. V praxi detektory nejsou citlivé přímo na rohy, ale právě na tyto lokální oblasti.

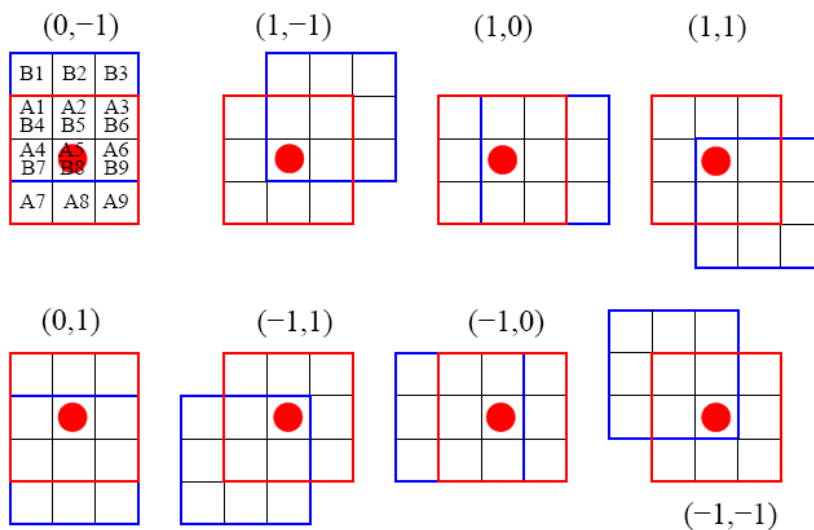
Detektory rohů v obraze vznikaly kvůli potřebám řešení konkrétních projektů, proto mají detektory různé vlastnosti a různé výsledky. Ty nejdůležitější ve významu rozvoje detekce významných bodů popíši níže.

Moravcův detektor

Tento algoritmus napsal Hans P. Moravec v roce 1977 pro řízení vozíku snímaného pomocí dvou kamer. Tato metoda je založena na hledání oblastí, které jsou lokálním maximem ve vypočtených směrových změnách intenzity zkoumaného obrázku. Tyto oblasti jsou reprezentovány okénky stejných velikostí a jedním z nich je pak následně pohybováno. Tento pohyb je proveden přičtením konstant u a v k souřadnicím oblasti, jak je ukázáno na obrázku 2.2. Mezi těmito oblastmi je pak spočítán součet čtverců rozdílů (SSD) podle vzorce:

$$E(u, v) = \sum_{(u,v) \in \{-1,0,1\}} (I(x, y) - I(x + u, y + v))^2 \quad (2.1)$$

kde $I(x,y)$ jsou hodnoty pixelů obrázku. Z těchto 8 hodnot $E(u,v)$ je vybrána nejmenší hodnota. Následně se hledá v lokální oblasti s nejmenší E hodnotou lokální maximum. Pokud toto lokální maximum je větší než zvolená prahová hodnota, je tento bod označen jako významný roh.



Obrázek 2.2: Pohyb oblasti pro výpočet SSD [5]

Jedinou výhodou tohoto detektoru je jeho jednoduchost a rychlost, jinak převažují nevýhody. Sám Moravec varoval, že detektor není izotropní, reaguje na hrany a šum v obraze.

Harrisův detektor

Tento text byl inspirován z [5].

Detektor vznikl v roce 1988 Chrisem Harrisem a Mikem Stephensem z nutnosti získat více informací z obrázků potřebných k rozpoznání objektů a ploch. Proto se zaměřili na sledování rohů i hran. Tento algoritmus je vylepšením Moravcova detektoru, ze kterého vycházeli. Je v něm ovšem pár významných změn:

- Pro sledování lokální oblasti je použito kruhové okénko oproti Moravcova čtverce. To zapříčiňuje nezávislost na rotaci. Je zde také použita Gaussova funkce pro vyvážení vnitřních hodnot. Díky ní bude bodům vzdálenějším od středu okénka přiřazován menší význam a tím se omezí vliv šumu.
- Dalším vylepšením je odstranění anizotropních odezví z Moravcova operátoru. K odstranění tohoto nedostatku je potřeba funkce, která je schopna měřit změnu intenzity ve všech směrech okolí bodu. Pro tento případ je použita autokorelační funkce.

Autokorelační funkce zahrnuje autokorelační matici, která určuje hlavní křivost plochy v okolí zkoumaného bodu. Toho se využívá při rozhodování, jestli je bod významný a případně jestli je to roh nebo hrana.

Pro hledání významných bodů se opět jako u Moravcova detektoru hledají lokální maxima z nalezených rohů. Pro generování těchto bodů se použije determinant a trasování matice M :

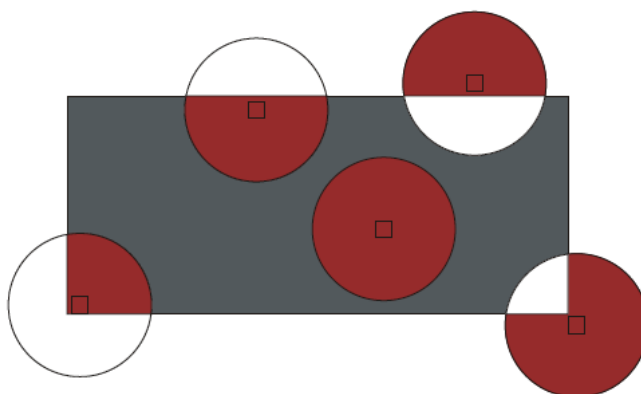
$$R(x, y) = \det(M) - k \cdot \text{trace}(M)^2 \quad (2.2)$$

Díky autokorelaci má Harrisův detektor lepší výsledky než Moravcův. Je invariantní vůči šumu, rotaci a posunu. Není však invariantní vůči měřítku a proto při hledání významných bodů reálných scén není použitelný.

SUSAN detektor

Text byl opět inspirován z [6].

SUSAN je zkratka pro Smallest Univalued Segment Assimilating Nucleus. Tento algoritmus předpokládá, že uvnitř kruhové masky mají všechny pixely objektu stejné úroveň jasu. Kontroluje tedy všechny body v této kruhové oblasti a porovnává je se středovým bodem (jádro). Body, které mají stejnou úroveň jasu, algoritmus označí jako body USAN (Univalued Segment Assimilating Nucleus). Rohové body se vyhledávají tak, že tato kruhová maska se aplikuje na každý bod obrazu a následně se vyhledávají lokální minima.



Obrázek 2.3: SUSAN algoritmus – červená místa představují (USAN oblasti), body se stejnou hodnotou úrovně jasu jako střed [6]

2.2 Detekce významných bodů

Dosud zmiňované detektory položily základy pro detektory významných bodů. Vycházejí z jejich poznatků a experimentů a jsou vylepšením dosud diskutovaných detektorů rohů v oblasti rychlosti, stability a potlačení šumu. V této oblasti se nejvíce objevují dva detektory – SURF a SIFT.

SURF (Speeded Up Robust Features) byl poprvé publikován v roce 2006 kolektivem autorů na konferenci v rakouském Grazu [7]. Inspirací autorům byla původem starší metoda SIFT (viz kapitola 3). Pro generování scale-space prostoru (viz kapitola 3.2) použili determinant Hessiánské matice. Tím odfiltrovali body, které měly velkou odezvu v okolí hran v metodě SIFT. Pro popis bodů použili SIFT deskriptory a významně je zjednodušili na 64 binů. Díky těmto a dalším změnám je SURF rychlejší než SIFT a je použitelný v real-time aplikacích detekce bodů (např. ve videu).

Metoda SIFT je detailně popsána v následující kapitole 3.

Detekce významných bodů se používá zejména při rozpoznávání objektů v obraze, 3D rekonstrukci objektů z 2D obrazu, sledování pohybu ve videu, segmentace, robotickém vidění a skládání panoramatických obrazů. Právě skládání panoramat, jak se čtenář může dále dočíst, je hlavním předmětem této práce.

3 SIFT

Algoritmus SIFT vytvořil v roce 2004 americký profesor David G. Lowe. Tato metoda byla navržena k extrakci významných bodů z obrázků a následného vyhledávání společných rysů.

Zkratka SIFT - Scale Invariant Feature Transform znamená v překladu „měřítkově nezávislá transformace rysů obrazu“. V praxi to znamená, že transformuje obrazová data do měřítkově nezávislých souřadnic a nalezené významné body jsou invariantní vůči měřítku, rotaci a posunu. Metoda se vyznačuje i odolností vůči afinním deformacím, šumu a změn v osvětlení ve vstupním obraze.

V nynější době je SIFT nahrazován algoritmem SURF, který pracuje s Fast-Hessian maticí a je rychlejší. Toto se děje hlavně při detekci v reálném čase (např. ve videu), kdy rychlost hraje důležitou roli. SURF ovšem nemá tak kvalitní výsledky jako SIFT a při výrobě panoramat je kvalitní detekce důležitějším hlediskem než rychlost, čili v tomto oboru je lepší než SURF.

Následující kapitoly popisující metodu SIFT jsou inspirovány z [3] a částečně z [5].

3.1 Popis metody

Proces získávání významných bodů z analyzovaného obrázku lze rozdělit do pěti fází.

1. Sestrojení scale-space a vyhledání lokálních extrémů

Jde o vytvoření měřítkově nezávislé reprezentace obrazu, tzv. scale-space (viz kapitola 3.2). Zde se vyhledávají přes všechna měřítká lokální extrémy. Pro rychlejší generování scale-space je použita místo Laplacianu Gaussovy funkce (LoG) jeho aproximace pomocí rozdílu Gaussových funkcí (DoG).

2. Lokalizace významných bodů

Body s nedostatečnou stabilitou se odstraní a u zbylých se pomocí interpolace upřesní jejich poloha v prostoru.

3. Odstranění bodů na hranách

Body ležící na hranách objektů jsou pomocí Hessianovy matice odstraněny.

4. Přiřazování orientací

Přiřazení dominantní orientace každému významnému bodu podle orientace gradientů v okolí bodu. Nutné kvůli invarianci vůči rotaci.

5. Tvorba deskriptorů

Pro každý významný bod je vypočten deskriptor podle orientace gradientů v okolí bodu. Pomocí deskriptoru je zjišťována korespondence mezi jednotlivými snímky.

3.2 Sestrojení scale-space

Scale-space je v podstatě reprezentace obrazových dat nezávislá na měřítku. Pro vytvoření scale-space musíme detekovat významné body pomocí přístupu filtrování kaskády používající efektivní algoritmus k identifikaci kandidátních lokalit, které jsou následně blíže zkoumány. První etapa detekce významných bodů je zjištění souřadnic bodů a jejich měřítká, které mohou být opakovaně přiděleny při různých pohledech. Nalezení souřadnic bodů invariantních ke změně pohledu může být provedeno tak, že se hledají stabilní body ve všech možných měřítkách použitím funkce scale-space (Witkin, 1983).

Studiemi Koenderinka (1984) a Linderberga (1994) bylo prokázáno, že jediné možné scale-space jádro je Gaussova funkce. Proto scale-space obrázku je definován jako funkce $L(x, y, \sigma)$ -

Laplacian Gaussovy funkce (LoG), která je složena z konvoluce Gaussové funkce $G(x, y, \sigma)$ s proměnlivým měřítkem a vstupního obrázku $I(x, y)$:

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

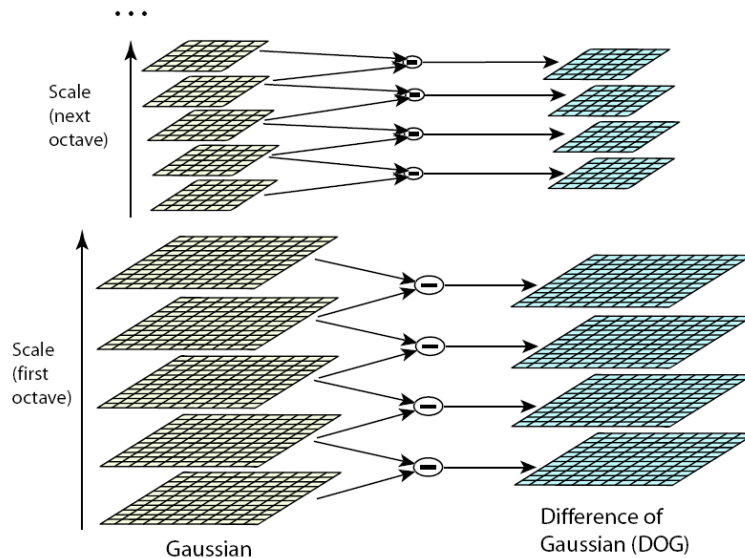
kdy $*$ je konvoluce nad x a y , a

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

Pro efektivní detekování stabilních souřadnic významných bodů v scale-space David Lowe [4] navrhl funkci používající rozdíly Gaussových funkcí v konvoluci se vstupním obrázkem $D(x, y, \sigma)$ - Difference of Gaussian (DoG). Tato funkce se vypočítá z rozdílu dvou sousedních měřítek obrázků oddělené konstantou k :

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

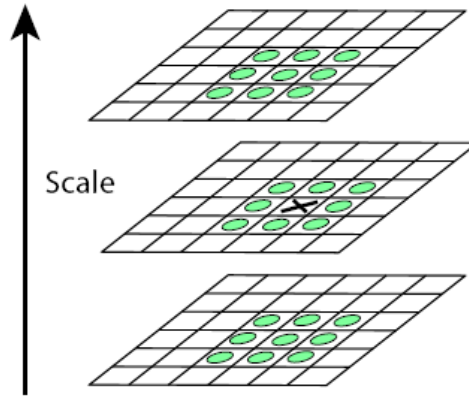
Existuje celá řada důvodů, proč zvolil zrovna tuhle funkci. Jeden z hlavních důvodů je rychlost a jednoduchost na výpočet. Po výpočtu Laplacianu Gaussovy funkce, která musí být vypočítána pro každé měřítko bodů, se funkce D vypočítá pouhým rozdílem jednoduchých obrázků.



Obrázek 3.1: Výpočet DoG pomocí rozdílu Gaussových funkcí [3]

Nyní je důležité vhodně zvolit konstantu k neboli vzorkovací frekvenci, jinými slovy kolik vrstev se musí generovat v každé oktávě. Zvolení této konstanty bude mít vliv na počet nalezených významných bodů. Pokud bude zvolen velký vzorkovací krok (vysoké číslo pro k) nemusí být v obrázku nalezeny všechny požadované významné body. Pokud však zvolíme příliš malý vzorkovací krok, bude se generovat velké množství měřítek v oktávě a vypočtení jednotlivých oktáv může být příliš časově náročné. Ve své studii [3] David Lowe došel k závěru, že optimální počet měřítek na oktávu je 3. Pokud je počet měřítek větší, naleznou se více významných bodů. Ty však, ale nejsou tolik stabilní vůči světelným a geometrickým změnám. Výsledek tímto krokem tedy nezlepšíme.

Po vygenerování všech oktáv můžeme začít počítat významné body. Tyto body leží v lokálních extrémech scale-space obrázku. Hledání se provádí tak, že se porovná sousedních 8 bodů hledaného



Obrázek 3.2: Sousední body testovaného významného bodu [3]

bodu v aktuálním měřítku, 9 bodů většího měřítka a 9 bodů menšího měřítka. Pokud je tento bod větší nebo menší než všech 26 sousedů je označen jako významný. Tento postup se provede pro všechna měřítka, samozřejmě kromě krajních, kdy chybí předchozí nebo následující měřítka. Po ukončení hledání se nalezené významné body podrobí detailnějšímu zkoumání.

3.3 Lokalizace významných bodů

Jakmile byl nalezen významný bod porovnáním s jeho sousedy, musíme bod přesněji lokalizovat. Lowe a Brown ve své studii (Brown a Lowe, 2002) zjistili, že efektivnější je nepoužívat celočíselné souřadnice. Brown vyvinul metodu pro přesnější lokalizaci extrému ve scale-space, kde používá aproximaci funkce DoG (viz kapitola 3.2). Tato aproximace se provádí Taylorovým rozvojem:

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

kde $D(x)$ je aproximace funkce DoG Taylorovým rozvojem a $x = (x, y, \sigma)^T$ je souřadnice ve scale-space. Hodnota $D(x)$ a jeho derivace je vypočítána v místě zkoumaného bodu. Extrémy tohoto Taylorova rozvoje jsou tzv. subpixelové a subměřítkové pozice. Extrémy se získají derivováním funkce podle x a položením rovno nule. Po vyjádření je subpixelová a subměřítková pozice dána posunutím \hat{x} od daného zkoumaného bodu:

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

Pokud je posun \hat{x} větší než 0.5, tak to znamená, že významný bod leží blíže jiného bodu než bodu zkoumaného. Pozice významného bodu je tedy vyměněna za pozici bližšího bodu a aproximace je provedena pro nový bod. Nakonec po nalezení správné pozice a správného posunu, je posun přičten k významnému bodu. Tím vznikne subpixelová a subměřítková souřadnice významného bodu.

3.4 Odstranění bodů na hranách

Pro detekci významných bodů je určitě nežádoucí detekovat body, které leží na hranách, neboť úkolem těchto detektorů je hledat spíše rohy objektů. Navíc tyto body jsou náchylnější na šum. Hledání bodů pomocí scale-space zaručuje nalezení bodů i na hranách. Proto je zapotřebí tyto body najít a odstranit.

Body ležící na hranách mají velkou hlavní křivost kolmo na hranu a malou podél hrany. Hlavní křivost bodu se vypočítá z vlastních čísel Hessianovy matice 2x2. Matici budeme značit \mathbf{H} .

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (3.6)$$

Při odstranění bodů se použije podobný princip jako u Harrisova detektoru, kde se okolí bodu popsalo pomocí matice. Zde se na rozdíl od Harrisova detektoru používá druhá derivace, místo první. Počítání vlastních čísel z Hessiány matice je výpočetně náročné, proto je vhodné zavést jednodušší způsob výpočtu. Pokud bod leží na hraně, jedno z jeho vlastních čísel bude výrazně větší než druhé. Proto stačí znát poměr mezi těmito čísly. Nechť číslo α je větší než číslo β . Pak můžeme vypočítat součet těchto čísel na hlavní diagonále \mathbf{H} a determinant matice \mathbf{H} :

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.7)$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.8)$$

Pokud je determinant záporný, zakřivení má jiný směr, čili bod není extrémem a my ho můžeme vyškrtnout. Nechť r je znázornění poměru mezi těmito dvěma čísly $\alpha = r\beta$, pak:

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad (3.9)$$

z toho vyplývá, že podíl $\text{Tr}(\mathbf{H})^2$ a $\text{Det}(\mathbf{H})$ závisí pouze na poměru čísel α a β . Pokud je tedy splněna nerovnice:

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \quad (3.10)$$

bod je zachován. Lowe ve své studii [3] používá $r = 10$.

3.5 Přiřazování orientací

Přiřazení orientace každému významnému bodu vychází z lokálních vlastností obrazu. Z těchto orientací vychází i výsledné deskriptory významných bodů, proto přiřazení orientace závisí na invarianci vůči rotaci. Nejdříve se vybere podle měřítka Gaussova plocha nejbližší vypočítanému bodu, která byla již dříve vypočítaná při sestavování scale-space, a označíme ji L . Následně se pro každý bod $L(x, y)$ vypočítá velikost gradientu $m(x, y)$ a orientace $\theta(x, y)$:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.11)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (3.12)$$

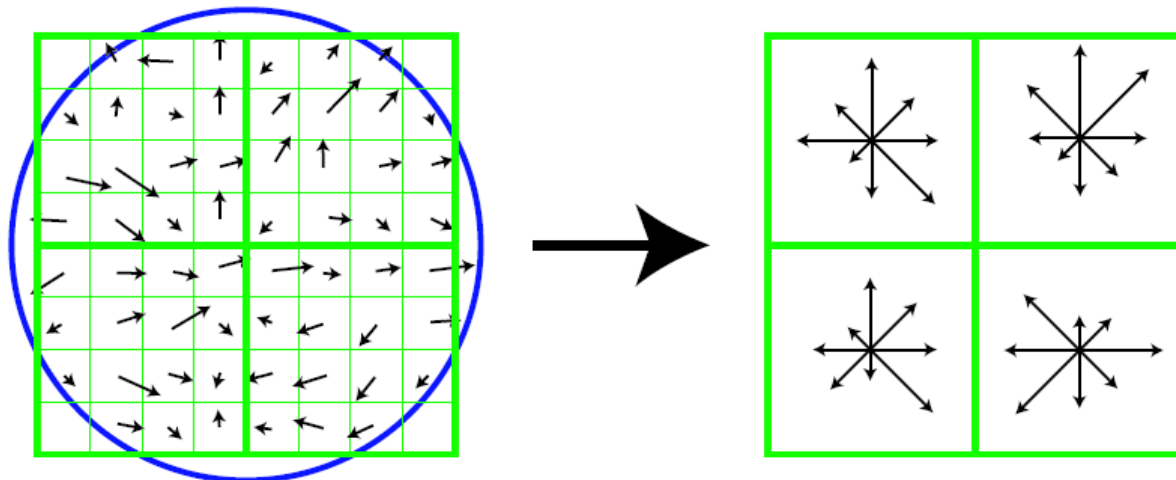
Pro každý významný bod je vytvořen histogram orientací z gradientů vypočtených z okolí. Histogram má 36 binů (směrů) a pokrývá úhel 360 stupňů. Ke každému binu přidanému do histogramu s odpovídající orientací je přičtena velikost gradientu vážená pomocí kruhového okénka s Gaussovským rozložením vah s σ rovným 1.5 násobku měřítka klíčového bodu.

Výsledná orientace je největší orientace z histogramu. Pokud jsou detekovány další orientace, které mají minimálně 80% hodnoty největší orientace, je pro ně vytvořen další významný bod se stejnými souřadnicemi, ale s novou orientací. Experimentálně je dokázáno, že takových bodů v obraze je pouze 15%, avšak významně přispívají k lepší stabilitě při hledání korespondencí.

3.6 Tvorba deskriptorů

V předchozích fázích jsme přiřadili každému významnému bodu pozici, měřítko a orientaci vůči obrázku. Teď je zapotřebí tyto informace jednoznačně uložit, aby se s nimi dalo jednoduše pracovat

a hledat korespondenci mezi nimi. Je tedy důležité, aby informace byly zapsány invariantně vůči drobným geometrickým a jasovým změnám. K takovému zápisu významně přispěli Edelman, Intrator a Poggio, kteří vynalezli systém deskriptorů vycházející z modelu biologického vidění. Tento deskriptor je v podstatě popis okolí významného bodu pomocí orientace gradientů a umožňuje drobné změny v jejich umístění. Tím dosáhneme požadovanou drobnou invarianci vůči geometrickým změnám.



Obrázek 3.3: Vypočtení deskriptorů z orientace gradientů okolí bodů [3]

Pro konstrukci deskriptoru je potřeba znát gradienty obrázku v okolí klíčového bodu. Jeho konstrukce tedy začíná stejně, jako tomu bylo ve fázi, kde se bodům přiřazovaly orientace. Můžeme tedy využít již předpočítané hodnoty a ušetřit tím výkon výpočetního systému. Pro zachování nezávislosti na měřítku se opět vybírá z předpočítaných vrstev podle měřítka klíčového bodu. Pro zachování nezávislosti na rotaci obrázku je využita orientace přiřazená bodu v předchozí fázi. Podle této orientace se natáčí orientace gradientů, s kterými se pracuje tak, aby v případě odlišné orientace korespondujícího významného bodu v jiném obrázku byly výsledné orientace gradientů stejné a tím vedly na stejný deskriptor. Pro účely výpočtu se okolí bodu rozdělí na několik čtvercových oblastí a pro každou se sestrojí histogram orientací (obr. 3.3) stejným způsobem jako tomu bylo ve fázi 3.5. Veškeré informace obsažené v těchto histogramech tvoří data deskriptoru. Pokud se tedy okolí rozdělí na 4 oblasti a histogramy orientací mají po osmi bincích, potom se deskriptor skládá ze $4 \times 8 = 32$ hodnot. Lowe používá 128binové deskriptory, kdy okolí dělí na 4×4 oblasti a histogramy orientace rozdělují mezi 8 binů. Výhoda takovéto reprezentace pomocí hodnot histogramů sestrojených na oblastech o $n \times n$ prvcích spočívá v možnosti malého posunutí gradientu v obrázku bez toho, aby se to nějak projevilo ve výsledném vzhledu histogramu a tím také deskriptoru. Představme si například gradient vypočtený pro bod ležící na levém okraji oblasti. I když se gradient vlivem geometrických deformací posune o n míst doprava, tak stále bude patřit do stejného histogramu a na výsledku se to nijak neprojeví. Nebezpečí je ovšem v posunech (i malých), kde by se gradient dostal do jiné oblasti anebo kdyby se změnila jeho orientace a byl přidán do jiného binu histogramu. Za tímto účelem metoda SIFT používá trilineární interpolaci a rozkládá pomocí ní hodnotu každého gradientu mezi sousední biny histogramu.

4 RANSAC

Text byl převzat a upraven z [2] a inspirován z [13] a [14].

RANSAC je zkratka slov RANdom SAMple Consensus, neboli v doslovném překladu „Shoda náhodného vzorku“. Ve své podstatě je to metoda vybírající náhodnou skupinu dat ze všech vzorků. Její algoritmus byl poprvé publikován Fischlerem a Bollesem v roce 1981 a metoda je určena pro odhad parametrů matematického modelu ze sady sledovaných dat. Metoda je založena na tom, že libovolná obrazová data obsahují chyby (např. šum v obraze).

Základní předpoklad metody spočívá v tom, že naměřená data jsou složena ze vzorků odpovídajících měřenému modelu (inliers) a ze vzorků odpovídajících měřicím chybám (outliers). Obě skupiny vzorků mohou být navíc zatíženy šumem neznámého rozložení a parametrů.

Chyba jednoho vzorku může být způsobena šumem, chybnou detekcí nebo špatnou interpretací dat. Metoda RANSAC automaticky předpokládá část chybných dat a na základě většinou malého souboru vybraných dat určuje správné parametry modelu.

Vlastnosti metody:

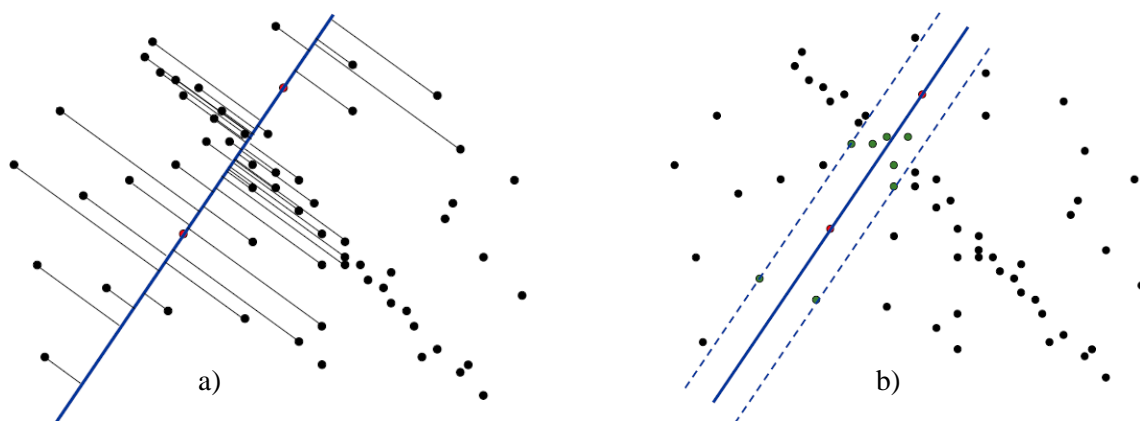
- K dispozici je celkem M vzorků měřených dat.
- Parametry modelu mohou být odhadnuty z $N \subseteq M$ vzorků.
- Pravděpodobnost náhodného výběru skupiny dat, která právě náleží správnému modelu je p_g .
- Pravděpodobnost ukončení výpočtu za předpokladu existence alespoň jednoho vyhovujícího modelu, aniž by byl nalezen jakýkoliv jiný dostačující model je p_{fail} .

4.1 Určení modelu

K vypočtení výsledného modelu můžeme dojít několika způsoby. Některé z nich si ukážeme na příkladu hledání přímky.

Jedním ze způsobů, jak určit model, je vypočítat vzdálenost všech bodů od zvolené přímky. Obrázek 4.1a ukazuje situaci při takovémto vybírání modelu. Červené body znázorňují vybrané náhodné body. Těmito body je proložena přímka a je zjišťována suma vzdáleností všech bodů od této přímky. Model s nejmenší sumou vzdáleností je hledaný nejlepší model.

Častěji používaným způsobem vybírání modelu je určení hranice, která bude oddělovat inlier body, čili body zapadající do modelu, od outlier bodů. Situaci lze vidět na obrázku 4.1b. Červené body opět znázorňují náhodně vybrané body. Opět je jimi proložena přímka. Pro vybrání inlier bodů se určí maximální vzdálenost od přímky, do které mohou body ležet, aby byly vybrány do modelu. Zbývající body jsou outlier body. Nejlepší model je pak ten, který má ve „své oblasti“ nejvíce bodů.



Obrázek 4.1: Vybrání modelu a) metodou sumy vzdáleností, b) metodou určení hranice [14]

4.2 Popis metody

Metoda se dá popsat jednoduchým obecným algoritmem. Nejdříve definujeme vstupy a výstupy.

Formální vstupy:

data – sada pozorovaných vzorků

model – model, který může odpovídat vzorkům

N – minimální počet vzorků potřebných pro určení modelu

L – maximální počet povolených iterací

T – hodnota prahu určující přesnost dat s modelem

K – množina inlier bodů – bodů zapadajících do modelu

D – počet vzorků potřebných ke konstatování, že model dobře odpovídá datům

Formální výstupy:

Ω – parametry modelu, který nejlépe vyhovuje datům popř. nula, pokud není nalezen žádný vhodný model

Postup výpočtu:

1. Náhodný výběr N vzorků.
2. Výpočet parametrů modelu.
3. Stanovení počtu K vzorků z M , jež vyhovují vypočtenému modelu s parametry a s předem stanovenou tolerancí T .
4. Pokud je počet vyhovujících vzorků K dostatečně vysoký (je splněna podmínka $K > D$), je přijat aktuálně vypočtený model a výpočet je úspěšně ukončen.
5. Opakování kroků jedna až čtyři L -krát a poté ukončení výpočtu neúspěšně.

Vstupem pro algoritmus je tedy sada sledovaných dat, parametrizovaný model a známé důvěryhodné parametry. Algoritmus dosáhne svého cíle iterativním vybíráním náhodné podskupiny z celé skupiny vstupních dat pro verifikaci modelu. Tyto vzorky mohou být hypoteticky bezchybné a tato hypotéza se testuje. Model je nastaven na hypoteticky bezchybná data, čili všechny volitelné parametry jsou rekonstruovány podle vybraných vzorků. Všechny další vzorky jsou pak testovány vzhledem k novému modelu, tzn. pro každý vzorek je určena míra shodnosti s modelem – určí se, jestli zapadá do modelu s danou tolerancí T . Pokud vzorek dobře odpovídá modelu, je zařazen jako hypoteticky bezchybný (přispívá k hodnotě K), je ohodnocen jako inlier bod. Proces je několikrát opakován (předem daný maximální počet opakování L) a vždy je vyprodukován buďto model, který je zamítnut, protože bylo příliš málo bodů vyhodnoceno jako bezchybných, nebo je vytvořen

vyhovující model i s odpovídajícími velikostmi chyb. Nakonec je získán nejlepší model, pro nějž je chyba minimální. Volba parametru K závisí na poměru bezchybných dat z celkového počtu M .

Pravděpodobnost, že ani po L opakováních nebude nalezeno N vzorků dostatečně odpovídajících modelu, je:

$$p_{fail} = (1 - (p_g)^N)^L. \quad (4.1)$$

Z čehož plyne maximální počet povolených opakování L :

$$L = \frac{\log(p_{fail})}{\log(1 - (p_g)^N)}. \quad (4.2)$$

Rychlost výpočtu metody RANSAC závisí:

- na velikost chybové kontaminace dat, která ovlivňuje nutný počet vybíraných bodů N
- na počtu požadovaných správných bodů K pro odsouhlasení modelu
- na celkovém počtu měřených dat M

4.3 Výhody a nevýhody

Výhodou metody je možnost robustního určení parametrů modelu, jelikož lze určit parametry modelu s vysokou přesností díky úplnému vyřazení chybných dat.

Nevýhodou metody je, že vyžaduje nastavení prahu pro určení množiny inlier bodů. Toto nastavení musí provést uživatel odhadem ze známých parametrů.

Další nevýhodou metody je, že neexistuje horní časová hranice určující dobu trvání výpočtu. Pokud je pevně stanovena, výsledný model nemusí být nutně ten nejlepší ze všech možností. Ideální počet iterací je tedy složité určit – návodem může být tabulka 4.1, která ukazuje optimální počet těchto iterací v závislosti na počtu vzorků a počtu outlier bodů.

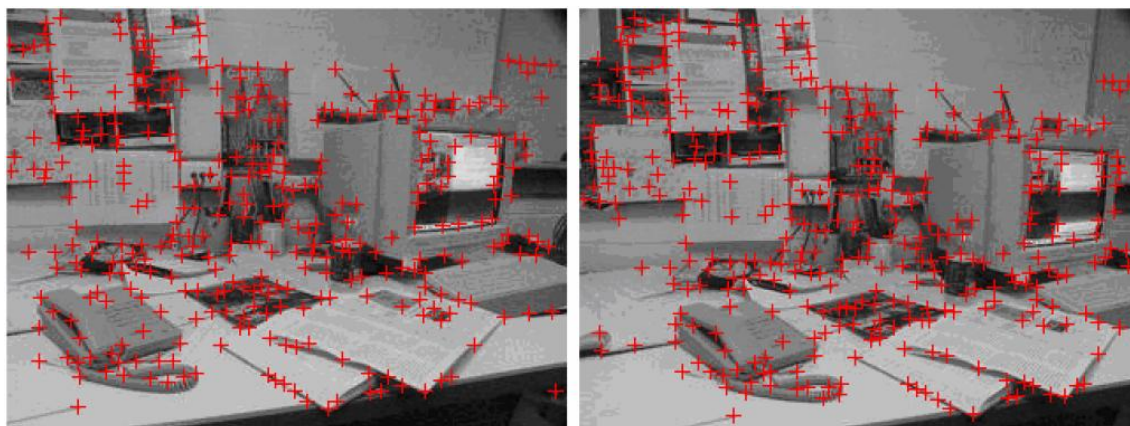
počet vzorků	procentuální výskyt hodnot (outliers), které nepatří hledanému modelu						
m	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Tabulka 4.1: Počet iterací pro různé parametry [14]

4.4 RANSAC na korespondenčním problému

Korespondenční problém je také předmětem řešení při spojování snímků a vytváření panoramatických fotografií, čili je významnou součástí této práce. Zde je jen naznačeno, jakou roli bude hrát RANSAC při spojování snímků, detailněji bude probrán v kapitole 6.3.

Na dvojici podobných snímků jsou nalezeny významné body – rohy, hrany apod. pomocí nějakého detektoru významných bodů, např. SIFT detektoru. V okolí každého významného bodu prvního snímku je hledána poloha nejpodobnějšího segmentu ze snímku druhého. Metodou RANSAC jsou z množiny všech nalezených možných korespondencí odstraněny ty, které nevyhovují nalezenému modelu (jsou chybné - outlier). Správné korespondence (inlier) jsou dále zpracovávány.



Obrázek 4.2: Detekce významných bodů podobných obrázků [2]



Obrázek 4.3: Správné korespondence (inlier - modré linie) a vyloučené korespondence (outlier - červené linie) nalezené metodou RANSAC [2]

5 Panoramatická fotografie

Hlavním cílem této práce je vytvořit aplikaci na výrobu panoramatických fotografií. V této kapitole popíšeme, co to vlastně je panoramatická fotografie, jakým způsobem se pořizuje a jakých postupů se využívá při její tvorbě.

V minulosti, kdy ještě pro úpravu fotografií nemohly být využívány počítače, bylo pořizování panoramat záležitostí fotografování jednoho snímku pomocí speciálního objektivu. V dnešní době je panoramatická fotografie ve výsledku spojením několika snímků (obrázků) do jedné fotografie. Vznikne tedy fotografie, která bude mít záběr pohledu všech spojovaných snímků. Výsledná fotografie by měla budit dojem, že se nejedná o spojení více snímků, ale o jednu fotografii pořízenou jako jeden snímek.



Obrázek 5.1: Ukázka panoramatické fotografie

Při pořizování snímků na panoramata je nutné dodržet několik zásad, aby výsledná fotografie měla co nejlepší výsledky. Při zachycování fotografií by mělo být optické zařízení ve stabilní poloze a pořízení všech snímků z jednoho místa, čili při fotografování je vhodné použít stativ. Další zásadou je zajištění konstantního osvětlení, tzn. při každém snímku, by mělo být nastavení čočky objektivu stejné (clona, čas expozice), v horším případě alespoň podobné. Pořizovatel by měl zaručit také zaostření na stejnou vzdálenost, aby objekty na fotografii nebyly v jednom místě ostré a v druhém rozmazané. Poslední zásadou je dodržení způsobu pořízení panorama. Způsob pořízení vybereme podle toho, jaké panorama požadujeme.

Existují 4 druhy panoramatických fotografií: horizontální, vertikální, slepované a panoramata v rozsahu 360°. Při pořizování horizontálních nebo vertikálních panoramat vycházíme z názvu, čili při horizontálních pořizujeme snímky vedle sebe a při vertikálních pod sebou. Slepovaná panoramata jsou kombinací horizontálních a vertikálních panoramat, čili jsou fotografovány vedle sebe i pod sebou. Tímto typem panoramatu můžeme vytvořit efekt tzv. rybího oka (obrázek 5.2). Poslední panoramata v rozsahu 360° jsou speciálním typem panoramat, kdy snímky jsou pořízeny ve všech směrech fotografova pohledu. Při prohlížení těchto fotografií je zapotřebí nějakého vizualizačního nástroje (často využíván Flash), ve kterém uživatel vidí pouze určitý výřez panoramatu a je schopen otáčet pohledem tak, aby měl pocit, že si prohlíží nekonečný pás (točí se pořád dokola).



Obrázek 5.2: Efekt rybího oka [10]

Při tvorbě panoramatických fotografií se používá mnoho různých druhů projekcí. Jsou rozděleny podle toho, na jaký geometrický objekt se projektují (mapují). Zde jsou popsány některé z nich:

- spojení snímků – klasický 2D plochý pohled na scénu, kdy jsou obrázky pouze jednoduše spojeny bez jakékoli perspektivy.
- sférická – tato projekce se mapuje na kouli. Používá se při tvorbě panoramat do úhlu záběru 120° , kdy nevzniká velká deformace protažením snímku. Takovýto negativní úkaz u této projekce při záběru větším jak 120° je znázorněn na obrázku 5.3.



Obrázek 5.3: Sférická projekce – protažení snímku díky velkému úhlu záběru [9]

- cylindrická – tato projekce se mapuje na válec. Používá se nejčastěji na panoramata s rozsahem 360° a vytváří přirozený pohled na fotografii i při tomto rozsahu. Ukázka cylindrické projekce je na obrázku 5.4.



Obrázek 5.4: Cylindrická projekce – panorama v rozsahu 360° [11]

Princip cylindrické (sférické) projekce

Zjednodušeně řečeno, mapování pomocí cylindrické projekce se provádí tak, že obrázek se „nalepí“ (namapuje) na hypotetický válec, čili body, které jsou v prostředním sloupci obrazu, zůstanou nezměněny (ty jsou nám nejbliže) a sloupce pokračující vlevo či vpravo se jakoby oddalují (v podstatě jejich pozice se posunuje více ke středu obrazu) – to způsobuje zakřivení hypotetického válce. Jak moc se oddálí (posunou ke středu) záleží na parametrech válce. Výsledkem projekce je pak pohled na válec zepředu tak, jak ho vidíme. Celý obraz se v horizontálním směru zúžil.

Sférická projekce má stejný princip s tím rozdílem, že se obraz nemapuje na hypotetický válec, ale na hypotetickou kouli. Mapování následně vypadá tak, že sloupce bodů se „neoddalují“ (nemapují) jenom v horizontálním směru, ale i ve vertikálním, čili už se nepracuje se sloupci, ale s body. Nejbliže nám je tedy bod, který je v samém středu obrazu. Výsledkem je tedy zúžený obraz v horizontálním i ve vertikálním směru.

Jedním z problémů spojování fotografií je vyhlazení přechodů mezi fotografiemi. Tento problém řeší profesionální programy celkem bez problémů, pokud jsou vstupní fotografie pořízeny za zhruba ideálních podmínek, čili nelze rozpoznat, kde fotografie začíná a kde končí. Větší problém však nastává, pokud fotografujeme záběr větší než 120° (pomocí speciálních objektivů), kdy se projevuje nepříjemné ztmavení rohů (objevuje se u méně kvalitnějších obyčejných objektivů). Pokud si s tím program neporadí, ve výsledné fotografii je vidět ztmavení hran spojovaných obrázků a výsledná fotografie nevypadá dobře. Tento problém je zapotřebí odstranit už při pořizování snímků. Fotografie vyfotografované takovýmto méně kvalitním fotoaparátem a jejich výsledek po spojení je ukázán na obrázku 5.5.

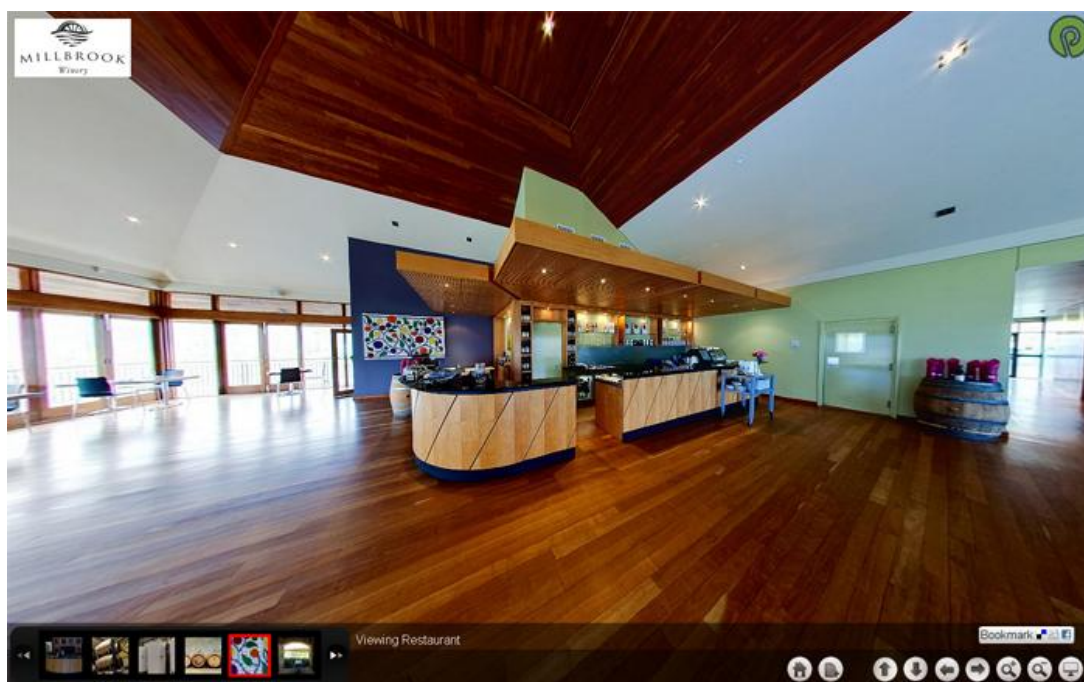


Obrázek 5.5: Spojení snímků se ztmavenými rohy

Pořízení snímků nemůže být nikdy ideální a vždycky se projeví nějaký problém. Odstranění těchto problémů nebo alespoň potlačení jejich důsledků je pak na programu, který vytváří panoramata. Profesionální programy, které vytvářejí panoramatické fotografie, spojí libovolné množství fotografií, vytvoří plynulý přechod na hranici dvojice snímků, a na uživateli je pak volba,

kteřou projekci použít, aby např. horizonty byly ve vodorovném směru, anebo stěny a svislé hrany objektů byly ve směru svislém a nebyly nijak zakřiveny. Čím více druhů projekcí program obsahuje, tím více druhů panoramat jsme schopni vytvořit.

Panoramata mají dnes široké možnosti využití a díky velkému úhlu záběru dopřávají divákovi úchvatný zážitek z fotografie. Panoramata jsou hodně používána při fotografování krajiny z nějakého vyvýšenějšího místa, aby záběr byl co největší a na fotografii byl zachycen co možná největší výhled na krajinu. Panoramata jsou také používána např. při tzv. virtuálním procházení místností, kdy celá místnost je nafotografována z jednoho místa a po spojení snímků se vytvoří panorama v rozsahu 360° a uživatel má možnost pohybovat se kurzorem myši nebo klávesnicí po panoramatu a vidět tak celou místnost z výřezu ve webovém prohlížeči. Tuto formu prohlížení místností využívají např. realitní kanceláře pro uživatele, kteří hledají a vybírají byty na internetu.



Obrázek 5.6: Ukázka virtuálního prohlížení místnosti [9]

6 Návrh řešení

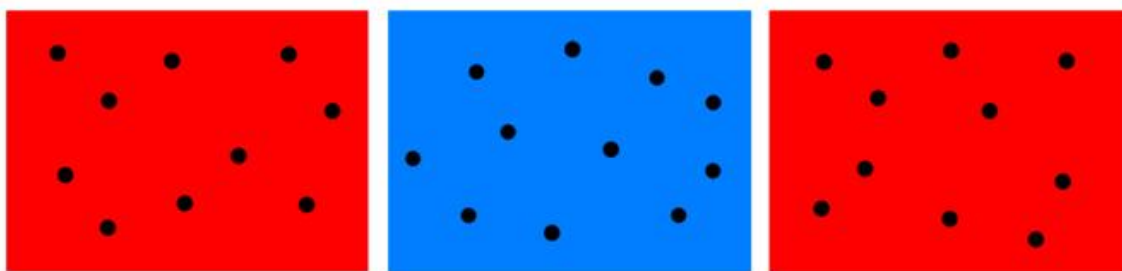
Cílem diplomové práce je vytvořit aplikaci, která bude spojovat fotografie a vytvářet panoramatické snímky (kapitola 5). V této kapitole jsou popsány metody vývoje a postup při vytváření této aplikace. Po důkladném prostudování metod a algoritmů popisujících detekci významných bodů a vyhledávání korespondenci mezi jednotlivými skupinami bodů v obrázcích jsem dospěl k použití metod SIFT a RANSAC. Důvod použití těchto algoritmů je uveden níže.

6.1 Rozbor navrženého řešení

Proces skládání snímků se dá rozdělit na několik částí:

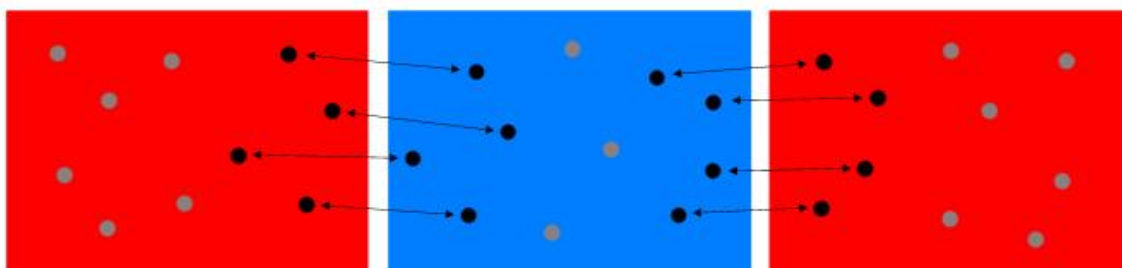
- detekce významných bodů,
- vyhledání korespondence,
- transformace obrázků a jejich vykreslení,
- vyhlazení přechodů,
- srovnání úrovní jasu.

Ze vstupních obrázků se provede detekce významných bodů. Tuto detekci bude provádět algoritmus SIFT popsany v kapitole 3. Výstupem detekce bude množina bodů, která obsahuje souřadnice bodů i s jejich popisem ve formě deskriptorů bodů.



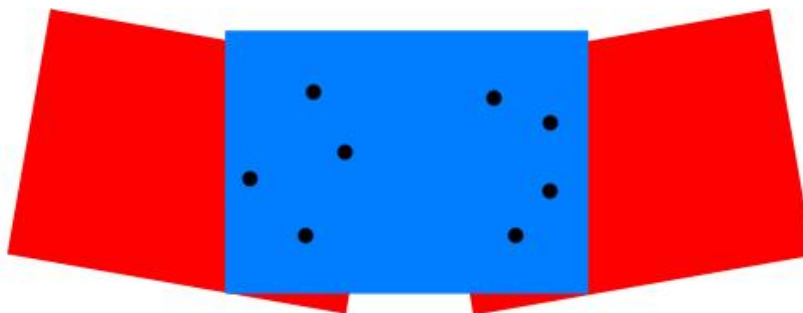
Obrázek 6.1: Nalezené významné body algoritmem SIFT

Z nalezených významných bodů se algoritmem RANSAC zjišťuje geometrická konzistence, tím že se hledají korespondenční body. Detailnější popis je v kapitole 6.3.



Obrázek 6.2: Korespondence mezi jednotlivými body, které našel algoritmus RANSAC

Po nalezení korespondencí mezi jednotlivými body se pomocí transformační matice, která byla vypočtena z transformací v průběhu výpočtu algoritmu RANSAC, transformují požadované obrázky tak, aby představovaly výsledné panorama.



Obrázek 6.3: Transformované obrázky do výsledného panorama

Během transformací obrázků i po nich se aplikují techniky, které se snaží vyhladit přechody mezi spojovanými snímky. Posledním krokem vytváření panoramatu je srovnání úrovní jasu všech vstupních obrázků. Tento krok se však provádí ještě před detekcí významných bodů, čili po dobu procesu výroby panorama jsou tyto úrovně již srovnané.

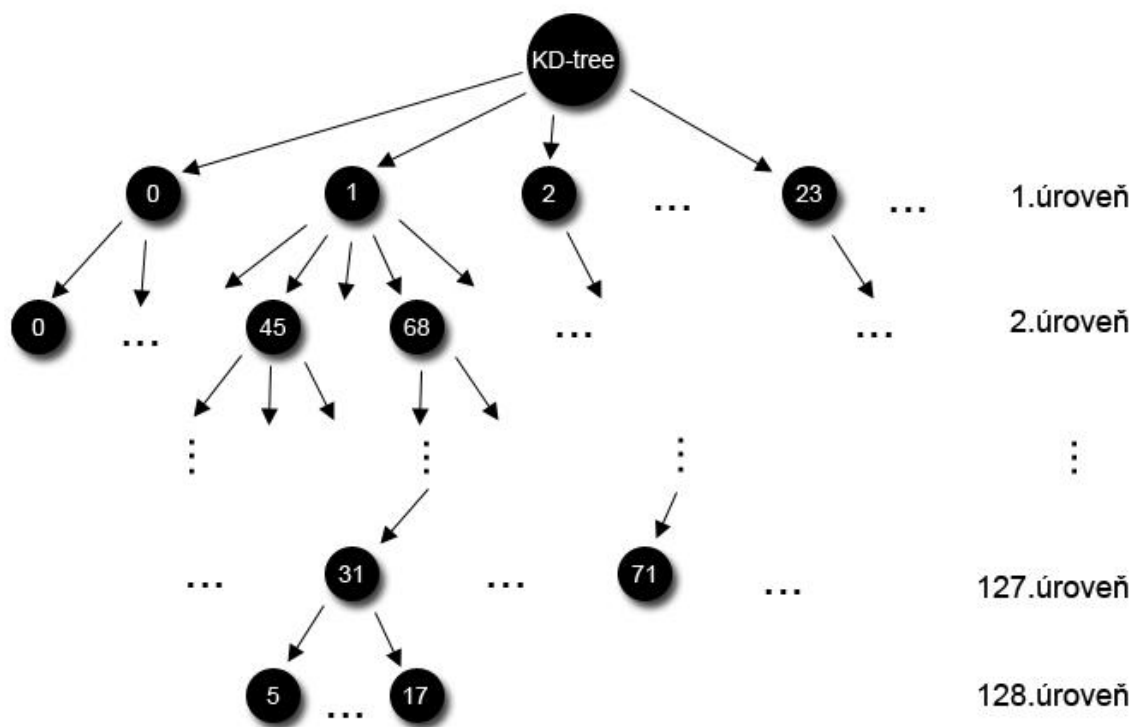
V následujících podkapitolách je tento postup detailněji rozebrán.

6.2 Detekce významných bodů

Základem panoramatické fotografie je detekce významných bodů. Abychom mohli zjistit, jak fotografie transformovat tak, aby na sebe plynule navazovaly a byly součástí jednoho celku, je nutné najít v obraze nějaké body, podle kterých jsme schopni se orientovat. Proto je třeba zvolit vhodný detektor významných bodů. V kapitolách 2 a 3 jsem rozebral detektory s různým přístupem k obrazu. Podle rozsahu popisu detektorů je patrné, že na detekci významných bodů pro svou aplikaci jsem použil algoritmus SIFT. Tento algoritmus se vyznačuje invariancí vůči afinním transformacím obrazu (měřítko, posunutí, rotace). Díky invarianci vůči těmto transformacím a značně detailnímu popisu bodů pomocí deskriptorů je SIFT téměř ideální metodou pro výrobu panoramatických fotografií, neboť při pořizování snímků fotograf nemůže stoprocentně zaručit, že pořízené snímky budou mít rovnoběžné horizonty (obrázky nebudou nijak pootočené) nebo mít stejné měřítko. Právě při pořizování snímků je kladen největší důraz na to, aby fotograf měl, co největší „svobodu“ – toto hledisko je jedno z nejdůležitějších při určování kvality programu pro výrobu panoramatických fotografií. Proto zvolení algoritmu SIFT při výrobě panoramatických fotografií jsem hodnotil jako dobrou volbu.

Aplikace bude moci spojovat nekonečné množství obrázků. Je proto nutné postupně vzít jeden obrázek po druhém a aplikovat na něj detektor SIFT. Výsledkem samotného vyhledávání detektoru SIFT je pole obsahující souřadnice bodů datových typů double, čili neceločíselných datových typů. Tento datový typ je zaveden kvůli přesnější subpixelové lokalizaci významných bodů v obrázku (kapitola 3.3), kdy algoritmus si vypočítává významné body i v prostoru mezi celočíselnými konstantami.

Hodnoty výsledného pole významných bodů jsou uloženy pro přehlednější a rychlejší hledání do datové struktury kd-tree strom. Strom obsahující významné body nese v sobě i informaci o deskriptorech těchto bodů. Parametry deskriptoru jsou vypočítány z gradientů obrázku v okolí klíčového bodu (viz kapitola 3.6). Tento strom má 128 úrovní, protože deskriptory mají 128 parametrů. Díky tomu je možné vyhledávat body pomocí deskriptorů a deskriptor je tedy klíčem hledání v tomto stromě. Vyhledávání v kd-tree stromě se děje pomocí algoritmu Best-Bin-First. Tento algoritmus je uzpůsoben pro vyhledávání ve více rozměrném kd-tree stromu, čili je pro náš případ velice vhodný. Deskriptory se dále využijí při hledání korespondence.



Obrázek 6.4: KD-tree pro ukládání bodů; čísla jsou hodnoty deskriptoru bodu a podle nich se tento bod mapuje

6.3 Vyhledání korespondence

Nyní máme zjištěny významné body obrázků. Ke spojení obrázků je nutné vyhledat korespondující body mezi jednotlivými obrázky a označit je za shodné. Tímto získáme propojení všech obrázků, pokud tedy nějaké existuje, a následně můžeme tyto obrázky spojit.

Pro vyhledání korespondence jsem vybral algoritmus RANSAC (popsán v kapitole 4). Tento algoritmus náhodným testováním vzorků vyfiltruje významné body, které se neshodují s žádnými dalšími významnými body v jiných obrázcích, označí tyto body, které nějakou korespondenci mají a vytvoří z nich geometrický model, ze kterého se následně vypočte transformační matice. Zjednodušeně řečeno: RANSAC hledá nejvhodnější parametry geometrického modelu náhodným testováním vzorků. Nalezení správných parametrů modelu závisí na dvou aspektech:

- požadovaný počet bodů modelu
- minimální počet bodů k určení modelu

Požadovaný počet bodů je číslo, které udává kolik bodů je nutných k vytvoření geometrického modelu. Toto číslo je potřeba odhadnout, proto bude uživateli v aplikaci umožněno ho zadat. Minimální počet bodů k určení modelu by měl být minimálně 2, abychom mohli zjistit parametry transformace (posunutí, rotace, měřítko). K přesnějším výsledkům budeme v aplikaci počítat minimálně se třemi body.

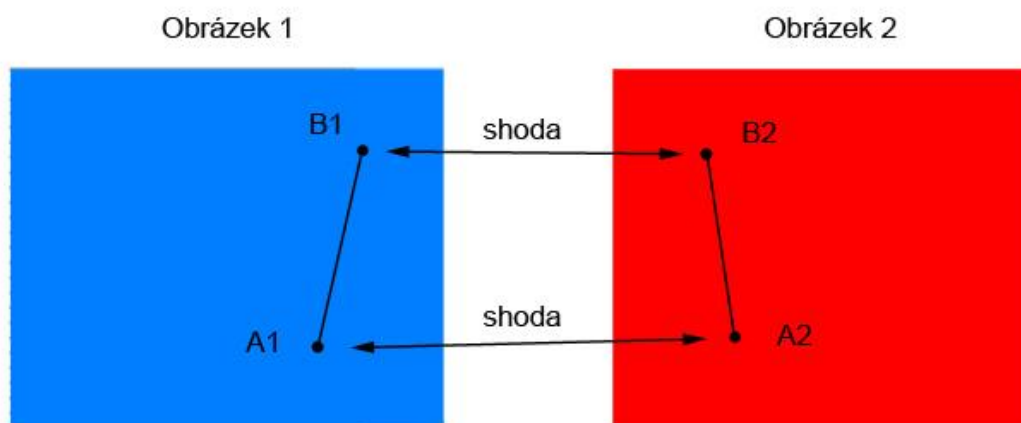
Geometrický model má za úkol mapovat souřadné systémy mezi dvěma obrázky. Jelikož všechny obrázky jsou dvourozměrné a předpokládáme, že za účelem vytvoření panoramatické

fotografie jsou zhruba ve stejném měřítku, budeme volit geometrický model jako dvourozměrnou transformační matici homogenních souřadnic obou vstupních obrázků. Tuto matici nám vytvoří algoritmus RANSAC postupným vybíráním náhodných dvojic významných bodů z různých obrázků (zde algoritmus začíná).

Výpočty, které se budou provádět s body a s transformační maticí, musí být v homogenních souřadnicích. Tento převod je nezbytně nutný, neboť transformační matice je tvaru 3x3 (viz rovnice 6.5) a násobení této matice a bodu v kartézském souřadném systému, který má pouze 2 rozměry (x, y), není možné. Převod se provede přidáním z souřadnice, která se vždy bude rovnat 1, čili bod v homogenních souřadnicích bude mít tvar [x,y,1].

Princip metody RANSAC pro účely aplikace

Algoritmus tedy zpočátku náhodně vybere dvojici bodů z *obrázku 1* a k ní dvojici korespondenčních bodů z *obrázku 2*. Pro lepší představu spojíme body v jednotlivých obrázcích a tím nám vzniknou přímky, jak je naznačeno na obrázku 6.5.



Obrázek 6.5: Shodná dvojice přímek použita pro výpočet transformace z obrázku 2 do obrázku 1

Nyní je třeba vypočítat transformace, které existují mezi přímkami A1-B1 a A2-B2. Tento výpočet provedeme tak, že „namapujeme“ významné body z *obrázku 2* do *obrázku 1*. Proces mapování významných bodů je rozdělen na několik fází:

- posunutí T1 – bod A2 je posunut do počátku své soustavy souřadnic, čili

$$T1 = T(-A2_x; -A2_y) \quad (6.1)$$

- rotace R – přímka A2-B2 se „odrotuje“ o úhel α

$$\alpha = \arctan^2(B1_y - A1_y; B1_x - A1_x) - \arctan^2(B2_y - A2_y; B2_x - A2_x) \quad (6.2)$$

- změna měřítka S – změna měřítka s přímkou A2-B2 na velikost přímky A1-B1

$$s = \left(\frac{|B1 - A1|}{|B2 - A2|} \right) \quad (6.3)$$

- posunutí T2 – bod A2 je posunut na pozici bodu A1, čili

$$T2 = T(A1_x; A1_y) \quad (6.4)$$

Vypočítané transformace nyní převedeme do transformační matice, která bude mít tento tvar:

$$M = \begin{bmatrix} s * \cos(\alpha) & s * (-\sin(\alpha)) & s * (\cos(\alpha) * (-A2_x) - \sin(\alpha) * (-A2_y)) + A1_x \\ s * \sin(\alpha) & s * \cos(\alpha) & s * (\sin(\alpha) * (A2_x) + \cos(\alpha) * (-A2_y)) + A1_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6.5)$$

Nyní potřebujeme zjistit, jestli počáteční náhodný výběr shodných souřadnic byl správný. Vezmeme postupně všechny významné body $P(x; y)$ z *obrázku 2* v homogenních souřadnicích a vynásobíme je transformační maticí M . Výsledkem je pozice transformovaného bodu P' v *obrázku 1*.

$$P' = M * P = M * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6.6)$$

Ke zjištění, jestli transformovaný bod P' leží poblíž korespondujícího bodu z *obrázku 1*, nám bude sloužit funkce d . Tato funkce zjistí vzdálenost mezi předpokládanými dvojicemi bodů. Funkce vypadá následovně:

$$d(P, M, K) = |K - M * P| \quad (6.7)$$

kde K je korespondující bod z *obrázku 1*.

Výsledek funkce d bude jeden z následujících čtyř:

- d je malé, shoda je správná
 - Toto je náš požadovaný výsledek. Vzdálenost d je malá, proto se můžeme domnívat, že jsme na začátku vybrali správnou korespondenci. Transformační matice je uložena.
- d je malé, shoda je nesprávná
 - Tento případ je nepravděpodobný, ale může nastat, pokud existuje nějaká shoda zapadající do výsledného geometrického modelu. Tento případ může nastat např. u řady stejných oken seřazených vedle sebe. Způsob, jak zabránit této události, může být takový, že kdykoliv algoritmus nalezne dva významné body z *obrázku 2* mapující se do stejného významného bodu z *obrázku 1*, oba tyto body odstraní, neboť jen těžko můžeme určit, který z nich je ten správný. To by mělo odstranit většinu vyskytujících se nesprávných shod u opakujících se objektů.
- d je velké, shoda je správná
 - Toto je případ, který nechceme, aby nastal u panoramatického snímku. Případ může nastat při pohybu objektů (např. aut, lidí), kdy objekty jsou výrazně posunuty vzhledem k pozadí. Tato shoda je odstraněna.
- d je velké, shoda je nesprávná
 - Tento případ nastává nejčastěji, neboť počet významných bodů obrázku je většinou větší než požadovaný počet bodů modelu. Tato shoda je odstraněna.

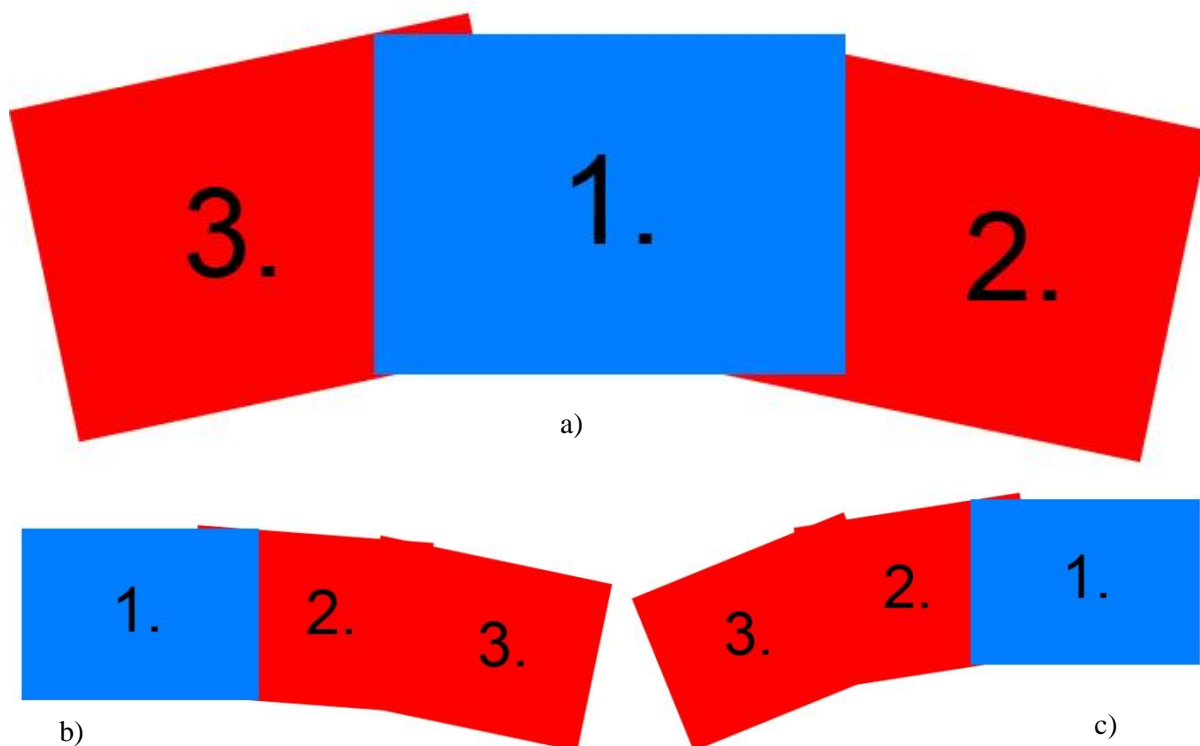
Pokud tedy transformovaný bod P' leží poblíž bodu K (d je dostatečně malé) a shoda je správná, je tato korespondence uložena do geometrického modelu a vybere se další náhodná dvojice k testování na shodu. Takto algoritmus RANSAC testuje všechny významné body na jejich korespondenci, dokud není splněn limit požadovaného počtu bodů. Samozřejmě musí být ještě

splněna podmínka minimálního počtu bodů modelu, jinak algoritmus vyhodnotí situaci tak, že nebyla nalezena žádná shoda.

6.4 Transformace obrázků a jejich vykreslení

Významné body, jejich korespondence a výslednou transformační matici máme již vypočteny. Nyní je třeba pomocí této matice transformovat vstupní obrázky.

Ještě než uživatel spustí proces vytváření panoramatu, musí vybrat ze seznamu obrázků *středový obrázek*¹. „Středový“ neznamená, že bude uprostřed panoramatu, ale že to bude základní obrázek, od kterého se bude vyvíjet panorama. Na obrázcích 6.6 jsou naznačeny kombinace, které mohou nastat při zvolení *různého středového obrázku*.



Obrázek 6.6: Možné kombinace panoramat při zvolení *různého středového obrázku*

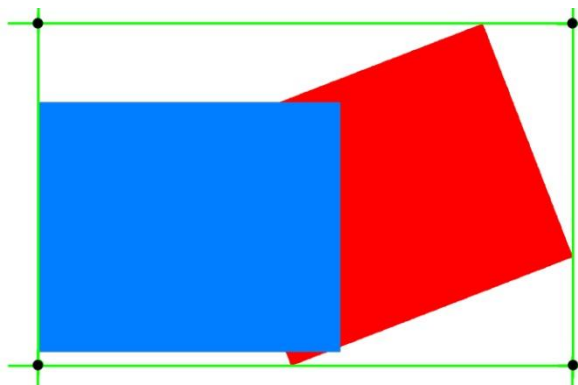
Na obrázcích 6.6b a 6.6c jsou zobrazeny kombinace při zvolení *středového obrázku* na okraji panoramatu. Tato volba není doporučena, neboť pokud panorama obsahuje nějaké horizontální linie, tak na straně *středového obrázku* (např. na obrázku 6.6b vlevo) je vše v pořádku, ale na opačné straně (na obr. 6.6b vpravo) je linie už zřetelně deformovaná. Na obrázku 6.6a je zvolen *středový obrázek* uprostřed. Pokud panorama obsahuje nějaké horizontální linie, tak ve výsledku linie na krajních obrázcích není až tak deformovaná jako v předchozím případě. Proto zvolení *středového obrázku* uprostřed je silně doporučeno.

Nyní přistoupíme k vybrání obrázků, které se mají spojit. Abychom splnili možnost vytvoření panoramatu z nekonečně mnoha obrázků, celou tvorbu panorama si rozdělíme na spojování jednotlivých dvojic. Pro názornost vysvětlím vybírání na obrázku 6.6a. První dvojici budou tvořit *středový obrázek* a jeho pravý následovník. *Středový obrázek* zůstane nezměněn a jeho pravý následovník se transformuje podle vypočtené transformační matice (od tohoto místa dále budu

¹ *Středový obrázek* – obrázek, od kterého se odvíjí ostatní obrázky panorama; je to označení obrázku, který nebude nijak transformován, ale bude pouze vložen do výsledného panoramatu a ostatní obrázky se budou na něho „nalepovat“.

používat termín *středový obrázek* ve významu statického obrázku, čili obrázku, který se nebude transformovat, ale podle něho se budou ostatní obrázky transformovat). Po spojení vytvoříme z této dvojice tzv. dočasné panorama, které si uložíme do souboru a opět načteme jako další obrázek. Tento obrázek si označíme jako *středový* a do dvojice vezmeme třetí obrázek, který je vlevo od prvního. Z dalších ukázek kombinací panoramat z obrázků 6.6 lze odvodit postup vybírání. Začíná se od středového obrázku a vybírají se postupně obrázky z pravé větve (čím blíže je obrázek obrázku *středovému*, tím dříve je vybrán). Pokud jsou všechny obrázky z pravé větve vyčerpány, začíná se brát z větve levé. I pro levou větev platí stejná pravidla jako pro větve pravou. Jinak řečeno, po každém spojení jedné dvojice se dočasné panorama uloží do souboru, označí se jako *středový obrázek* a přidá se k němu pravý nebo levý následovník, podle toho jestli není nebo je pravá větev již zcela vybrána. Počet dočasných panoramat (části panorama, které jsou během vytváření uloženy do souboru) je tedy o jeden obrázek menší než počet vstupních obrázků.

Pro samotnou transformaci musíme prvně zjistit, jak velké bude výsledné panorama, abychom věděli, jak velké pole pro výsledný obrázek máme alokovat. Velikost zjistíme z pozic krajních bodů spojovaných obrázků. *Středový obrázek* bude vždy neměnný, čili transformujeme pouze krajní body přidávaného obrázku. Nyní máme 8 bodů s hodnotami pozic x a y a hledáme jednoduchým porovnáváním nejvyšší (nejnižší) hodnoty souřadnic zvlášť pro hodnoty x a zvlášť pro y . Následnou správnou kombinací těchto souřadnic nám vzniknou rohové body panorama. Situace je naznačena na obrázku 6.7.



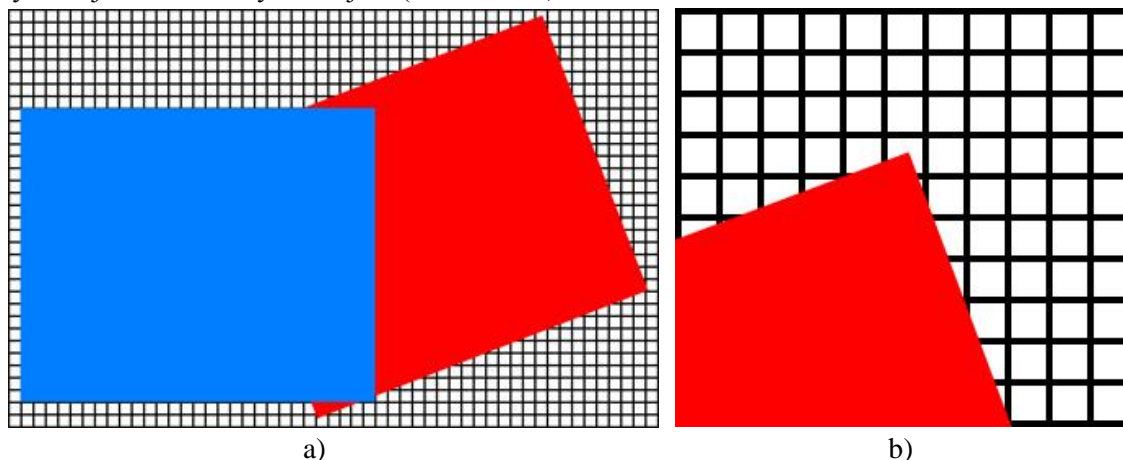
Obrázek 6.7: Souřadnice výsledného panorama

Při vykreslování dvojic obrázků pracujeme s barevným modelem RGBA, který obsahuje kromě základních barev červené, zelené a modré také *alfa* složku k nastavení míry průhlednosti bodu.

Na začátku vykreslování si připravíme černé plátno o velikosti, kterou jsme zjistili v předešlém odstavci. Nyní vložíme *středový obrázek* do tohoto plátna. Tento obrázek má být bez jakékoli transformace, proto hodnoty barevných složek jeho obrazových bodů jednoduše nakopírujeme do plátna. Odsazení obrázku zjistíme vzhledem k bodům, které nám vyšly jako krajní body panorama. První obrázek máme na plátně, nyní je třeba vložit druhý, transformovaný.

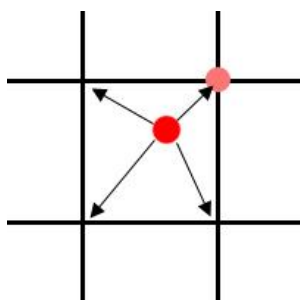
Pro vložení druhého obrázku do plátna budeme postupně procházet všechny souřadnice bodů plátna a pomocí inverzní transformační matice zjišťovat, jestli jsme zrovna na pozici souřadnice bodu druhého obrázku. Čili při procházení plátna vezmeme aktuální souřadnice plátna, vynásobíme je inverzní transformační maticí (vše je prováděno ve výše zmíněných homogenních souřadnicích) a zjistíme, jestli jsme se trefili do souřadnic druhého obrázku. Pokud ano a na aktuální pozici plátna je černá barva, vezmeme barevnou hodnotu bodu druhého obrázku a vložíme ji na aktuální pozici plátna. Pokud ne, nic se neděje a pokračujeme dále. Takto program projede všechny souřadnice bodů plátna.

V tomto okamžiku máme na plátně oba vykreslené obrázky. Nastal zde však problém. Výpočty souřadnic bodů, které se prováděly násobením transformační matice a souřadnicemi bodů, nevycházejí v celočíselných údajích (obrázek 6.8).



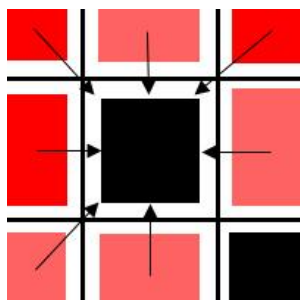
Obrázek 6.8: a) Spojení dvou fotografií, b) Detail rohu transformované fotografie (roh nezapadl do mřížky, má neceločíselné souřadnice)

Tento fakt je pro nás nežádoucí, neboť mapování souřadnic plátna se provádí pouze v celočíselných hodnotách. Je tedy nutné provést konverzi výsledného bodu do celočíselných údajů. Tímto krokem se dopouštíme nepřesnosti z hlediska zaokrouhlení. Situace je znázorněna na obrázku 6.9.



Obrázek 6.9: Zaokrouhlení bodu na celočíselné hodnoty

Díky této nepřesnosti se mohou transformace bodu zaokrouhlit na jiné souřadnice, bod se posune na jinou pozici a na jeho místě, kde se přepokládalo, že se bude nacházet, vznikne tzv. díra. Tyto díry je nutné nějak „zacet“ (interpolovat). Díru jsem interpoloval pomocí jejího 8mi okolí (v kapitole 8.1 je vysvětleno proč jsem tak učinil). Proto po vykreslení obou obrázků na plátno opět procházíme všechny body plátna a opět díky inverzní transformační matici zjišťujeme, jestli bod náleží do druhého obrázku. Pokud ano a na jeho místě je černá barva, zjistí se barevné hodnoty jeho 8 sousedů a vypočte se průměrná barva pro tento bod (obrázek 6.10). Pokud některý z jeho sousedů má černou barvu, nebere se tento bod v potaz.



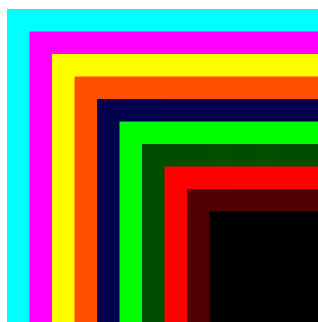
Obrázek 6.10: Interpolace díry

V tomto okamžiku máme spojené obrázky se „zacelenými“ dírami. Panorama však ještě není dokonalé. Stále jde rozeznat hranice mezi jednotlivými obrázky. Tento problém je vyřešen dalšími technikami v následujících podkapitolách.

6.5 Vyhlazení přechodů

Jedním z největších problémů při skládání panoramatických fotografií je odstranění viditelného přechodu mezi jednotlivými obrázky. Pokud si program na vytváření panoramat s tímto problémem neporadí dostatečně dobře, na panoramatu půjdou vidět hranice vstupních obrázků a to určitě ve výsledku nevypadá dobře. Řešení vyhlazení přechodů je obtížné, proto tento problém odstraníme pouze částečně.

Pro vyřešení budeme využívat alfa složku obrazových bodů. V kapitole 6.4 v části, kdy kopírujeme *středový obrázek* do výsledného plátna panoramatu, nastavíme různé hodnoty *alfa* složky prvních 10 krajních úrovní bodů tohoto *středového obrázku*. Na obrázku 6.11 je zobrazen levý horní roh obrázku a úrovně 10 krajních bodů jsou znázorněny různými barvami. Černá barva znázorňuje body, u kterých se *alfa* složka nemění, je tedy nastavena na hodnotu 255. Čím dál je úroveň bodů od středu obrázku, tím menší je hodnota *alfa* složky. 10 jasových úrovní se rovnoměrně rozloží do rozsahu *alfa* složky od 0 do 255.



Obrázek 6.11: 10 jasových úrovní středového obrázku

Při vykreslování druhého transformovaného obrázku se kontroluje, jestli se bod bude vykreslovat na pozici, kde už nějaký bod je vykreslen nebo ne. Pokud se vykresluje na pozici, kde už nějaký bod vykreslen je, přidáme zde kontrolu, jakou hodnotu má jeho *alfa* složka. Pokud bude mít jinou hodnotu než 255, váženým průměrem sloučíme hodnoty barevných složek středového a transformovaného obrázku. Rozložení barev je následovné:

<i>Alfa</i> složka	Podíl barvy 1. obrázku	Podíl barvy 2. obrázku
25	0,1 * hodnota	0,9 * hodnota
50	0,2 * hodnota	0,8 * hodnota
75	0,3 * hodnota	0,7 * hodnota
100	0,4 * hodnota	0,6 * hodnota
125	0,5 * hodnota	0,5 * hodnota
150	0,6 * hodnota	0,4 * hodnota
175	0,7 * hodnota	0,3 * hodnota
200	0,8 * hodnota	0,2 * hodnota
225	0,9 * hodnota	0,1 * hodnota
255	1,0 * hodnota	0 * hodnota

Tabulka 6.1: Vážené rozložení barev na přechodu

Podle tabulky 6.1 se tedy zprůměruje výsledná barevná hodnota bodu v panoramatu. Až se toto zprůměrování provede pro všechny body, které mají jinou *alfa* složku než 255, nastaví se pro všechny body panoramatu *alfa* složka na hodnotu 255, aby žádný bod nebyl průhledný.

6.6 Srovnání úrovní jasu

Další technikou, která může výrazně přispět k lepšímu vzhledu panorama, je vyrovnání úrovní jasu všech spojovaných obrázků. Při pořizování snímků se může stát, že fotograf nenastaví stejné parametry fotografie (clonu, expozici) a pořízené snímky nebudou mít stejný jasový odstín, čili některé mohou být světlejší nebo tmavší. Pokud jsou fotografie odděleně, nemusí být tento rozdíl patrný, ve výsledném panoramatu však jde vidět velice zřetelně a fotografie tak nebudí dojem jednoho celku.

V aplikaci nebudeme až tak úplně vyrovnávat jasové úrovně, ale spíše barevné hodnoty RGB obrazových bodů. To však ve výsledku bude mít stejný účinek jako vyrovnávání celkového jasu obrazu. Pro vyrovnání barevných hodnot obrázků je zapotřebí zjistit průměrnou hodnotu červených, zelených a modrých složek ze všech obrázků. Cyklem tedy projedeme všechny barevné hodnoty všech obrázků panorama, sečteme je a nakonec vydělíme počtem bodů všech obrázků. Tím získáme průměrné hodnoty barevných složek ze všech obrázků. Tyto hodnoty pak zpětně porovnáme s hodnotami jednotlivých obrázků a vypočteme rozdíl hodnot. Rozdíl se bude pohybovat řádově v jednotkách. Ten pak přičteme nebo odečteme od každé hodnoty obrazového bodu. Tím zajistíme, že barevné složky budou mít světlejší nebo tmavší odstín a dosáhneme požadovaného vyrovnání barev.

7 Implementace

V této kapitole popíši implementační nástroje, které jsem využil při tvorbě aplikace.

Celá aplikace je napsána v jazyce C# v implementačním prostředí Microsoft Visual Studio 2005. V projektu jsou využity funkce z balíčku knihoven Microsoft .NET Framework SDK 2.0, proto při kompilaci nebo spouštění aplikace je nutné mít tento balík nainstalován.

Pro detekci významných bodů obrazu jsem se rozhodl využít knihovnu libsift [1]. Tato knihovna obsahuje implementované algoritmy SIFT a RANSAC, které jsou potřebné pro detekci a vyhledání korespondence významných bodů. Knihovna je uzpůsobená pro detekci korespondence mezi vstupními obrázky a je použita v existujícím programu autopano-sift [12], který je součástí většího projektu pro skládání panoramat z názvem Hugin. Díky programu autopano-sift jsem snáze pochopil práci s touto knihovnou. Tento projekt je od stejného autora jako knihovna libsift, čili i hlavičky zdrojových kódů jsou opatřeny názvem tohoto programu. Program však vůbec neřeší výsledné skládání panoramatické fotografie, ale jen detekci korespondujících bodů vstupních obrázků. Rozhodl jsem se tedy vytvořit program, který vytvoří obrázek výsledného panoramatu s použitím této knihovny.

Vstupní obrázky mohou být načteny ve formátu BMP, JPG, JPEG, JPE, JFIF, TIF, TIFF. Výsledné panorama může být pak uloženo do formátů BMP, JPEG, GIF, TIFF a PNG.

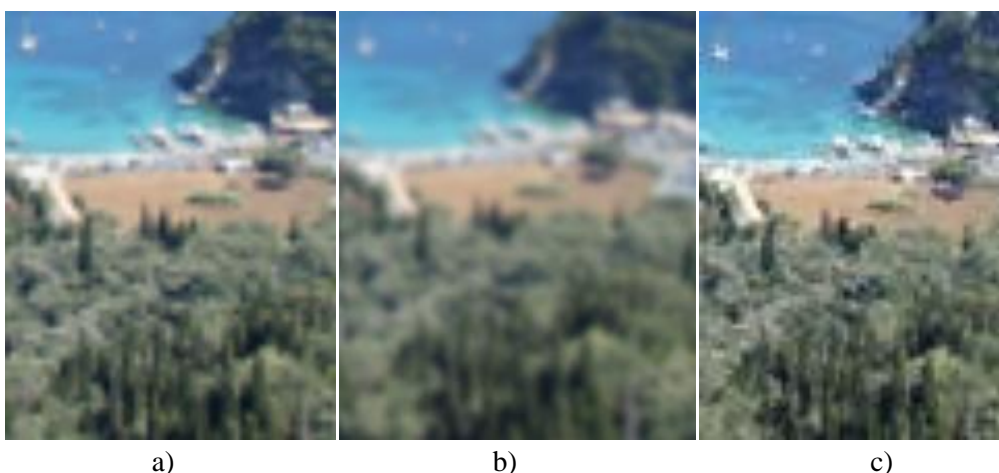
8 Testování a zhodnocení výsledků

Tato kapitola popisuje výsledky testování, jejich zhodnocení a vysvětlení použití některých metod.

8.1 Interpolace transformovaného obrazu

Pokud potřebujeme transformovat obrázek, důležitou roli zde hraje interpolace celého obrázku. Závisejí na ní výsledný zřehled panorama.

Při rozhodování jakou interpolací zvolit jsem měl na výběr z několika možností. První z možností byla využití služeb knihoven AForge.NET. Tento soubor knihoven je napsán pro jazyk C# a je navržen pro oblast počítačového vidění a umělé inteligence. V tomto souboru je knihovna AForge.Imaging.Filters, která nabízí spoustu funkcí pro práci s obrazem. Z této knihovny jsem tedy využil funkce pro změnu velikosti (ResizeBicubic) a rotaci obrazu (RotateBicubic). Funkce měly znázorňovat transformace změny měřítka a otočení. Jak je patrné z názvu funkcí, byla použita bikubická interpolace. Tato interpolace je nejkvalitnější z nabízených interpolací v knihovně.



Obrázek 8.1: Interpolace a) originál, b) bikubická z AForge, c) 8mi okolím

Podle výsledků na obrázcích 8.1 se bikubická interpolace příliš neosvědčila. Detail fotografie je rozmazaný a celkový pohled na panorama s touto interpolací transformovaných obrázků nebyl uspokojivý. Díky rozostřeným obrázkům bylo vyloženě patrné, které obrázky byly transformované a které ne.

Další použitelnou možností bylo interpolovat obrázek „8mi okolím bodu“ (viz kapitola 6.4). Tento druh interpolace v podstatě neexistuje, čili bylo nutné vytvořit vlastní verzi. Na obrázku 8.1c je výsledek této interpolace. V porovnání s bikubickou interpolací z knihovny AForge vypadá detail fotografie již mnohem lépe. Při detailnějším pohledu má fotografie schodovité hrany, což však budí celkový dojem ostřejší fotografie. Proto jsem zvolil tuto metodu.

8.2 Vyhlazení přechodu

V kapitole 6.4 je rozebrána technika vyhlazení přechodů mezi dvojicemi spojovaných obrázků. Pozorný čtenář může z tohoto textu vydedukovat, že v místě, kde má dvojice obrázků společné body se neprovádí žádná sumarizace nebo průměr barev. Na těchto souřadnicích jsou barevné hodnoty pouze *středního* (netransformovaného) obrázku.



a)

b)

Obrázek 8.2: Přejchod mezi dvojicí obrázků a) průměrování barev, b) přechod s originálem

Na obrázcích jsou zachyceny výsledky při vyhlazování přechodů. Červené čáry vymezují prostor společných bodů obou fotografií. Na obrázku 8.2a je výsledek přechodu při průměrování barev. Jde zřetelně vidět, že některé body na hranách se mísí mezi sebou a tím vzniká v těchto místech drobný šum. Tento výsledek dávám za příčinu nepřesnosti transformační matice. Věřím, že například po použití nějaké projekce (sférické nebo cylindrické) by byl transformovaný obrázek namapován na geometrický objekt (koule nebo válec), tím by se shodné body k sobě více přiblížily a celkový výsledek na přechodu by vypadal lépe.

Pro vyřešení vyhlazení přechodu jsem tedy zvolil jednoduché řešení, a to ponechat v oblasti přechodu hodnoty obrazových bodů netransformovaného obrázku (*středového obrázku*). K tomuto závěru mě vedl fakt, že obrazové body na přechodu budou mít hodnoty originálu a to je nejlepší možný výsledek (obrázek 8.2b).

8.3 Průměrování na hraně obrázku

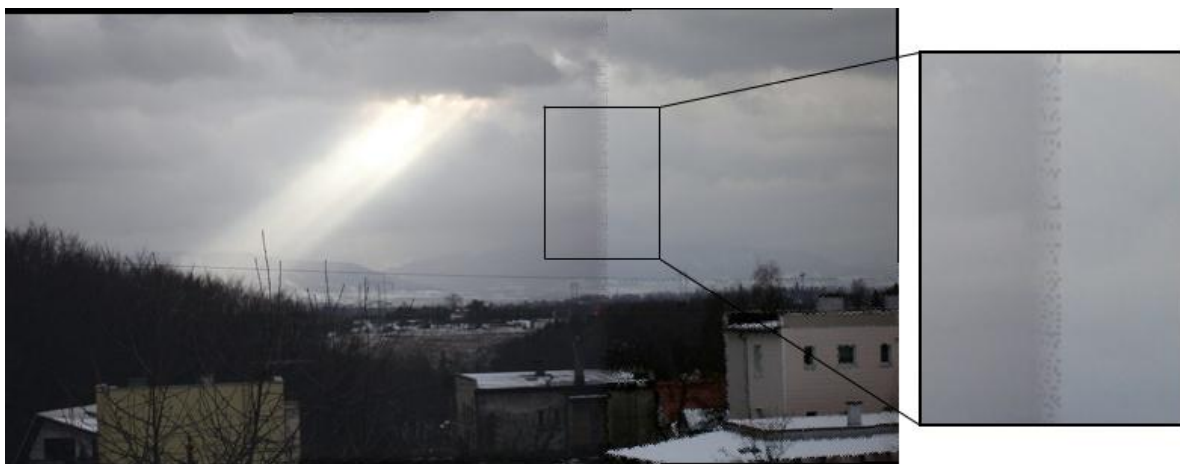
Pokud v oblasti přechodu jsou hodnoty obrazových bodů netransformovaného obrázku, jediný přechod vzniká tedy na hraně tohoto obrázku. Je tedy nutné vyhladit tuto hranu.



Obrázek 8.3: Vyhlazení přechodu bez vyrovnání jasu

Problém je vyřešen váženým průměrováním hodnot obrazových bodů (viz kapitola 6.5). Tady však nastává stejný problém jako v předchozí kapitole 8.2, kdy barvy bodů na hraně se mísí mezi sebou a vytvářejí nepříjemný bodový šum.

Tento šum je určitě nepříjemný, avšak tento šum nemusí vůbec vzniknout. Jak je vidět z obrázku 8.3 celkové panorama není správně vyfotografované. Levý obrázek je více podexponován než pravý. Podexponování je poměrně markantní, ale není zde použita funkce na vyrovnání jasu. Zkusíme tedy tuto funkci použít.



Obrázek 8.4: Vyhlazení přechodu s vyrovnáním jasu

Na obrázku 8.4 je použita funkce vyrovnání jasu. Přechod se poměrně zlepšil, ale pořád na hraně obrázku jde vidět bodový šum.

Jak jde vidět, kvalita přechodu mezi snímky v panorama závisí na kvalitě pořízení snímků. Pokud se ovšem fotograf bude řídit pravidly, které jsou popsány v kapitole 5, a dodrží stejné parametry fotografií (clona, čas expozice), snímky pak budou jasově vyvážené a přechod by neměl být vidět.

8.4 Srovnání úrovní jasu

Tato funkce v některých případech výrazně vylepšuje výsledky panorama. Konkrétní příklad jde vidět na obrázcích 8.3 a 8.4. Tady funkce nevyřešila problém viditelného přechodu na hraně, ale výrazně dopomohla ke srovnání barev.

Jsou však případy, kdy tato funkce je spíše na škodu. Takový příklad jde vidět na obrázku 8.5. Oba snímky na panoramatu jsou sice vyfotografovány se stejným nastavením, ale jsou relativně tmavé. Levý snímek však obsahuje poměrně hodně bílé barvy (oblohy ve vrchní části) oproti pravému. Touto plochou se výrazně posunula průměrná hodnota barev levého obrázku směrem nahoru. Proto se úroveň jasu zvýší i u druhého obrázku a výsledek je takový, že pravý snímek je více přeexponovaný než levý. Výsledné panorama tak opět nebude dojem jednotného celku.



Obrázek 8.5: Srovnání úrovní jasu – obrázek vpravo je ve výsledku příliš přexponovaný

8.5 Testování

Aplikace umožňuje na svém vstupu zadat libovolné množství vstupních snímků. Před samotným výpočtem panoramatu, má uživatel možnost měnit vlastnosti detektoru významných bodů. Program zvládá spojovat, jak horizontální tak vertikální panoramata.

Program byl testován na 3 sadách snímků o různém rozlišení. Byly zvoleny i různé druhy předpokládaných panoramat. V tabulce 8.1 je souhrn informací o těchto sadách.

Název sady	Počet snímků	Panorama
Sada 1 – moře	2	horizontální
Sada 2 – přehrada	3	horizontální
Sada 3 – zahrada	3	vertikální

Tabulka 8.1: Testovací sady snímků

Tabulka 8.2 obsahuje naměřené časové hodnoty výsledných panoramat z průběhu testování. Sloupec střed v tabulce znamená, který obrázek v pořadí snímků byl označen za *středový obrázek* (význam pojmu vysvětlen v kapitole 6.4). Pole „downscale“ je nastavení počáteční úrovně rozlišení detektoru významných bodů. Toto rozlišení bylo nastavováno podle větší hodnoty ze dvojice rozměrů šířka a výška. Počet bodů znamená hodnotu požadovaného počtu korespondujících bodů k vypočtení geometrického modelu.

Název	Střed	Šířka [px]	Výška [px]	Downscale	Počet bodů	Čas
Sada 1	1	200	150	200	12	16,1s
Sada 1	2	200	150	200	12	15,1s
Sada 1	1	400	300	400	100	1m16s
Sada 1	2	400	300	400	100	1m14s
Sada 1	1	800	600	800	1000	8m6s
Sada 1	2	800	600	800	1000	8m4s
Sada 2	1	150	200	200	12	25,5s
Sada 2	2	150	200	200	12	25,4s
Sada 2	3	150	200	200	12	26s
Sada 2	1	300	400	400	100	2m38s
Sada 2	2	300	400	400	100	2m40s
Sada 2	3	300	400	400	100	2m41s
Sada 2	1	600	800	800	1000	10m6s
Sada 2	2	600	800	800	1000	10m15s
Sada 2	3	600	800	800	1000	10m1s
Sada 3	1	200	150	200	12	31,8s
Sada 3	2	200	150	200	12	35,6s
Sada 3	3	200	150	200	12	33,5s
Sada 3	1	400	300	400	100	2m31s
Sada 3	2	400	300	400	100	2m33s
Sada 3	3	400	300	400	100	2m30s
Sada 3	1	800	600	800	1000	14m11s
Sada 3	2	800	600	800	1000	13m59s
Sada 3	3	800	600	800	1000	14m5s

Tabulka 8.2: Tabulka naměřených časových hodnot

Všechny sady byly měřeny s rozlišením 200x150px, 400x300px a 800x600. Je patrné, že i při takových malých rozlišení snímků, které jsou mimochodem menší než rozlišení dnešních běžných fotografií (kolem 1600x1200), dosahuje program relativně dlouhých časů. Můžeme konstatovat, že čím větší rozlišení obrázků a větší počet těchto obrázků, tím je delší doba vytváření panorama. Jak se můžeme přesvědčit z grafu na obrázku 8.6, tato doba stoupá v závislosti na počtu obrazových bodů obrázků lineárně.

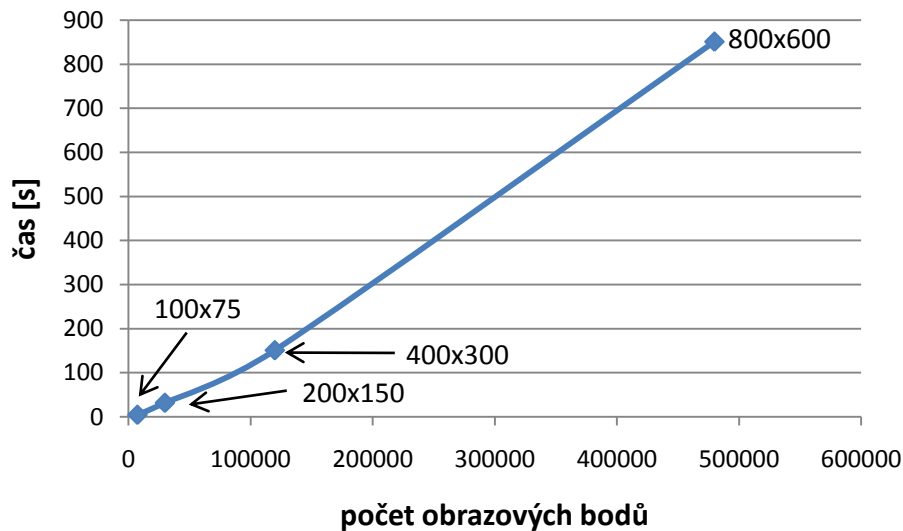
Hodnota „downscale“ byla volena podle rozlišení spojovaných obrázků, neboť při vyšších hodnotách „downscale“ než rozlišení obrázků, se vždy bere v potaz vyšší z rozlišení obrázků. Nižší hodnoty mají sice na detekci vliv, urychlí celkový čas vytváření panorama, avšak počet nalezených korespondenčních bodů není dostatečný, proto se transformační obrázky netransformují do panorama tak, jak se předpokládá. Tento parametr je zde jen pro experimentování.

Další parametr je „počet bodů“ značící požadovaný počet významných bodů k vytvoření modelu. Tato hodnota byla volena podle velikosti rozlišení obrázků a nemá a ani nemůže mít vliv na dobu trvání výpočtu. Program tuto proměnnou využije na konci algoritmu RANSAC, kdy zkontroluje, jestli počet korespondenčních bodů je větší nebo alespoň roven této hodnotě.

Zhodnocení testování

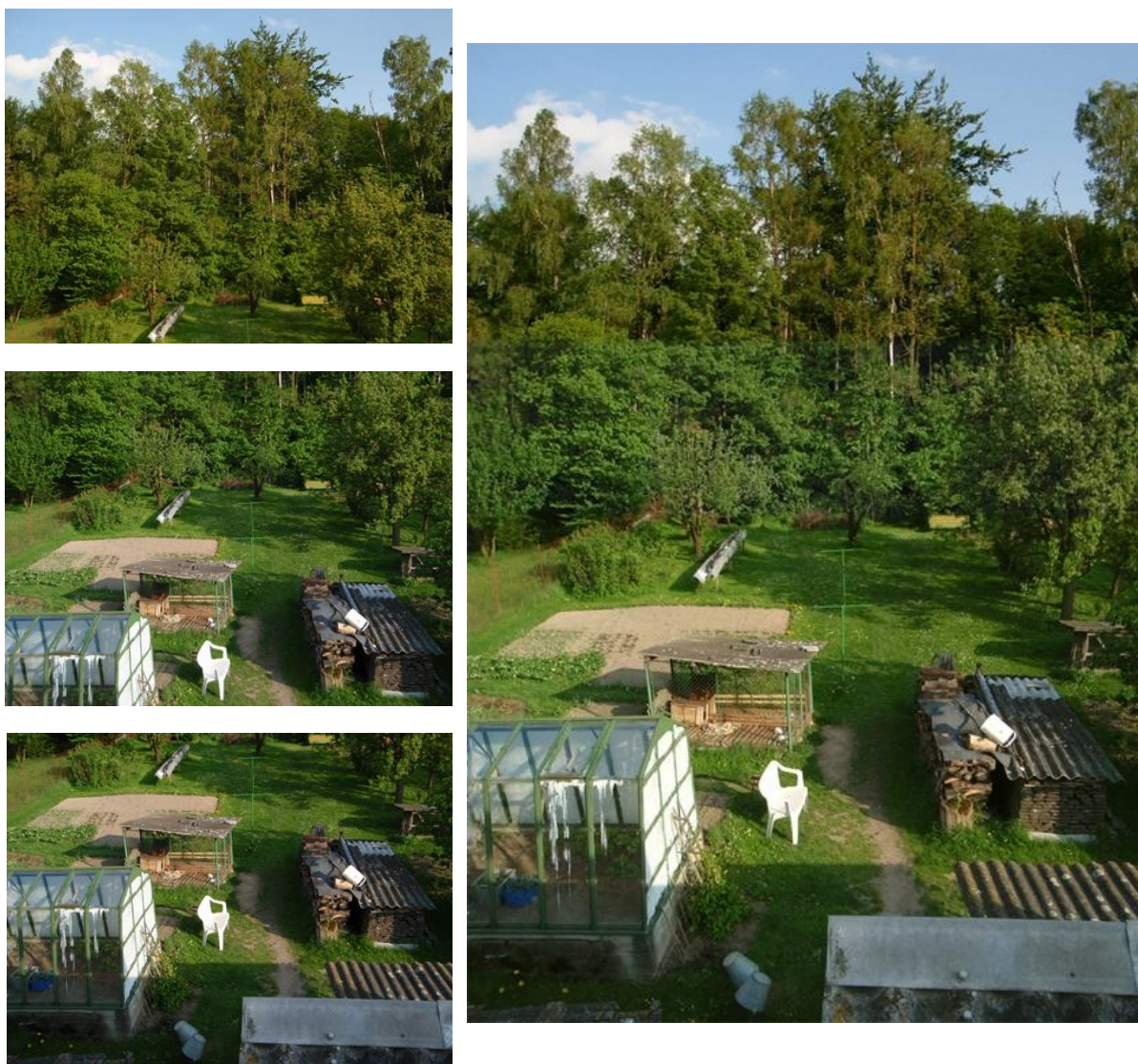
Výsledná panoramata, která aplikace vytvořila, jsou z pohledu umístění transformovaných obrázků uspokojivá. Hodně záleží na kvalitě pořízení vstupních obrázků. Pokud jsou snímky pořízeny podle pravidel vytváření panoramat (popsaných v kapitole 5), program má celkem dobré výsledky. Co se

týče kvality vytvořených přechodů, tak na panoramatech, kde jsou velké jednobarevné plochy (voda, nebe) a ve vstupních obrázcích jsou vyrovnané jasové úrovně, nejsou poznat. V panoramatech, kdy na přechodech je velká četnost nějakých objektů, opět záleží na kvalitě snímků, jak hodně musí program fotografie transformovat. Co se týče časových údajů, tak je nutná optimalizace kódu detektoru významných bodů a vyhledávání korespondence, neboť doba vytvoření panorama je poměrně velká. V tomto hledisku program zatím není schopný konkurovat komerčním programům na vytváření panoramatických snímků.



Obrázek 8.6: Graf časové závislosti na počtu obrazových bodů

Aplikace spojuje poměrně spolehlivě a relativně rychle obrázky s rozlišením do hodnoty 400x300. Od této hodnoty výpočet trvá poměrně dlouho oproti ostatním komerčním programům a není vždy nalezena požadovaná shoda v obrázcích. Ideální hodnota „downscale“ je, jak již bylo zmíněno, hodnota „default“, což značí velikost většího z rozměrů obrázku. Aplikace nalézá shodu dvou obrázků, které mají společnou překryvnou plochu od zhruba 8%, což je celkem pozitivní číslo.



Obrázek 8.7: Vertikální panorama vytvořené aplikací

9 Závěr

Tato diplomová práce se zabývá detekcí významných bodů v obraze. Hlavním tématem je vytvoření aplikace na tvorbu panoramatických fotografií. Aplikace je založena na detekci významných bodů prováděnou algoritmem SIFT a nalezení korespondence těchto bodů, kterou hledá algoritmus RANSAC. V této technické zprávě jsou popsány tyto metody a další techniky, které se používají při vytváření panoramatických snímků. V implementační části je pak probrán vývoj aplikace na tvorbu panoramat.

Výsledná aplikace umožňuje spojovat libovolné množství snímků. Je možné skládat horizontální či vertikální panorama. Pro experimentování s hledáním korespondencí významných bodů je umožněno uživateli měnit několik parametrů detektoru. V aplikaci není použit žádný druh projekce, jedná se o skládání snímků, které vznikne transformacemi obrázků – posunutí, rotace a změna měřítka. Tyto transformované obrázky se „nalepí“ na tzv. středový obrázek, který zůstává netransformovaný a tvoří základ pro ostatní snímky. K lepšímu vzhledu celého panorama je použito několik přídatných technik, jako vyhlazení přechodů mezi jednotlivými obrázky panorama nebo vyrovnání úrovní jasu vstupních obrázků.

Celkový dojem z vytvořených panoramat je podle mého názoru uspokojivý. Z pohledu umístění transformovaných obrázků jsou zde drobné odchylky korespondujících významných bodů, tato chyba však není výrazně viditelná. Tady záleží hodně na kvalitě pořízení vstupních obrázků. Pokud jsou snímky pořízeny podle pravidel vytváření panoramat, program má z tohoto pohledu celkem dobré výsledky. To samé platí o hodnocení kvality vyhlazených přechodů mezi obrázky. Pokud jsou dodržena pravidla pořizování snímků, přechody nejsou na první pohled viditelné. Co se týče časových údajů, aplikace nedosahuje takových výsledků jako její komerční konkurenti. Obrázky o velikosti 30000 obrazových bodů (což odpovídá rozlišení 200x150) zvládá zhruba za stejnou dobu jako komerční aplikace obrázky s 1 920 000 obrazovými body (rozlišení 1600x1200 – běžné rozlišení dnešních fotografií).

Jak již bylo zmíněno, korespondence mezi umístěním transformovaných obrázků a *středového obrázku* má drobné odchylky, což má za následek nepřesnou transformaci. K potlačení tohoto nedostatku by mohlo dopomoci použití geometrických projekcí, které by mapovaly obrázek na geometrické tvary a tím by mohly přispět k transformační korekci. Přidání geometrických projekcí může být jedno z dalších rozšíření aplikace, které by také dopomohlo k reálnějšímu vzhledu panoramat, kdy např. horizonty by byly ve vodorovné poloze a hrany by byly kolmé k horizontům. K reálnému využití aplikace je třeba také výrazně vylepšit proces vyhlazení přechodů mezi snímky. K tomu by mohlo pomoci použití např. gaussového rozložení barevných hodnot obrazových bodů do okolí. K lepšímu vzhledu celého panorama by přispělo přidání možnosti práce s histogramy obrázků a jejich jasové korekce, jako např. ekvalizace histogramu nebo roztažení histogramu. Poslední vylepšení bych viděl ve snížení doby vytváření panorama, která je nyní poměrně vysoká. To má za příčinu vysoký počet korespondenčních bodů v obrázku. Ve standardním obrázku s velikostí 1600x1200 se počet těchto bodů pohybuje řádově v tisících bodů. Ovšem obrázek o velikosti např. 200x175 pixelů má korespondenčních bodů řádově v desítkách. Proto bych viděl jednu z optimalizací v převedení vstupního obrázku na menší rozlišení, tím se stane počet korespondenčních bodů přijatelný vzhledem k časové náročnosti a bude zároveň dostačující pro vytvoření panoramatu. Po nalezení korespondence je samozřejmě potřeba přepočítat pozice bodů z menšího obrázku do většího a vygenerovat panorama s požadovaným rozlišením. Tímto krokem se také můžeme přiblížit kvalitě komerčních konkurentů.

Literatura

- [1] Nowozin, S., libsift - Scale-Invariant Feature Transform implementation [online], 2005, [cit. 26.12.2008], dostupné z URL: <<http://user.cs.tu-berlin.de/~nowozin/libsift/>>.
- [2] Horák, K., Dynamické obrazy [online], [cit. 28.12.2008], dostupné z URL: <www.uamt.feec.vutbr.cz/vision/TEACHING/MPOV/12%20-%20Dynamicke%20obrazy.pdf>.
- [3] Lowe, D., Distinctive Image Features from Scale-Invariant Keypoints [online], 2004, [cit. 29.12.2008], dostupné z URL: <www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.
- [4] Lowe, D., Object Recognition from Local Scale-Invariant Features [online], 1999, [cit. 29.12.2008], dostupné z URL: <www.cs.ubc.ca/~lowe/papers/icc99.pdf>.
- [5] Bílek, P., Významné body v obraze: detekce, korespondence a lokalizace ve 3D [online], 2007, [cit. 3.1.2009], dostupné z URL: <https://dip.felk.cvut.cz/browse/pdfcache/bilekp3_2007bach.pdf>.
- [6] Kaněčka, P., Vyhledání význačných bodů v rastrovém obraze [online], 2007, [cit. 3.1.2009], dostupné z URL: <<http://www.fit.vutbr.cz/study/DP/rpfile.php?id=1354>>.
- [7] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., SURF: speeded up robust features [online], 2006, [cit. 3.1.2009], dostupné z URL: <<http://www.vision.ee.ethz.ch/~surf/>>.
- [8] Wikipedia, Scale-invariant feature transform [online], 2009, [cit. 11. 05. 2009], dostupné z URL: <http://en.wikipedia.org/wiki/Scale-invariant_feature_transform>.
- [9] Pixelcase, webvideophotography [online], 2009, [cit. 12. 05. 2009], dostupné z URL: <<http://pixelcase.com.au/products/>>.
- [10] Psphotodesign, Little Planets [online], 2007, [cit. 12. 05. 2009], dostupné z URL: <<http://psphotodesign.com/?p=7=1>>.
- [11] Panoramio, Raglan Castle 360 Panorama [online], 2009, [cit. 12. 05. 2009], dostupné z URL: <<http://www.panoramio.com/photo/3375782>>.
- [12] Nowozin, S., autopano-sift [online], 2005, [cit. 17.05.2009], dostupné z URL: <<http://user.cs.tu-berlin.de/~nowozin/autopano-sift/>>.
- [13] Wikipedia, RANSAC [online], 2009, [cit. 18. 05. 2009], dostupné z URL: <<http://en.wikipedia.org/wiki/RANSAC>>.
- [14] Navrátil, J., Transformace a RANSAC [online], 2008, [cit. 18. 05. 2009], dostupné z URL: <https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_09_transformace_a_RANSAC.pdf>.

Seznam příloh

Příloha 1. Uživatelská příručka (manuál)

Příloha 2. Výstupy aplikace

Příloha 3. CD

Příloha 1 – Uživatelská příručka (manuál)

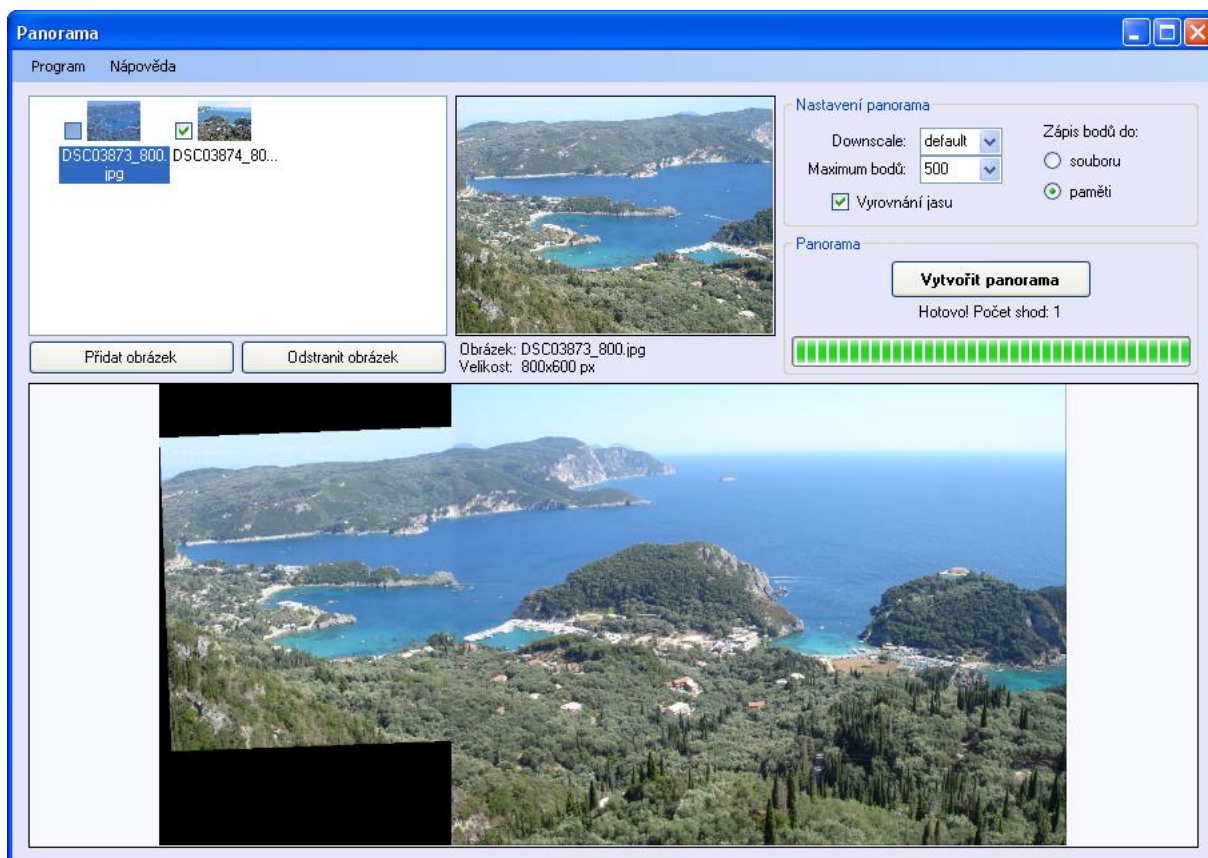
Program je vytvořen pro prostředí Windows jako Windows Application. Má tedy své GUI (graphic user interface - uživatelské rozhraní aplikace), pomocí kterého se program ovládá. Náhled aplikace je zobrazen na obrázku 1. Následuje popis ovládacích prvků aplikace.

Pro načtení obrázků slouží seznam listViewBox, který zobrazuje snímky v malém náhledu s jejich názvem souboru. U každého obrázku je zaškrťovací tlačítko, díky kterému uživatel zvolí středový snímek panoramatu. K ovládání seznamu s obrázky slouží 2 tlačítka – Přidat obrázek, Odstranit obrázek. Uživatel může vybírat obrázky ze seznamu pro jejich větší náhled a zobrazení informací o něm.

Pole s názvem „Nastavení“ slouží převážně k nastavení detektoru významných bodů. Obsahuje dva prvky comboBox. První z nich „Downscale“ nastavuje počáteční úroveň rozlišení obrázku pro scale-space metodu. ComboBox „Maximum bodů“ určuje požadovaný počet bodů geometrického modelu pro vyhledání korespondence. Dále je zde možnost volby ukládání významných bodů s deskriptory do souboru nebo do paměti. Poslední volbou je použití vyrovnání jasových úrovní vstupních obrázků pomocí zaškrťovacího tlačítka „Vyrovnání jasu“.

Samotné vytvoření panoramatu se provádí v poli „Panorama“, které obsahuje tlačítko pro spuštění aplikace a prvky pro zobrazování aktuálního průběhu – progressBar s popiskem.

Pokud program úspěšně vytvoří panoramatickou fotografii, je tato fotografie zobrazena v dominantním pictureBox ve spodní části aplikace. K uložení obrázku postačí kliknout na tento pictureBox a objeví se dialogové okno pro umístění jeho uložení.



Obrázek 1: GUI aplikace

Příloha 2 – Výstupy aplikace

