

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

TVORBA PANORAMATICKÝCH FOTOGRAFIÍ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP DOSPIVA

BRNO 2015



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

TVORBA PANORAMATICKÝCH FOTOGRAFIÍ

PANORAMIC PHOTO CREATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

FILIP DOSPIVA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. KAMIL BEHÚŇ

BRNO 2015

Abstrakt

Tato práce obsahuje srovnání přístupů pro implementaci tvorby panoramatických fotografií. Jsou zde rozebrány dílčí kroky tvorby panoramat a metody přístupu, které jsou pro tyto kroky používané. Je zde vytvořen náhled nad jednotlivými metodami a sepisuje jejich výhody a nevýhody. Součástí je také srovnání těchto metod a zhodnocení jejich použitelnosti.

Abstract

This thesis contains a comparison of approaches to implementation of the panoramic photos. There are analyzed partial steps to create panoramas and access methods that are used for these steps. There is a preview of the methods and schedules their advantages and disadvantages. Also included is a comparison of these methods and evaluation of their usability.

Klíčová slova

panoráma, koláž, panoramatické fotografie, skládání obrázků, klíčové body, detekce klíčových bodů, deskripce klíčových bodů, smíchání obrazu

Keywords

panorama, image mosaic, panoramic photo, image stitch, keypoints, keypoint detection, keypoint description, image blending

Citace

Filip Dospiva: Tvorba panoramatických fotografií, bakalářská práce, Brno, FIT VUT v Brně, 2015

Tvorba panoramatických fotografií

Prohlášení

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

.....

Filip Dospiva
20. května 2015

Poděkování

Velice rád bych poděkoval vedoucímu mé bakalářské práce panu Ing. Kamilovi Behúňovi za jeho odbornou a obětavou pomoc, která mi byla oporou při plnění této práce.

© Filip Dospiva, 2015.

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

1 Úvod	2
2 Tvorba panoramat	3
2.1 Detekce příznaků obrazu	4
2.1.1 Klíčové body	4
2.1.2 Harrisův detektor rohů	4
2.1.3 SIFT	6
2.1.4 SURF	11
2.2 Selektce příznaků obrazu	13
2.2.1 Výběr N nejsilnějších	15
2.2.2 ANMS	15
2.2.3 Selektce pomocí k-d stromu	15
2.3 Spojování příznaků obrazu	17
2.3.1 Best Bin First algoritmus	17
2.3.2 Homografie	17
2.3.3 RANSAC	19
2.3.4 Projekce	19
2.4 Smíchání obrazu	19
2.4.1 Průměr intenzit	20
2.4.2 Prolnutí přechodem	20
2.4.3 Pyramidové smíchání	20
2.4.4 Graph cuts	21
3 Datová sada	24
4 Porovnávání metod pro tvorbu panoramat	26
4.1 Porovnání detektorů klíčových bodů	26
4.2 Porovnání metod pro selekci klíčových bodů	27
4.3 Porovnání deskriptorů klíčových bodů	27
4.4 Porovnání přístupů pro smíchání obrazu	31
5 Závěr	36
A Obsah CD	40

Kapitola 1

Úvod

Ne vždy se dá v objektivu fotoaparátu zabrat vše, co by fotograf chtěl zachytit na jeden snímek a proto tvorba panoramatických fotografií je jednou z nejznámějších a nejžádanějších úprav fotografií, která se při zpracování obrazu využívá. Jde vlastně o spojení více obrázků do jednoho celku, kdy výsledná fotografie vytváří obraz, jehož úhel pohledu nad požadovanou scénérií je větší než nad jednotlivými snímky. Zvětšení úhlu pohledu však není jedinou výhodou, kterou lze u panoramatických fotografií najít. Druhou velkou výhodou je, že tyto obrázky díky slučování jednotlivých fotografií mohou mít velice vysoké rozlišení (až gigapixely), i když fotoaparát, kterým jsou dílčí fotografie pořizovány, toto rozlišení neumožňuje.

Hlavním cílem této práce je zaměřit se na tvorbu panorámat a také na srovnání metod, které se při této tvorbě využívají. Postup tvorby panoramatických obrázků je zde sepsán do jednotlivých kroků. V těchto krocích jsou podrobně rozepsány metody a algoritmy, které danou problematiku řeší. Při porovnávání používaných metod je postupováno stejným způsobem a pro každý krok postupu tvorby jsou metody srovnávány zvlášť.

Kapitola 2

Tvorba panoramat

Algoritmus pro tvorbu panoramatických fotografií lze obecně rozdělit do několika po sobě jdoucích kroků. Jednotlivé přístupy pro tvorbu panoramat se pak liší na základě toho, jaké metody jsou pro jednotlivé kroky použity. Dílčí kroky algoritmu pro tvorbu panoramat jsou:

1. V každém obrázku se naleznou jeho význačné (klíčové) body, které se poté popíší deskriptorem neboli popisem tohoto bodu.
2. Ze všech klíčových bodů se vybere N těch nejlepších a nejvhodnějších pro další řešení problému.
3. Deskriptory význačných bodů se posléze porovnávají se všemi deskriptory význačných bodů v okolních obrázcích. Pokud se naleznou dva deskriptory, které jsou shodné či velmi podobné, tak se tyto dva body mohou označit za korespondující (dále už jen korespondence).
4. Na základě těchto korespondencí se pro sousedící obrázky vypočítá jejich matice homografie, která udává vztah mezi korespondujícími body obrazů (deformaci obrazu).
5. Odfiltrují se špatně určené korespondence. Korespondence mohou být určeny špatně např. u periodicky opakujících se objektech jako jsou stromy v lese nebo okna domů.
6. Obrázky se nakonec spojí do jednoho celku a odstraní se jeho nežádoucí prvky. Těmito prvky mohou být viditelné hrany jednotlivých dílčích obrázků nebo nežádoucí prvky v obraze, které vznikly nepřesným spojením dvou obrázků či díky pohybujícím se objektům. Zde uvedené prvky jsou tedy odstraněny pomocí smíchání dvou sousedících obrazů.

V této kapitole jsou uvedeny jednotlivé dílčí postupy, které jsou použity při tvorbě panoramat. Je zde vytvořen náhled nad problematikou postupu tvorby a blíže rozepisuje některé často používané metody k jejich bližšímu seznámení. Je zde rozebráno, co to jsou klíčové body, příznaky obrazu a jak tyto příznaky spojovat mezi jednotlivými obrazy. Dále je tady uvedeno, jak přiřadit k sobě dva shodující se body a jak rozhodnout zdali k sobě tyto body patří. Je zde také uvedeno, jak deformovat tvar jednotlivých obrazů tak, aby vytvořili požadované panoráma. V neposlední řadě je pak této kapitole uvedeno, jak smíchat dva okolní obrazy za pomoci přechodů, rozmazání a míchání obrazů tak, aby byly odstraněny prvky, které ve výsledném panoramatu jsou nežádoucí.

2.1 Detekce příznaků obrazu

Při tvorbě panoramat se při spojování dvou sousedících obrazů využívá toho, že se spojují pouze určité klíčové (významné) body, které jsou popsány v 2.1.1. Těchto bodů je poznatelně menší počet než pixelů v obrázku, a proto spojování obrazů pouze pomocí klíčových bodů je mnohem vhodnější než spojování pixel po pixelu, protože by to bylo velice náročné na paměť počítače a jeho výkon a především by byl tento přístup naprosto bezúčelný, protože při jakékoliv transformaci jednoho z obrazů by již stejný bod nemusel být nalezen v tom druhém. Jelikož klíčový bod nenese příliš mnoho informací, využívá se proto také bodů v jeho okolí. Oblasti okolo klíčového bodu mohou být různě velké v závislosti na použité metodě. Deskriptory, popisující tyto oblasti nazýváme příznaky obrazu (Features). Tyto deskriptory (popisy) slouží k tomu, aby uchovaly o významných částech obrazu jejich informace. Příznaky obrazu mohou být globální, což by znamenalo, že jeden obrázek by byl popsán pouze jako jeden klíčový bod, což není vhodné pro tvorbu panoramat, proto se tedy vždy využívá příznaků lokálních, kterých je v obraze nalezeno větší množství, a proto se dají snadno spojovat mezi sousedními obrázky.

Metod pro samotné získávání příznaků obrazu je spousta a jejich průběh lze rozdělit do dvou hlavních fází. V první fázi se vyhledají klíčové body obrazu pomocí detektoru významných bodů jako je třeba Harrisův detektor rohů, který je popsán v 2.1.2. Druhou fází je pak sestavení deskriptoru z okolí klíčového bodu pomocí deskriptoru významných bodů jako jsou například metody SIFT (2.1.3) nebo SURF (2.1.4).



Obrázek 2.1: Klíčové body fotografie nalezené pomocí SURF.

2.1.1 Klíčové body

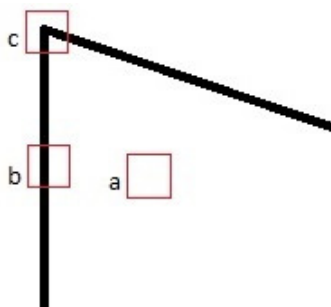
Jak již bylo uvedeno tak prvním krokem při tvorbě panoramat je detekce klíčových bodů v obraze. Vlastnostmi těchto bodů jsou snadná vyhledatelnost, lokalizace a opakované nalezení shodných struktur klíčových bodů i po geometrických a fotometrických změnách. Další jejich vlastností je, že struktura v obraze okolo klíčového bodu je bohatá na informace, které se využívají pro další zpracování. Klíčovými body jsou zpravidla rohy, hrany, skvrny (bloby) nebo T-spoje v obraze [3], jelikož tyto oblasti jsou téměř neměnné vůči změnám měřítka, otočení, atd [1]. Každá metoda pro práci s klíčovými body pak většinou využívá pouze jeden druh bodů. Např. Harrisův přístup (2.1.2) využívá pro svou práci rohy objektů

v obraze, zatímco detektor používaný pro popis oblasti pomocí SIFT deskriptoru pracuje s bloby v obraze (2.1.3). Z výše uvedeného lze usoudit, že tyto body nejsou náhodné, a proto se musí v obraze nějak vyhledat. K tomuto vyhledání slouží detektory klíčových bodů v obraze.

Na obrázku 2.1 jsou vyobrazeny všechny klíčové body nalezené pomocí Fast-Hessian detektoru, který se používá při popisu klíčových bodů pomocí SURF deskriptorů. Je zřejmé, že body byly nalezeny především v centrální části obrázku, což pro tvorbu panoramat není nejvhodnější, jelikož pro úspěšné spojení dvou sousedních obrázků je potřeba mít dostatek vhodných klíčových bodů po stranách obrázku. Proto se využívá selekce vhodných bodů, která je popsána v sekci 2.2.

2.1.2 Harrisův detektor rohů

Chris Harris a Mike Stephens poprvé sepsali a publikovali tento detektor roku 1988 [16]. Byl vytvořen tak, aby odstranil nedostatky staršího detektoru Moravec [23], který byl sice jednoduchý a rychlý, avšak náchylný na šum a byl nestabilní. Pro tvorbu panoramat je jedním z často užívaných detektorů rohů v obraze ([25], [9]) kvůli své invarianci proti otočení, posunu a drobné transformaci. Snaží se nalézt lokální maxima změn intenzit v obraze. Metoda je založená na autokorelační funkci, která reflektuje lokální změny intenzity pixelů v obraze. Základní myšlenkou, jak snadno rozpoznat klíčový bod, je dívat se na bod skrze malé okénko. Posouvání tohoto okénka v jakémkoliv směru by pak mělo indikovat velké změny intenzit barev v jednotlivých pixelech. V případě Harrisova detektoru můžeme pohybovat okénkem v osmi směrech po 45°.



Obrázek 2.2: Příklady umístění vzorkovacího okénka u Harrisova detektoru.

Na obrázku 2.2 jsou vyobrazeny 3 případy vzorkovacího okénka. a) V případě, kdy se okénko nachází na "plochách", tak se neprojevují žádné změny intenzit při jakémkoliv posouvání vzorkovacího okénka. b) V případě, kdy se okénko nachází na "hranách" objektů, tak se neprojevují žádné změny intenzit při posouvání vzorkovacího okénka ve směru hrany, avšak projevují se velké změny intenzit ve směru kolmém na hranu objektu. c) V případě, kde se okénko nachází v "rohu" objektu, tak se projevují výrazné změny intenzit při posouvání vzorkovacího okénka v jakémkoliv směru.

Aby byl detektor invariantní vůči šumu, na který je velmi náchylný, může využívat některou z vyhlazovacích funkcí, která bude nad zorným okénkem. Příkladem může být tzv. Gaussova funkce daná výrazem:

$$g(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \quad (2.1)$$

kde σ je směrodatná odchylka určující vyhlazenost obrazu. Autokorelační funkce změn intenzit při posunu $[u, v]$ je pak vypočítána jako:

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2, \quad (2.2)$$

kde $w(x, y)$ je funkce okénka (Gaussova funkce) a I jsou intenzity posunutého a aktuálního bodu. Ekvivalentem pro malé posuny $[u, v]$ je bilineární aproximace:

$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}, \quad (2.3)$$

kde matice M je dána výrazem:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad (2.4)$$

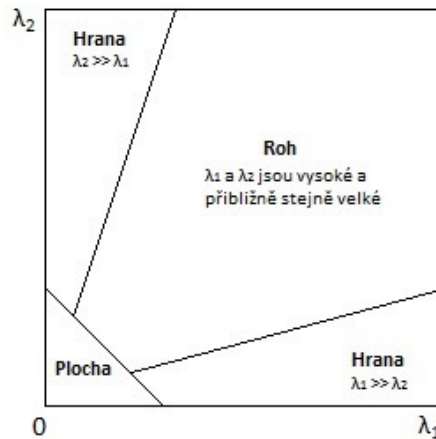
kde I_x a I_y jsou deriváty intenzity pixelu v x a y směru. Výsledná míra odpovědnosti rohů (angl. measure of corner response) pro každý pixel je pak spočítána jako:

$$R = \det(M) - k * (\text{trace}(M))^2 \quad (2.5)$$

$$\det(M) = \lambda_1 \lambda_2 \quad (2.6)$$

$$\text{trace}(M) = \lambda_1 + \lambda_2, \quad (2.7)$$

kde $\det(M)$ je determinant odpovídající součinu dvou vlastních hodnot matice M a $\text{trace}(M)$ je suma hodnot na hlavní diagonále. Přidružené vlastní vektory tvoří dva navzájem kolmé směry největší a nejmenší změny intenzity v obraze λ_1 a λ_2 . k je zde jako faktor citlivosti. Nakonec se vyberou pouze ty body, jejichž hodnota R je vyšší než zvolený práh (threshold).



Obrázek 2.3: Klasifikace obrazových bodů pomocí vlastních hodnot λ_1, λ_2 matice M .

Z obrázku 2.3 vyplývá, že pokud u zkoumaného klíčového bodu je jedna vlastní hodnota λ matice M mnohem větší než jeho druhá vlastní hodnota, bude se jednat o hranu v obraze. V případě, kdy budou obě dvě vlastní hodnoty velmi nízké nebo téměř rovny nule, bude se jednat o bod nalezený v plochách, kde není výrazná změna intenzity v okolních bodech. V poslední řadě, pokud budou obě dvě vlastní hodnoty vysoké nebo jsou si podobné, lze tento klíčový bod prohlásit za roh v obraze, jelikož jsou indikovány změny intenzit v minimálně dvou směrech.

2.1.3 SIFT

Jedením z často využívaných deskriptorů klíčových bodů ([8], [1]) nejen při tvorbě panoramat je Scale-invariant feature transform algoritmus, který vytvořil a publikoval David Lowe v roce 1999 [1]. Využívá detektoru, který pracuje se scale-space a "bez-měřítkovým" popisem obrazu. Tento detektor oproti Harrisovu detektoru pracuje se skvrnami v obraze a umí mnohem lépe najít klíčové body obrazu. Popsání klíčového bodu SIFT deskriptorem pak zaručí, že body jsou invariantní vůči změnám měřítka (z angl. scale), rotaci, posunu, změně úhlu pohledu, změnám kontrastu i šumu v obraze, na druhou stranu je pro svou složitější konstrukci náročnější na potřebné prostředky.

Hlavní myšlenkou, jak vylepšit uchování klíčových bodů získaných detektorem, bylo přiřazení prvku orientace. Tato orientace je vypočítávána z rozmazaného přechodu intenzit v okolí klíčového bodu. Nalezený klíčový bod se posléze popisuje deskriptorem nebo-li vektorem jednotlivých orientací v okolí tohoto bodu (feature vector), obvykle o velikosti 128 hodnot.

Obecný princip SIFT detektoru a deskriptoru je:

1. Detektor

- Detekce scale-space extrémů v obraze
- Lokalizace klíčových bodů a odfiltrování špatných bodů

2. Deskriptor

- Přidělení orientace
- Vytvoření deskriptoru pomocí histogramu orientací

Detekce scale-space extrémů v obraze

Cílem detekce scale-space extrémů je identifikovat lokace a jejich škály (měřítka), které mohou být opakovaně přiřazeny pod různými pohledy na stejnou scénérii. Scale-space je reprezentace dat obrázku nezávislá na měřítku, kterou popsal ve své práci Tony Lindeberg [20]. Metoda tedy vyhledává stabilní příznaky obrazu napříč vícero škálami za použití spojité funkce škálovatelnosti. Nejlépe se osvědčila LoG (Laplacian of Gaussian) [21], kde se jako jádro užívá Gaussova funkce $G(x, y, \sigma)$.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.8)$$

Scale-space obrázku je tedy funkce, která je vytvořena z konvoluce Gaussova jádra (v různých škálách) se vstupním obrazem. Detektor je založen na výpočtu determinantu Hessovy matice o tvaru:

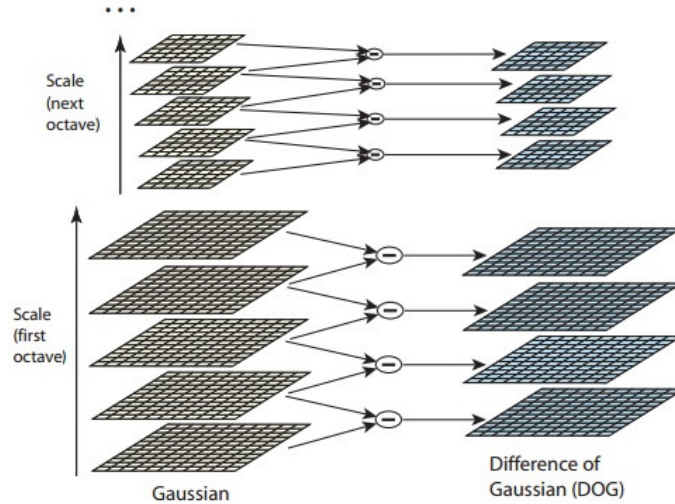
$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (2.9)$$

kde \mathbf{x} představuje bod ve vstupním obraze I zadaný pomocí x, y a $L(\mathbf{x}, \sigma)$ jsou konvoluce druhé derivace Gaussovy funkce $\frac{\partial^2}{\partial x^2} G(x, y, \sigma)$ se vstupním obrázkem I . LoG je pak součtem prvků hlavní diagonály matice H .

$$\nabla^2 L(x, y, \sigma) = \sigma \left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \right) = \sigma \nabla^2 (G(x, y, \sigma) * I(x, y)) \quad (2.10)$$

LoG sice dosahuje nejlepších výsledků při získávání klíčových bodů, ale je však velmi výpočetně náročná a její zpracování trvá příliš mnoho času, proto se pro efektivnější získávání klíčových bodů používá její aproximace DoG (Difference of Gaussians)[1]. Konvoluce rozdílu Gaussových funkcí $D(x, y, \sigma)$ se vstupním obrázkem je vyobrazena na obrázku 2.4 a je vypočítána z rozdílu dvou blízkých měřítek oddělených konstantním multiplikačním faktorem k :

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2.11)$$



Obrázek 2.4: Konstrukce scale-space za pomoci DoG (převzato z [21]).

Jak lze vidět na obrázku 2.4, tak pro každou oktávu scale-space se počáteční obraz opakovaně konvoluje gaussovským rozostřením aby vytvořil sadu scale-space obrázků (vlevo). Sousedící Gaussovy obrázky se posléze od sebe odečtou a vytvoří tak DoG obrázky (vpravo). S každou další oktávou jsou Gaussovy obrázky převzorkovány a proces se opakuje.

DoG funkce umožňuje blízkou aproximaci k LoG, $\sigma^2 \nabla^2 G$. Lindeberg ukazuje, že normalizace Laplaciánu s faktorem σ^2 je potřebná pro skutečnou měřítkovou invarianci. Detailními experimenty a porovnáváním prokázal Mikołajczyk [22], že maxima a minima z $\sigma^2 \nabla^2 G$ produkují nejvíce stabilní příznaky obrazu ve srovnání s jinými možnými obrazovými funkcemi, jako jsou přechod, Harrisova rohová nebo Hessianová funkce. Vztah mezi $D(x, y, \sigma)$ a $\sigma^2 \nabla^2 G$ může být pochopen z difúzní rovnice:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G. \quad (2.12)$$

Z této rovnice můžeme vidět, že $\nabla^2 G$ může být vypočítána z konečného rozdílu aproximace $\frac{\partial G}{\partial \sigma}$ pomocí rozdílu dvou sousedících měřítek $k\sigma$ a σ :

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.13)$$

a z toho vyplývá že,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G. \quad (2.14)$$

Tato rovnice tedy ukazuje, že když DoG funkce má škálovatelnou diferenci za pomoci konstantního faktoru, který již zahrnuje σ^2 měřítkovou normalizaci potřebnou k měřítkově-invariantnímu Laplacianu. Faktor $(k - 1)$ v rovnici je konstantní napříč všemi měřítky, a proto nemá žádný vliv na lokaci extrémů.

Pokud již je vytvořený scale-space, tak pro detekci lokálních minim a maxim z $D(x, y, \sigma)$ je každý vzorkový bod porovnán s jeho osmi okolními sousedy, s devíti sousedy v nejbližším menším a větším měřítku, to dělá dohromady 26 sousedů okolo každého bodu. Z toho lze usoudit, že nelze hledat lokální maxima v krajních vrstvách oktávy, jelikož body v těchto vrstvách nemají dostatečný počet svých sousedů. Tento bod je zvolen pro další zpracování pouze tehdy, pokud je větší nebo menší než všichni jeho sousedé, proto se zapamatují jeho souřadnice ve scale-space.

Lokalizace klíčových bodů a filtrování

Jakmile byl klíčový bod nalezen porovnáním se svými sousedy, dalším postupem je provést detailní lokalizaci do okolních dat pro přesnou lokaci, měřítko a poměr hlavních zakřiveních. Tato informace umožní, aby body s nízkým kontrastem nebo špatně lokalizované body podél hran mohly být odfiltrovány. Prvotní implementace tohoto přístupu [1] byla příliš primitivní, a proto Brown v roce 2002 vyvinul metodu pro umístování 3D kvadratické funkce do lokálních bodů, aby určila interpolovanou polohu. Tímto dosáhl podstatného zlepšení při spojování a stabilitě. Jeho přístup využívá Taylorův rozvoj nad $D(x, y, \sigma)$:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \quad (2.15)$$

kde D a jeho deriváty jsou vyhodnocovány v místě zkoumaného klíčového bodu o souřadnicích v scale-space $\mathbf{x} = (x, y, \sigma)^T$. Lokace extrému $\hat{\mathbf{x}}$ je určena zderivováním funkce dle \mathbf{x} a položením rovno nule. Přesná pozice extrému vůči zkoumanému bodu je určena offsetem:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}. \quad (2.16)$$

Pokud je v jakémkoliv směru $\hat{\mathbf{x}} > 0.5$, tak extrém leží blíže k sousednímu bodu než k aktuálnímu, proto se tento soused vymění za klíčový bod a začne se znovu aproximovat pomocí Taylorova rozvoje. Finální offset $\hat{\mathbf{x}}$ je přidán k lokaci zkoumaného bodu.

Pro odstranění nestabilních extrémů s malým kontrastem, se využívá funkční hodnoty $D(\mathbf{x})$, kterou lze získat jako:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}}. \quad (2.17)$$

Absolutní hodnota $|D(\hat{\mathbf{x}})|$ je pak porovnána s prahem, který slabé a nevhodné klíčové body odstraní.

Tato filtrace klíčových bodů s nízkým kontrastem však pro stabilitu celé metody nestačí, a proto se musí odfiltrovat i další klíčové body. DoG funkce může naleznout také klíčové body na hranách nebo na výrazných čarách v obraze, ty však pro SIFT nejsou vhodné a musí se tedy odstranit. Takovýto špatně zvolený klíčový bod má velké hlavní zakřivení, které je kolmé na směr hrany a velmi malé zakřivení podél hrany. Tyto hlavní zakřivení lze spočítat z Hessovy matice, \mathbf{H} , vypočtena v lokaci a měřítku klíčového bodu:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \quad (2.18)$$

Hlavní zakřivení z D je přímo úměrné vlastním hodnotám matice \mathbf{H} . Výpočet vlastních hodnot je výpočetně náročný, tak tomu bylo např. u Harrisova detektoru, proto se zde počítá pouze s jejich poměrem. Nechť \mathbf{H} má své vlastní hodnoty α, β , pak můžeme spočítat součet vlastních hodnot na hlavní diagonále matice $Tr(\mathbf{H})$ a jejího determinantu $Det(\mathbf{H})$:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta, \quad (2.19)$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \quad (2.20)$$

Nechť r je poměr mezi větší a menší vlastní hodnotou \mathbf{H} tak, že $\alpha = r\beta$, potom:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}, \quad (2.21)$$

z čehož vyplývá, že závisí pouze na poměru mezi vlastními hodnotami než na jejich samotné hodnotě. Pokud je hodnota $\frac{(r+1)^2}{r}$ minimální, znamená to, že vlastní hodnoty matice jsou téměř stejné, naopak se tato hodnota zvětšuje s rozdílem mezi vlastními hodnotami. Pro samotnou filtraci už stačí jen určit, zda je poměr hlavních zakřivení menší než určitá prahová hodnota r (Lowe testy došel ke vhodné hodnotě $r = 10$), proto musí platit:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r + 1)^2}{r}. \quad (2.22)$$

Přidělení orientace

Přidělování orientace ke každému klíčovému bodu umožňuje, aby byly tyto body invariantní vůči rotaci obrazu. Díky měřítku klíčového bodu se vybere obrázek, na který byl použit Gaussův filtr L s co nejhodnějším měřítkem, což pro další počítání zajistí nezávislost na měřítku. Nejprve se pro každý obrazový vzorek $L(x, y)$ v jeho měřítku přepočítá za pomoci difference pixelů velikost jejich gradientu $m(x, y)$ a jejich orientace $\theta(x, y)$:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}, \quad (2.23)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}\right). \quad (2.24)$$

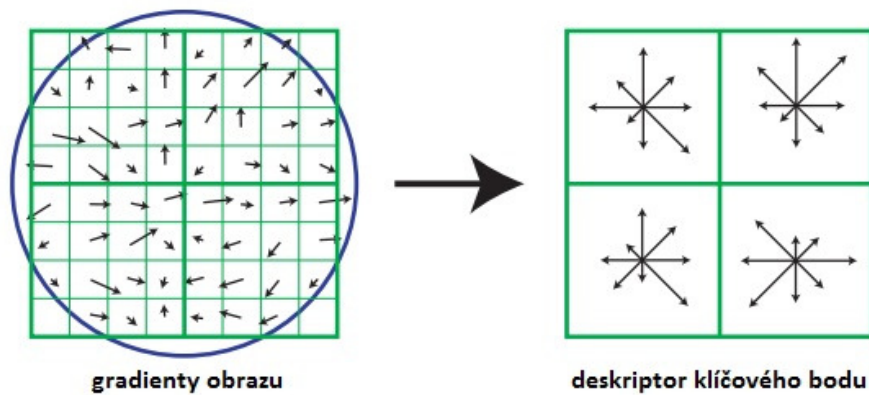
Potom se vytvoří histogram orientací z orientací gradientů jednotlivých vzorkovaných bodů v okolí klíčového bodu. Tento histogram má 36 zásobníků pokrývajících rozsah orientací 360° , každý po 10° . Každý prvek, který je vkládaný do histogramu je vážen hodnotou jeho gradientu a Gaussovsky vážen v kruhovém okénku s hodnotou σ , která je 1.5 x větší, než měřítko klíčového bodu.

Vrcholy v histogramu orientací udávají hlavní směry lokálních gradientů. Nejvyšší vrchol pak v histogramu určuje orientaci klíčového bodu. Pokud je některý vrchol alespoň z 80 % tak velký jako ten nejvyšší, vytvoří se z něj také klíčový bod s touto orientací. Proto, pro lokace s více vrcholy histogramu stejné velikosti, bude vytvořeno více klíčových bodů na stejném místě, stejného měřítko, ale různé orientace. Toto se však stává pouze u okolo 15 % klíčových bodů, avšak i tak se výrazně zvyšuje stabilita při spojování klíčových bodů napříč obrazy. Nakonec se 3-mi hodnotami histogramu, které jsou nejbližší k vrcholu, proloží parabola, která interpoluje pozici vrcholu pro lepší přesnost.

Vytvoření deskriptoru

Po předchozích řešeních má již každý klíčový bod přiřazenou jeho lokaci, měřítko a orientaci. Pomocí těchto parametrů už lze utvořit opakovaně získatelný 2-D koordinační systém, pomocí kterého se může popsat klíčový bod, a tím se také zaručí invariance vůči těmto parametrům. Finálním krokem této metody je tedy popsat každý klíčový bod deskriptorem jeho okolí.

Jedním z možných přístupů je použití intenzit obrazu v okolí klíčového bodu na příslušném měřítku. Pro následné jejich spojování se pak využívá normalizované korelační míry. Tento přístup není však vhodný, jelikož je velice náchylný na afinní transformace v obraze, 3-D změny úhlu pohledu nebo nerigidní deformace objektů v obraze. S lepším přístupem, který tyto nedostatky řešil přišli Edelman, Intrator a Poggio roku 1997 [14]. Jimi navržená reprezentace je založena na modelu biologického vidění. Lowe se pak ve své práci [21] inspiroval tímto přístupem, avšak využívá jiné výpočetní mechanismy.



Obrázek 2.5: SIFT deskriptor znázorněn pomocí orientací jednoho klíčového bodu (převzato z [21]).

Jak vypadá deskriptor klíčového bodu vidíme na obrázku 2.5. Nejdříve jsou u bodů v okolí klíčového bodu určeny míry gradientů a jejich orientace, za využití měřítka klíčového bodu, které určí úroveň Gaussovsky transformovaného obrazu. Pro zachování invariance vůči otočení jsou souřadný systém a orientace gradientů pootočený do směru orientace celého klíčového bodu. Pro zvýšení efektivity jsou gradienty přepočítány pro veškeré úrovně scale-space stejně tak, jako v předchozí sekci. Tyto přepočtené gradienty jsou pak na obrázku 2.5 znázorněny jako malé šipky v celém okolí klíčového bodu (vlevo).

Poté se využije Gaussova vázící funkce s hodnotou σ odpovídající 1/2 velikosti okénka nad klíčovým bodem (znázorněno kruhem vlevo). Tato funkce se využívá kvůli tomu, aby se zamezilo náhlým změnám v deskriptoru při malých změnách v pozici okénka a také proto, aby na gradienty, které jsou blíže ke středu deskriptoru, byl kladen větší důraz než na ty, které jsou od středu dál.

Samotný deskriptor klíčového bodu je pak vyobrazen na obrázku 2.5 vpravo. Pro umožnění výrazných posunů v pozicích gradientů se vytvoří histogramy orientací nad 4 x 4 vzorky v okolí. Každý histogram má osm orientací s velikostmi šipek, jejichž délka závisí na vstupních gradientech. Vzorek gradientu může být posunut až o 4 vzorkovací pozice a stále bude náležet stejnému histogramu.

Je důležité zamezit hraničním změnám, díky kterým se deskriptor náhle změní, jako je přesun vzorku gradientu z jednoho histogramu do druhého při nepřesné lokalizaci klíčových

bodů. Z tohoto důvodu se využívá trilineární interpolace, která distribuuje hodnotu každého vzorku gradientu do zásobníků sousedních histogramů.

Deskriptor je vytvořen jako vektor, který obsahuje hodnoty ze zásobníků všech histogramů orientací, které korespondují s délkami šipek na pravé straně obrázku 2.5. Na obrázku je tedy znázorněno pole 2 x 2 histogramů orientací, avšak nejlepších výsledků dosahuje pole o velikosti 4 x 4 s osmi zásobníky orientací. Tento počet histogramů orientací pak vytváří vektor 128 hodnot (4 x 4 x 8) pro každý klíčový bod.

Nakonec je tento vektor vlastností modifikován tak, aby redukoval efekty světelných změn v obraze. To znamená, že se nejdříve normalizuje vektor na jednotkovou délku. Změna kontrastu obrazu, ve které každá hodnota pixelu je vynásobena konstantou způsobí, že každá hodnota gradientu bude také vynásobena touto konstantou, a proto bude tato změna vyrušena vektorovou normalizací. Změna jasu v obraze, u které je ke každému pixelu obrazu přičtena stejná konstanta, nebude mít na hodnoty gradientů, ta se totiž vyruší při diferencii pixelů. Tímto je deskriptor invariantní vůči světelným změnám v obraze. Pro odstranění také nelineárních světelných změn v obraze se redukuje vliv větších hodnot gradientu užitím prahu na hodnoty v jednotkovém vektoru tak, aby žádná hodnota nebyla větší 0.2, posléze se musí opět vektor normalizovat na jednotkovou délku.

2.1.4 SURF

Při tvorbě panoramat je další z používaných popisovacích přístupů metoda Speeded-Up Robust Features ([17], [18]), která popisuje jednotlivé příznaky obrazu pomocí deskriptorů. Bay, Tuytelaars a Gool vytvořili a sepsali tuto metodu v roce 2006 [3]. Cílem při jejím vytváření byla především rychlost, jíž budou data zpracovávána při zachování vysoké kvality ve výsledcích jako u jiných metod. Výsledkem vznikla metoda tak rychlá, že může být využitelná i při real-time zpracování. V mnohém se tato metoda inspirovala výše uvedenou metodou SIFT 2.1.3, kterou zrychlila a zdokonalila.

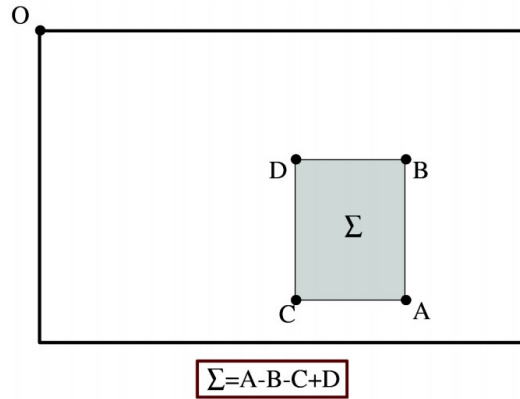
Při popisu klíčových bodů se klade velký důraz na míru invariance, protože se předpokládají geometrické a fotometrické změny v obraze, které jsou způsobeny změnou úhlu pohledu na scénérii. Proto je u SURF velmi důležitá invariance vůči změnám měřítka a rotaci obrazu, na kterých si metoda velice zakládá. Jako méně důležité při invarianci jsou pak brány účinky způsobené zkosením obrazu, anizotropními změnami měřítka a perspektivními změnami v obraze.

SURF pro detekování klíčových bodů využívá detektoru Fast-Hessian, který pro výrazné zvýšení rychlosti využívá integrálního obrazu. Pomocí tohoto obrazu je umožněno získat údaj o intenzitě oblasti obrázku v konstantním čase s potřebou znát jen krajní body oblasti. Poté je takovému bodu přiřazena jeho orientace a na závěr je tento bod popsán vektorem vlastností, který je na rozdíl od metody SIFT mnohem menší.

SURF deskriptory jsou velice podobné deskriptorům SIFT (2.1.3). Jsou založené na stejných vlastnostech, avšak SURF jsou mnohem komplexnější. Prvním krokem při tvorbě deskriptoru je určení orientace, která je založená na informacích získaných z kruhové oblasti okolo klíčového bodu. Poté se sestrojí čtvercová oblast okolo klíčového bodu, která je orientována v získaném směru. Nakonec se z této oblasti extrahuje SURF deskriptor.

Integrální obraz

Integrální obraz [2] se používá pro rychlé výpočty v obdélníkových konvolučních filtrech. Vstup integrálního obrazu $I_{\Sigma}(\mathbf{x})$ v bodě $\mathbf{x} = (x, y)^T$ představuje suma všech pixelů ve



Obrázek 2.6: Integrální obraz užívaný ve Fast-Hessian detektoru (převzato z [2]).

vstupním obrazu I uvnitř obdélníkové oblasti ohraničené levým horním bodem obrazu a bodem \mathbf{x} :

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \quad (2.25)$$

Když už byl integrální obraz jednou vypočítán, pak pro vypočtení sumy intenzit v jakkoliv velké obdélníkové oblasti postačí pouhé tři jednoduché sčítací operace, jak lze vidět na obrázku 2.6. Tímto je zaručeno, že doba výpočtů takovýchto oblastí je nezávislá na jejich velikosti, což je důležité u přístupů, kdy se používají velké velikosti filtrů.

Fast-Hessian detektor

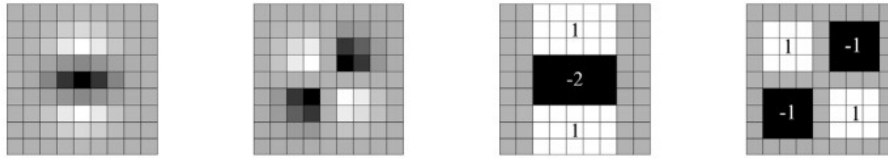
Fast-Hessian detektor využívá Hessovy matice především z důvodu jeho vysokého výkonu ve výpočetním čase a přesnosti. Oproti detektoru využívaného u SIFT se však místo rozdílných měřítek určujících lokaci a měřítko klíčového bodu spoléhá na determinant Hessovy matice. Je-li dán bod $\mathbf{x} = (x, y)$ v obraze I , pak Hessova matice $H(\mathbf{x}, \sigma)$ v bodě \mathbf{x} a měřítku σ je definována jako:

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}, \quad (2.26)$$

kde $L_{xx}(\mathbf{x}, \sigma)$ je konvoluce Gaussova jádra druhé derivace $\frac{\partial^2}{\partial x^2} G(\sigma)$ s obrazem I v bodě \mathbf{x} , obdobně platí také pro $L_{xy}(\mathbf{x}, \sigma)$ a $L_{yy}(\mathbf{x}, \sigma)$.

Cílem bylo zjednodušit přístup, kdy Lowe používal LoG [1], a proto se zde používá ještě větší aproximace a to za pomoci obdélníkových filtrů. Gaussovy funkce jsou sice optimální pro scale-space analýzu, ale nejsou nejvhodnější pro výslednou rychlost zpracování, a proto musí být nejprve diskretizovány a ořezány, jako lze vidět na obrázku 2.7. Tato aproximace druhé derivace Gaussovy funkce může být vypočtena velice rychle za použití integrálních obrazů (2.1.4) a to nezávisle na velikosti obrazu.

Na obrázku 2.7 jsou znázorněny obdélníkové filtry o velikosti 9×9 , které jsou aproximací druhé derivace Gaussovy funkce s $\sigma = 1.2$. Váhy využívané v obdélníkových oblastech jsou pro výpočetní efektivitu uchovávané jednoduše. Výsledky u takovýchto filtrů pak musejí být normalizovány s ohledem na velikost maskované oblasti. Pro tuto měřítkovou normalizaci se používá konstantní Frobeniova norma.



Obrázek 2.7: Konvoluční jádra Fast-Hessian detektoru (převzato z [3]). Bráno zleva lze vidět diskretizované a oříznuté druhé derivace Gaussovy funkce v y a xy směru, a poté jejich aproximace použitím obdélníkových filtrů. Šedé části obrázku jsou rovny nule.

Scale-space je zde implementováno pomocí obrazových pyramid. Obrázek je opakovaně Gaussem rozostřen a následně pod-samplován, aby se dosáhlo vyšších vrstev pyramidy. Díky použití obdélníkových filtrů a integrovaných obrazů se nemusí iterativně používat stejný filtr na výstup již dříve filtrované vrstvy, ale můžeme použít filtr jakékoliv velikosti se stejnou rychlostí přímo na původní obrázek (dokonce může pracovat i paralelně). Proto je scale-space analyzován raději zvětšováním měřítka velikosti filtru než iterativním redukováním velikosti obrazu.

Výstup po použití výše uvedených filtrů je považován za počáteční vrstvu scale-space. Následující vrstvy jsou pak získány postupně filtrováním obrazu s většími maskovanými oblastmi. U větších měřítek by se krok mezi po sobě následujícími velikostmi filtrů měl zvětšovat odpovídajícím způsobem, proto pro každou novou oktávu se nárůst velikost filtru zdvojnásobí. Zároveň se mohou zdvojnásobit vzorkovací intervaly pro extrakci klíčových bodů. Poměr uspořádání filtrů zůstává po změně měřítka konstantní, a proto se aproximované Gaussovy derivace škálují odpovídajícím způsobem.

Poté co je vytvořeno scale-space mohou být lokalizovány klíčové body napříč všemi měřítky. Naleznou se lokální maxima a jsou srovnána s jejich 26 sousedy v jejich okolí stejně jako u SIFT.

Přidělení orientace

Proto, aby byla nalezené klíčové body invariantní vůči rotaci obrazu, musí jim být přiřazena orientace. Nejprve se vypočítají Haar-wavelet hodnoty [2] v x a y směru uvnitř kruhového okolí o poloměru $6s$ okolo klíčového bodu, kde s je měřítko v kterém byl klíčový bod detekován. U větších měřítek se opět využívá integrálních obrazů pro zvýšení rychlosti, což zaručuje použití pouhých šesti operací pro výpočet hodnot v jakémkoliv měřítku. Jelikož je vzorkovací krok získávání gradientů z obrazu závislý na měřítku, proto je nastaven na hodnotu s . Velikost používaného Haar-wavelet jádra je nastavena na $4s$. Získané hodnoty se posléze váží Gaussovou funkcí s hodnotou $\sigma = 2.5s$, která je vycentrována do klíčového bodu.

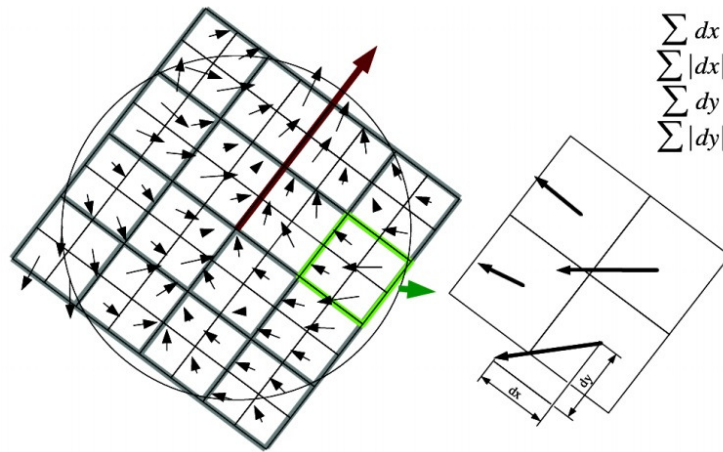
Výsledná orientace je pak určena na základě sumy všech hodnot uvnitř posuvného okénka pokrývajícího úhel $\frac{\pi}{3}$. Hodnoty ve směru x a y jsou sumarizovány. Dva sumarizované výsledky pak vytváří nový vektor. Ze všech získaných vektorů se vybere ten nejdelší a jeho orientace pak určuje orientaci celého klíčového bodu.

Tvorba deskriptoru

Pro vytvoření deskriptoru je prvním krokem sestavení čtvercové oblasti vycentrované okolo klíčového bodu, která je orientovaná ve směru získaného v předchozím kroku. Tato oblast

má délku strany $2s$ a celá tato oblast je rozdělena na 16 menších čtvercových podoblastí. Tyto podoblasti pak uchovávají důležité informace z okolí klíčového bodu. Pro názornost jsou všechny tyto oblasti vyobrazeny na obrázku 2.8. V každé podoblasti je spočítáno několik jednoduchých vlastností pro pár pravidelně rozmístěných bodů. Tyto vlastnosti jsou označeny jako d_x pro Haar-wavelet hodnotu v horizontálním směru a d_y pro Haar-wavelet hodnotu ve vertikálním směru. Velikost filtrů je $2s$ a jsou natočeny také ve směru orientace klíčového bodu. Pro zlepšení odolnosti vůči geometrickým deformacím a odstranění lokalizačních chyb jsou získané hodnoty d_x a d_y váženy Gaussovou funkcí s hodnotou $\sigma = 3.3s$.

Dalším krokem pak je, že se odezvy d_x a d_y sumarizují uvnitř každé podoblasti. Ve smyslu, aby byla uchována polarita změn intenzity v obraze, se používá taky suma absolutních hodnot $|d_x|$ a $|d_y|$. Potom může být každá podoblast popsána vektorem v , který popisuje její základní strukturu intenzity $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Konkatencí těchto vektorů z každé podoblasti získáme výsledný deskriptor o velikosti 64 hodnot. Takto vytvořený deskriptor je invariantní vůči změnám světelných podmínek. Invariance vůči změnám kontrastu je dosaženo změnou deskriptoru na jednotkový vektor.



Obrázek 2.8: SURF deskriptor (převzato z [2]). Pro sestavení deskriptoru se využívá čtvercová oblast s 4×4 podoblastmi, které leží okolo klíčového bodu. Důležité je, že hodnoty d_x , d_y , $|d_x|$ a $|d_y|$ jsou počítány relativně k orientaci klíčového bodu.

2.2 Selekcce příznaků obrazu

Často se stává že v obrazech, ze kterých se vytváří panoráma, se vyskytují objekty jako jsou lesy, moře nebo cihlové zdi. V těchto případech bývá v jednotlivých obrazech nalezeno příliš velké množství klíčových bodů. Množství nalezených klíčových bodů je tedy určeno počtem scale-space extrémů v obraze a nastavením prahů pro odstranění klíčových bodů, které mají nízký kontrast nebo příliš připomínají lineární hranu [13]. V každém obraze je nalezeno takových bodů v řádech stovek až tisíců. Práce s mnoha klíčovými body je velmi výpočetně náročná a především její výpočetní náklady rostou super-lineárně s počtem zpracovávaných bodů. To je však při zpracovávání obrazu nežádoucím prvkem, jelikož by pak práce s mnoha klíčovými body trvala mnoho strojového času. Z tohoto důvodu se musí tedy počet bodů nějakým způsobem redukovat.

Tento krok při tvorbě panoramat je velice důležitý z hlediska přesnosti a rychlosti zpracování výsledného obrazu. Pokud budou klíčové body nalezeny pouze ve středu obrazu a ne na okrajích, může být výsledný obraz značně nepřesný nebo se nemusí spojit vůbec, jelikož se nemusí najít dostatečný počet schodných klíčových bodů v sousedících obrazech. Stejný problém by také nastal v případě, kdy by byla velmi malá plocha, kde se dva sousední obrázky překrývají.

V případě, kdy je tedy klíčových bodů v obraze nalezeno příliš (zbytečně) mnoho a nebo jsou rozesety především v centrální části obrazu, tak se užívá různých přístupů jak tento počet snížit a přitom vybrat ty nejlepší z nich. Zde budou následně uvedeny tři metodiky pro vybrání těch nejvhodnějších bodů pro následné spojování napříč obrazy.

2.2.1 Výběr N nejsilnějších

Nezákladnějším a nejjednodušším přístupem je výběr N nejsilnějších klíčových bodů a jejich okolí. Jednoduše se všechny nalezené klíčové body ohodnotí a seřadí. Posléze se vybere N klíčových bodů, které mají nejlepší ohodnocení. Tyto body však mohou být situovány tak, že nepokryjí potřebnou část obrazu (např.: Pokud bychom chtěli vytvořit panorama z portrétů, tak nejsilnější body by byly v oblasti obličejů a ne na okrajích obrazů, které se budou spojovat.). Rozpoložení klíčových bodů, které byly vybrány tímto přístupem lze vidět na obrázku 2.9a. Lze tedy usoudit, že tento přístup není nejvhodnější pro tvorbu panoramat, a proto se častěji využívá selekce příznaků pomocí ANMS nebo K-D stromů.

2.2.2 ANMS

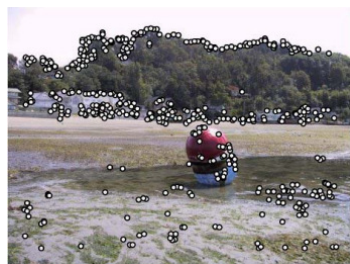
Složitějším a přijatelnějším přístupem pro výběr nejvhodnějších klíčových bodů je Adaptive Non-Maximal Suppression (ANMS), která je popsána např. v [10]. Tato metoda vybírá v obraze lokální maxima, jenž jsou založeny na síle rohu f_{HM} , jejichž odpovídající hodnoty jsou větší než všechny okolní body v okruhu potlačení, čili zde v tomto okruhu nemůže ležet další klíčový bod. Nejprve se poloměr potlačení nastaví na $r = 0$ a posléze je zvětšován až do té doby, než je nalezeno požadovaných N klíčových prvků. V praxi to vypadá tak, že se vytvoří z klíčových bodů uspořádaný seznam o velikosti N .

První položkou takového seznamu je globální maximum, které není žádným poloměrem potlačeno, a proto je poloměr potlačení nastaven na nekonečno. Poloměr potlačení se pak postupně zmenšuje s přidáváním dalších klíčových bodů do seznamu. Tento přístup je vhodný v tom, že pokud je jednou klíčový bod přidán do seznamu, tak už z něj nebude odebrán. Je tomu tak, protože pokud je klíčový bod maximem v poloměru potlačení r , tak bude také maximem v poloměru potlačení $r' < r$. V praxi Brown vytvořil ANMS více robustní požadavkem na to, aby nalezený soused měl dostatečně větší rohovou sílu než aktuální bod. Pak tedy minimální poloměr potlačení r_i je dán jako:

$$r_i = \min_j |\mathbf{x}_i - \mathbf{x}_j|, s.t. f(\mathbf{x}_i) < c_{robust} f(\mathbf{x}_j), \mathbf{x}_j \in \mathcal{I}, \quad (2.27)$$

kde \mathbf{x}_i je 2D lokace klíčového bodu a \mathcal{I} je sada lokací všech klíčových bodů. c_{robust} je hodnota, která zajišťuje, aby soused měl podstatně vyšší sílu pro to, aby byl umístěn do seznamu.

Na obrázku 4.9 lze vidět $N = 500$ klíčových bodů s nejvyššími hodnotami r_i v porovnání s výběrem N nejsilnějších. Také lze vidět, že při použití metody ANMS zůstávají takové klíčové body, které jsou rozesety po větší části obrazu, než u klasického výběru nejsilnějších klíčových bodů. Pokrytím celého rozsahu obrazu klíčovými body lze usoudit, že tato metoda je mnohem vhodnější pro vznik panoramatického obrazu než předcházející přístup.



(a) Nejsilnějších 500



(b) ANMS 500, $r = 16$

Obrázek 2.9: Srovnání rozpoložení klíčových bodů při selekci nejsilnějších bodů a při využití ANMS, kde r je poloměr potlačení klíčových bodů (převzato z [10]).

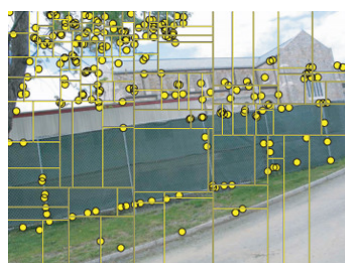
2.2.3 Selekcce pomocí k-d stromu

Posledním zde uvedeným přístupem je výběr za pomoci k-d stromu, který je blíže přiblížen v [13]. Cílem je získat N klíčových bodů pomocí postupného dělení obrázku na obdélníkové oblasti tak, aby každý nový řez rozdělil oblast s největším rozptylem klíčových bodů na dvě menší. Dělicí čáru mezi nově vzniklými oblastmi pak tvoří medián datových hodnot klíčových bodů.

K-d strom je 2-D generalizovaný binární strom, který bude uchovávat klíčové body v jednotlivých oblastech. Tento strom obsahuje b buněk tvořených ze souřadnic klíčových bodů. Pro každý uzel stromu jsou pak jeho data dělena na základě míry rozptylu. Jakmile je strom vytvořen, vybere se N/b nejsilnějších klíčových bodů z každé buňky k-d stromu. Jak je vidět na obrázku 4.10, body jsou rozesety po větší ploše než u klasického výběru nejsilnějších bodů. Rozpoložení finálně zvolených bodů je podobné jako u ANMS, ale díky různě velkým dílčím oblastem mohou být klíčové body stále hodně blízko u sebe.



(a) Nejsilnějších 256



(b) Nejsilnějších 256 po použití k-d stromu

Obrázek 2.10: Srovnání rozpoložení klíčových bodů při selekci nejsilnějších bodů a po použití dělení obrázku pomocí k-d stromu, kde r je poloměr potlačení klíčových bodů (převzato z [13]).

2.3 Spojování příznaků obrazu

Po výběru těch nejlepších a nejvhodnějších klíčových bodů včetně jejich deskriptorů se mohou tyto body propojit napříč obrazy. Spojováním příznaků se dá také nazvat hledáním korespondencí mezi obrazy, kdy se mezi sousedícími obrazy pracuje nad stejně velkými částmi obrazu. Cílem tohoto procesu je, aby byly nalezeny geometricky konzistentní korespondence mezi všemi obrazy. Prvním krokem je nalézt korespondence, poté se odstraní nekorespondující body a nakonec se použije metoda RANSAC, která odstraní zbylé špatně určené korespondence (outliery). Po spojení dostatečného počtu příznaků se může přejít k vytvoření samotného panoramatického obrazu.

Nejjednodušší přístup jak nalézt korespondenci je vzít příznak prvního obrazu a porovnat ho se všemi příznaky v druhém obrazu, při snaze najít co nejshodnější (nejpodobnější) příznak, který by mohl tomu prvnímu odpovídat. Tento výpočet je založen na klasickém použité Eukleidovské vzdálenosti mezi jednotlivými deskriptory příznaků. Problémem však je, že využití globálního prahu, který by určoval vzdálenost nejbližšího příznaku, nepracuje správně [1], a proto tento přístup není nejvhodnější. Rychlejší a vhodnějším přístupem je aproximace algoritmu nejbližšího souseda, která se nazývá Best-Bin-First (BBF). Samozřejmě je, že při jakémkoliv hledání korespondencí spousta příznaků nenalezne svou dvojici v množině všech příznaků ostatních obrazů, a proto je vhodné tyto osamocené příznaky nějakým vhodným způsobem odstranit.

2.3.1 Best Bin First algoritmus

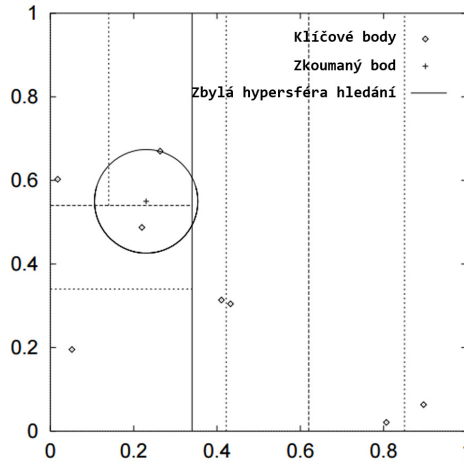
Algoritmus Best Bin First (BBF) [4] vznikl upravením přístupu hledání nejbližšího souseda pomocí k-d stromů v mnoha-dimenzionálním prostoru. Lowe ve své práci [21] uvádí, že tato metoda vrací správného nejbližšího souseda s vysokou pravděpodobností (až 95 %).

Základem algoritmu je dívat se na biny (uzly) v pořadí vzrůstající vzdálenosti od kořene stromu. Vzdálenost binu je definována jako nejkratší vzdálenost mezi kořenem a nějakým bodem na hranici binu. Toto prohledávání se dá snadno implementovat za pomoci prioritní fronty s malým množstvím režie. Do fronty se pak ukládá každý navštívený uzel stromu, pro který zde bude uložena jeho pozice a vzdálenost od kořene. Hlavním rozdílem od klasického hledání nejbližšího souseda pomocí k-d stromů je, že místo backtrackingu se z fronty vybírá uzel s aktuálně nejmenší vzdáleností. Dalším rozdílem je, že se pracuje pouze s omezeným počtem listů stromu (150 - 400) a dokonce jej může také redukovat za jeho chodu. Tomu je zapříčiněno díky použití hypersféry, jejíž poloměr je roven aktuální nejkratší vzdálenosti. Ta udává, že není třeba prohledávat ty biny, které jsou mimo tuto hypersféru, jelikož body v takovýchto binech už nemohou být nejbližším sousedem, a proto se listy stromu v těchto binech odstraní.

Na obrázku 2.11 lze vidět postupné dělení osmi bodů do dvou-dimenzionálního stromu. BFF zde nejprve zkoumá biny, které jsou umístěny nejbližší zkoumanému bodu. BFF zde také redukuje počet vyšetřených listových záznamů po zjištění, že nejbližší soused leží v nejbližším sousedním binu (přímo pod zkoumaným bodem) a ne v nejbližším binu ve stromové struktuře (vlevo od zkoumaného bodu).

2.3.2 Homografie

Homografie [12], [1] je projektivní (perspektivní) transformace určující vazbu korespondujících bodů mezi sousedícími obrázky. Lze také říct, že homografie mapuje body z prvního



Obrázek 2.11: K-d strom s počtem dimenzí $k = 2$ a 8 body (převzato z [4])

obrázku do druhého. Výhodou této transformace je, že pracuje pouze se n páry korespondujících bodů. Podmínkou pro správné řešení je, aby obrázky měly stejný střed promítání neboli se otáčely okolo stejného optického centra. Vztah mezi body jednotlivých obrázků lze zapsat jako:

$$\mathbf{x}' = \mathbf{H}\mathbf{x}, \quad (2.28)$$

kde \mathbf{x} a \mathbf{x}' jsou korespondující body obrázku a \mathbf{H} je 3×3 matice prvků homografie daná jako:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \quad (2.29)$$

Vzhledem k možné změně ohniskové vzdálenosti f se výsledný vztah $\mathbf{x}' = \mathbf{H}\mathbf{x}$ vyjádří jako:

$$\begin{bmatrix} fx' \\ fy' \\ f \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (2.30)$$

Pokud dosadíme f roznásobíme matice, získáme dvě lineární rovnice, které po drobných úpravách jsou:

$$h_{11}x + h_{12}y + h_{13} - h_{31}x' - h_{32}y' - h_{33}x' = 0 \quad (2.31)$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}y' - h_{32}x' - h_{33}y' = 0. \quad (2.32)$$

Řešení těchto rovnic tvoří výslednou matici homografie pro n korespondencí (většinou se používá $n = 4$ korespondencí):

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_2x_2 & -y'_2y_2 & -y'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0. \quad (2.33)$$

2.3.3 RANSAC

Pokud je nalezený dostatečný počet korespondencí, může se v nich přesto vyskytovat určitý počet outlierů, z tohoto důvodu se využívá RANSAC. Random Sample Consensus [15] je robustní odhadující procedura, která využívá minimální sadu náhodně vybraných korespondujících vzorků, aby spočítala transformační parametry obrazu a našla řešení matematického modelu (homografie), které se nejlépe schoduje s daty. Jelikož je tato sada vybírána ze všech korespondencí náhodně, je zde tedy vysoký požadavek na to, aby veškeré korespondence byly stanoveny co nejpřesněji, proto se pro co nejpřesnější výsledek korespondence nejdříve vyhledají pomocí např. BFF a až posléze je na tyto spojení aplikovaný algoritmus RANSAC.

Postup algoritmu

Prvním krokem algoritmu je, že se náhodně vybere sada r korespondujících bodů (podle Browna [8] $r = 4$), z kterých je následně vypočítán odhad homografie \mathbf{H} za použití přímé lineární transformace. Jelikož datová sada může obsahovat outliery, proto tato homografie nemusí být finální. Poté metoda spočítá počet inlierů, což jsou korespondence splňující vypočtenou homografii s tolerancí ϵ pixelů. Tento postup se n -krát opakuje a uchovává vždy to řešení, které má největší počet inlierů. Aby pravděpodobnost pro správné řešení byla co největší, musí být počet opakování vysoký (dle [8] $n = 500$). Vzhledem k pravděpodobnosti, že spojení příznaků obrazu je správné (pravděpodobnost inlieru) označené jako p_i , je pravděpodobnost nalezení správné transformace po n pokusech dána vzorcem:

$$p_h = 1 - (1 - (p_i)^r)^n. \quad (2.34)$$

Ze vzorce lze usoudit, že po velkém počtu iterací je pravděpodobnost nalezení správné homografie velmi vysoká. Minimální počet pokusů pak lze spočítat jako:

$$n = \frac{\log(1 - p_h)}{\log(1 - (p_i)^r)}. \quad (2.35)$$

2.3.4 Projekce

Veškeré předchozí řešení bylo takové, že byl vždy zvolen jeden obrázek jako hlavní (referenční) a ostatní se na něj vždy postupně napojovaly ze stran po jejich deformaci pomocí homografie. Tento přístup se nazývá projekce na rovinu, její výhodou je, že nezkrsluje

realitu, ale je vhodná pouze u malého počtu obrázků nebo u panoramat s celkovým malým zorným úhlem.

Mnohem lepším řešením je tedy projekce na nějaké 3-D těleso, ze kterých se nejčastěji používá válec nebo koule. Důkladné rozepsání těchto dvou projekcí sepsal Szeliski v [27]. V těchto případech se každý obrázek deformuje včetně jeho souřadného systému. Vznikají tedy obrázky v určité míře zkreslující realitu, ale díky projekci na stěny tělesa může výsledné panorama zobrazovat libovolný zorný úhel okolo otáčející se kamery. U projekce na kouli je pro nejpriznivější výsledek vhodné, aby fotografie pokrývaly celý plášť koule.

2.4 Smíchání obrazu

Po spojení do jednoho celku se může stát, že jednotlivé obrázky na sobě nesedí úplně přesně, nebo mají mezi sebou velké rozdíly intenzity barev. Proto v případě, kdyby dva sousedící obrázky panoramat byly pouze překryty jeden přes druhý, zůstala by ve výsledném obraze viditelná hrana těchto dílčích obrázků, nebo zkreslená realita části obrazu, která je nedeformována v jiném dílčím obraze. Pro odstranění těchto prvků se dva sousedící obrázky mezi sebou promísí (blending) tak, aby výsledek vypadal jako jeden celek.

Tato část postupu při tvorbě panoramat má velký vliv na výsledný obrázek. Pokud nebudou body přesně spojeny, avšak využije se vhodného smíchání obrazu, na výsledku se nic nepozná. Proto se u předchozích postupů může využívat větších aproximací za předpokladu použití vhodného a plynulého prolnutí napříč obrazy. Postupů jak docílit celistvého přechodu mezi obrazy je celá řada a zde budou uvedeny některé z nich.

2.4.1 Průměr intenzit

Nejjednodušší přístup jak požadovaného výsledku docílit je, že se vypočítá průměr intenzity barvy I každého pixelu, který se nachází v překrývající se oblasti dvou obrázků A a B , dán výrazem:

$$I_{AB} = \frac{I_A + I_B}{2}. \quad (2.36)$$

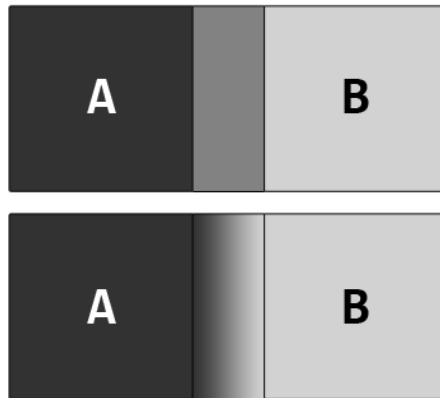
Jak takovéto spojení vypadá lze vidět v horní části obrázku 2.12. Tento postup se však nepoužívá, protože nežádoucí hrany jednotlivých obrázků se sice zeslabí, avšak stále jsou viditelné. Dalším nechtěným prvkem, který se může objevit v překrývající se oblasti dvou obrázků, mohou být tzv. duchové, což jsou polopropustné části jednoho obrazu, které se v druhém obraze vyskytují již na jiném místě (většinou pohybující se předměty). Tyto problémy v určité míře řeší prolnutí mezi jednotlivými obrázky za pomoci přechodu.

2.4.2 Prolnutí přechodem

Složitějším přístupem je lineární přechod z jednoho obrázku do druhého. Hodnota jednotlivých překrývajících se pixelů je vypočítána jako vážený průměr hodnot pixelů jednotlivých obrazů. Vzorec pro výpočet výsledné intenzity pixelu I_{AB} je tedy:

$$I_{AB} = \alpha I_A + (1 - \alpha) I_B, \quad (2.37)$$

kde α je váha pixelu. Na spodní části obrázku 2.12 lze vidět jeho lepší a přirozenější výsledek oproti předchozímu přístupu. Hrany dílčích obrazů sice zmizí, avšak polopropustné části zde přetrvávají většinou ve středové části, kdy žádná z dílčích intenzit nepřevažuje. Tato primitivní metoda se obvykle nepoužívá, ale ve své podstatě by mohla být využita v případě obrázků



Obrázek 2.12: Porovnání smíchání dvou obrazů pomocí průměru intenzit (nahore) a pomocí přechodu (dole).

s minimální změnou kontrastu a bez pohybujících se předmětů v obraze vyfocených za dokonalých podmínek.

2.4.3 Pyramidové smíchání

Další z metod smíchání obrazů, která už není tak primitivní, jako doposud uvedené metody je Laplaceovo pyramidové míchání [11], což je algoritmus využívající Gaussova rozostření pro plynulý přechod mezi dvěma obrazy. Algoritmus se také nazývá Multi-Band blending, který ve 2-bandové podobě využil Brown v jeho pracích [7] a [8], kde dokonce vylepšil toto smíchávání o "gain compensation", což je odstranění velkých rozdílů v jasu a kontrastu na okrajích obrázku.

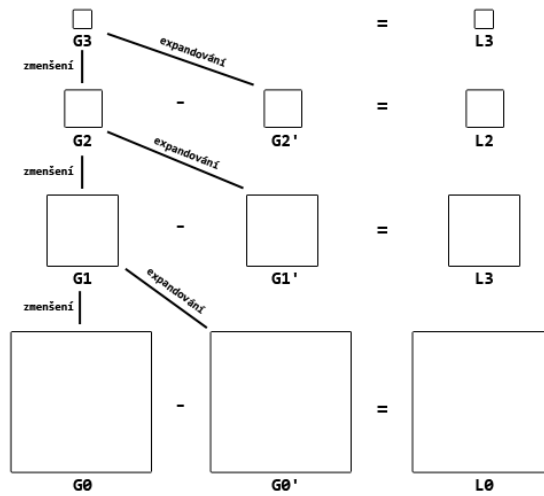
U tohoto druhu smíchání se pomocí Gaussových funkcí se postupně rozostřují originální obrázky v několika měřítkách (na obrázku 2.13 označeny jako $G_0 - G_3$). Poté se takto Gaussem rozostřený obrázek postupně expanduje z vyšších úrovní do nižších ($G_0' - G_2'$) a odečte od Gaussova obrázku v této úrovni, tímto se získají Laplaceovy obrázky ($L_0 - L_3$).

Po vygenerování Laplaceových pyramid pro jednotlivé překrývající se obrazy A a B , se tyto dva obrazy kombinují v jednotlivých Laplaceových úrovních kombinací dílčích obrazů každého z nich (na obrázku 2.14 označeny jako $LS_0 - LS_3$). Takto vzniklé obrazy se poté opět postupně expandují z nejvyšší úrovně do nižší ($LS_0' - LS_2'$) a jsou přičteny k původnímu Laplaceovu obrazu téže úrovně. Tento krok se opakuje, dokud se nedosáhne nejnižší úrovně (LS_0^*), čímž se dosáhne finálně smíchaného obrazu.

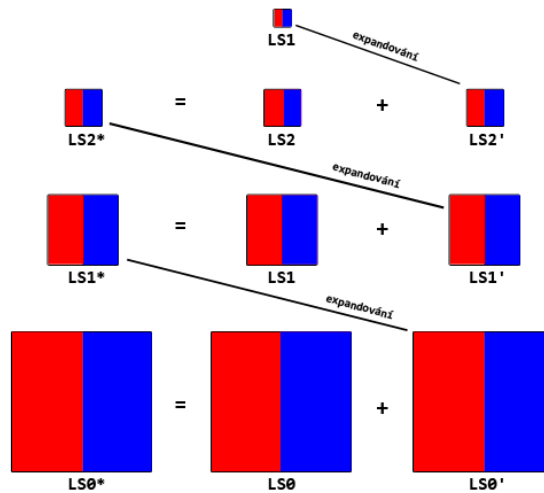
2.4.4 Graph cuts

Metodou naprosto odlišnou od předchozích přístupů ke smíchání obrazu jsou Graph cuts, jež jsou popsány v např. [19]. Tento přístup se liší v tom, že oproti míchání dvou obrazů pomocí přechodů nebo rozmazávání se snaží najít mezi nimi optimální řez. Metoda se snaží najít tento řez v takových místech, kde je rozdíl intenzity mezi dvěma obrazy co nejmenší. Výsledkem tedy jsou dva ořezané obrázky, které do sebe přesně zapadají a nemusí se tedy měnit hodnota intenzity každého pixelu zvlášť. Výhodou tohoto přístupu je, že má velice nízké výpočetní nároky.

Graf je matematická reprezentace navzájem propojených částí obrazu. Tyto části obrazu (uzly grafu) jsou získány za pomoci Záplavové segmentace. Každé propojení mezi uzly má

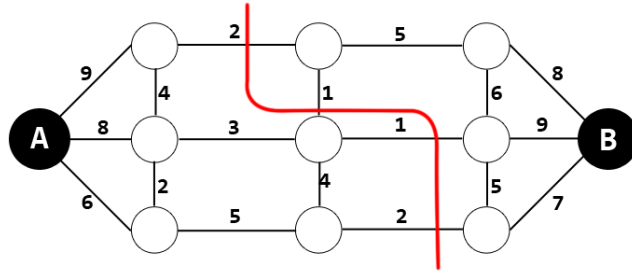


Obrázek 2.13: Tvorba Laplaceovy pyramidy pomocí rozdílu Gaussem rozostřených obrázků v několika úrovních.



Obrázek 2.14: Tvorba prolnutí obrázků A a B pomocí expandování Laplaceových obrázků z vyšších úrovní do nižších uvnitř pyramidy.

přidělenou svou váhu na základě změn intenzity v obrazech. Při hledání toho nejvhodnějšího řezu pro oddělení dvou obrázků se metoda snaží nalézt takové řešení, aby součet vah jednotlivých spojení uzlů v grafu byl co nejmenší. Příklad takového grafu a jeho řešení v podobě minimálního řezu lze vidět na obrázku 2.15.



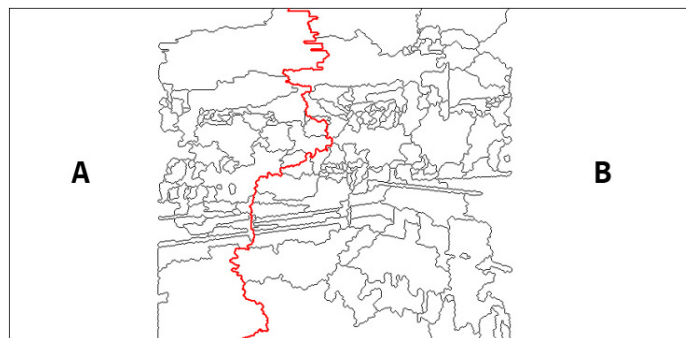
Obrázek 2.15: Příklad grafu pro metodu Graph cuts a jeho minimální řez (vyznačen červeně). Celkové ohodnocení řešení je $6 = 2 + 1 + 1 + 2$.

Postup metody

Nejprve se spočítá invertovaný rozdíl intenzit mezi jednotlivými obrázky I_1 a I_2 pomocí vzorce:

$$D(x, y) = 255 - \max(I_1(x, y) - I_2(x, y)). \quad (2.38)$$

Poté se nad takto upraveným obrazem využije Záplavové segmentace. Na obrázku 2.16 lze vidět vytvořené segmenty oblasti, kde se obrázky překrývají. Segmentací se oblasti, které jsou si podobné jsou spojovány, a proto při hledání nejvhodnějšího řezu by se mělo takovýmto oblastem vyvarovat. Veškeré segmenty pak tvoří uzly v grafu. Váha mezi jednotlivými uzly je dána celkovou sumou hraničních pixelů mezi dvěma segmenty. Nakonec se nalezne minimální řez, který je pak optimálním výsledkem (na obrázku 2.16 vyznačen červenou barvou).



Obrázek 2.16: Spojení dvou obrázků A a B pomocí metody Graph cuts a Záplavové segmentace. (převzato z [24])

Záplavová segmentace

Aby mohl být řez grafem co nejminimalističtější, musí se obraz rozčlenit do segmentů za pomoci Záplavové segmentace. Obrázek je zde brán jako topologická mapa, kde tmavší části obrázku představují údolí a světlejší části zase naopak vyvýšená místa. Vodní zdroj neboli zdroj záplav je umístěn do každého lokálního minima v obraze. Poté začne hladina vody stoupat a postupně zaplavovat území. Jakmile se voda ze dvou různých zdrojů setká, vytvoří tak hranici, která poté odděluje jednotlivé segmenty obrázku.

Kapitola 3

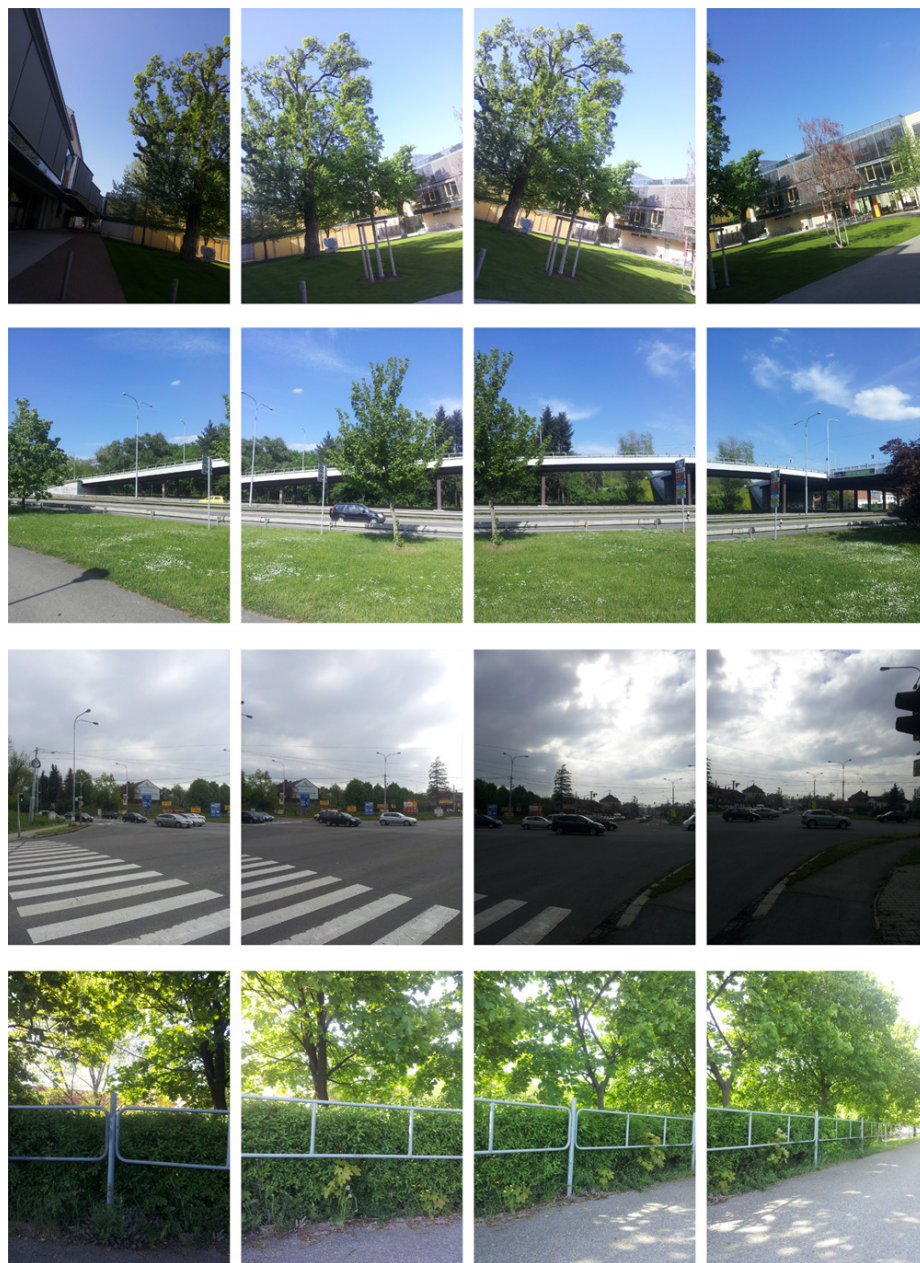
Datová sada

Datová sada pro tvorbu panoramat je zde volena na základě toho, aby pokryla celou škálu situací, které mohou při vytváření panoramatických obrázků nastat. Je důležité volit testovací sadu s rozmyslem na to, jak a za jakých podmínek mohou být jednotlivé dílčí obrázky pořízeny. Důraz tedy nemůže být kladen pouze na posun a rotaci fotoaparátu, ale také na světelné efekty, které mohou v obraze nastat jako je změna kontrastu. Metody využití při tvorbě panoramat musejí s těmito případy počítat za snahy vytvořit z fotografií co nejlepší výsledek.

Fotografie pro datovou sadu jsou zde vybrány tak, aby výše uvedené situace byly z co největší části pokryty. Proto jsou zde mimo klasicky pořízené obrázky také obrázky takové, které jsou "extrémny". Těmito extrémy lze označit obrázky takové, které jdou velmi těžko spojit. Jsou zde vybrány fotografie s hodně velkou nebo hodně malou plochou, kde se dva obrázky překrývají. Dále obrázky s mírným natočením a drobnou transformací, které jsou zde vybrány z důvodu testování invariance metod vůči otočení a posunu. Dalším extrémem jsou velké změny kontrastu mezi jednotlivými obrázky nebo také obrázky s horší kvalitou či vyšším šumem, které jsou zvoleny pro testování invariance vůči změnám kontrastu a šumu.

Posledním bodem na který je důležitý při výběru datové sady pohlížet je obsah fotografií. Není lehké spojit periodické objekty jako jsou např. lesy, vlny na moři, atd. Dalším případem je přesný opak a to, když je v obraze málo bodů pro spojení jako je např. nebe. V posledním případě jsou zde také fotografie s předměty pohybujícími mezi jednotlivými obrázky. Fotografie datové sady jsou vybírány také s ohledem na tyto případy.

Datová sada, která je přiložena k této práci, obsahuje sady fotografií pořízených mobilním telefonem většinou v okolí fakulty VUT FIT a města Brna. Její ukázkou lze vidět na obrázku 3.1. Pro rychlejší zpracování obrazu, je rozlišení oproti plné kvalitě fotoaparátu sníženo a u všech fotografií nastaveno na 1536 x 2048 pixelů (3 Mpx).



Obrázek 3.1: Ukázka fotografií datové sady

Kapitola 4

Porovnávání metod pro tvorbu panoramat

V této kapitole jsou popsány porovnávání jednotlivých metod při tvorbě panoramat. Jsou zde ukázány výhody a nevýhody metod jednotlivých dílčích kroků při vytváření panoramatických obrázků a jejich použitelnost v praxi. Je zde uveden rozdíl mezi jednotlivými přístupy vždy pro daný krok postupu tvorby zvlášť.

Pro srovnání metod byly použity implementace získané z online zdrojů a také vlastní zdrojové kódy. Využitými implementacemi je program Autostitch [6], který je založený na SIFT popisu klíčových bodů, metodě RANSAC, výpočtu homografie a také na všech druhích smíchání výsledného obrazu. Dalšími z využitých implementací byly vzorové programy Accord.NET [26], které obsahují implementace Harrisova detektoru, SURF detekce i deskripce, RANSAC, výpočtů homografie a lineárního smíchání. Součástí porovnání byly také vlastní zdrojové kódy implementující danou problematiku. Vlastní kódy jsou k práci přiloženy, obsahují implementaci jednotlivých metod pro detekci klíčových bodů, metod pro popis klíčových pomocí vektorů vlastností a také spojení výsledného obrazu.

4.1 Porovnání detektorů klíčových bodů

Porovnání mezi přístupy pro detekci klíčových bodů v obraze je zde zaměřeno především na výkon jednotlivých metod. Detektory byly testovány nad sadou osmi fotografií, které jsou zobrazeny na obrázku 4.1. Testovanými detektory klíčových bodů jsou Harrisův detektor rohů 2.1.2 (dále už jen HARRIS detektor), detektor blobů využívaný pro SIFT klíčové body 2.1.3 (dále už jen SIFT detektor) a detektor Fast-Hessian, který se využívá pro SURF klíčové body 2.1.4 (dále už jen SURF detektor). Detekce klíčových bodů pomocí těchto detektorů je implementována v souboru `src/detectors.cpp` za pomoci knihovny OpenCV [5]. Výsledky testování jsou zaznamenány v tabulkách 4.2, ?? a vyobrazeny na grafech 4.3, 4.4.

Na grafu 4.3 je znázorněn počet nalezených klíčových bodů v testovaných obrázcích. Jde vidět, že v případě vysokého kontrastu a velké členitosti fotografie (oplocení a semikráska) je počet nalezených bodů velmi nadprůměrný. V takovém případě může být problém s výsledným spojením obrazu, protože většina takovýchto bodů je velmi "slabá" a mohou tak ovlivnit výslednou tvorbu panoramatu (především pak výpočet homografie obrazu). Oproti tomu u fotografií se špatnými světelnými podmínkami nebo nízkou členitostí obsahu (chodba a malba) je počet nalezených bodů velice nízký. V případě velmi nízkého počtu klíčových



Obrázek 4.1: Obrázky použité pro testování detektorů a deskriptorů klíčových bodů

	HARRIS		SURF		SIFT	
	počet	čas [s]	počet	čas [s]	počet	čas [s]
chodba 3 Mpx	111	0,54	390	2,37	520	4,11
chodba 8 Mpx	197	1,09	643	5,93	1987	9,63
malba 3 Mpx	211	0,44	421	2,38	1161	4,19
malba 8 Mpx	1074	1,10	771	6,01	2704	4,11

Tabulka 4.1: Srovnání detekce klíčových bodů nad 3 Mpx a 8 Mpx obrázky.

bodů může nastat problém se spojením výsledného panoramatu, pokud není nalezen dostatek potřebných bodů na okrajích obrázku.

Rychlost detekce klíčových bodů nad jednotlivými obrázky je znázorněna na grafu 4.4. V případě SIFT s SURF detektorů lze vidět, že čas potřebný pro detekování závisí na členitosti obsahu. Lze vidět, že v případě, kdy se objevuje ve fotografii velké množství malých objektů, je čas potřebný pro detekci vyšší. U Harrisova detektoru členitost obrazu nemá vliv na jeho rychlost zpracování, ta byla pro každý testovaný obrázek téměř stejná. V tabulce 4.2 lze také vidět průměrná rychlost pro nalezení jednoho klíčového bodu. Je zřejmé, že Harrisův detektor nachází body rychleji, avšak je nevyhledává v multi-scale, což značí jejich nízký počet a nižší kvalitu při tvorbě panoramat. Detektory SURF a SIFT sice dosahují přibližně stejného poměru doby detekce a počtu nalezených bodů, avšak SURF detektor nachází klíčových bodů v průměru o polovinu méně, čímž se stává rychlejším.

V tabulce ?? je uveden rozdíl v počtu nalezených klíčových bodů a času potřebného pro detekci, pokud jsou fotografie vyfotografovány v rozlišení 1536 x 2048 pixelů (3 Mpx) a 2448 x 3264 px (8 Mpx). Lze vidět, že čím větší bude rozlišení obrázku, tím větší bude počet nalezených klíčových bodů, ale také čas potřebný pro jejich detekování (u SIFT detektoru až 10 sekund). Čas tvorby panoramatu se při větších rozlišeních velmi prodlužuje.

Na obrázku 4.2 lze vidět rozdíl v rozpoložení nalezených klíčových bodů obrazu za použití jednotlivých metod. Je zřejmé, že u všech metod je velké množství bodů rozmístěno ve středové části obrázku, což může způsobit problém při spojování výsledného obrazu. Pro další řešení tvorby panoramat je vhodné redukovat počet detekovaných klíčových bodů.



(a) HARRIS, 181 bodů



(b) SIFT, 499 bodů



(c) SURF, 420 bodů



(d) HARRIS, 100 nej. bodů



(e) SIFT, 100 nej. bodů

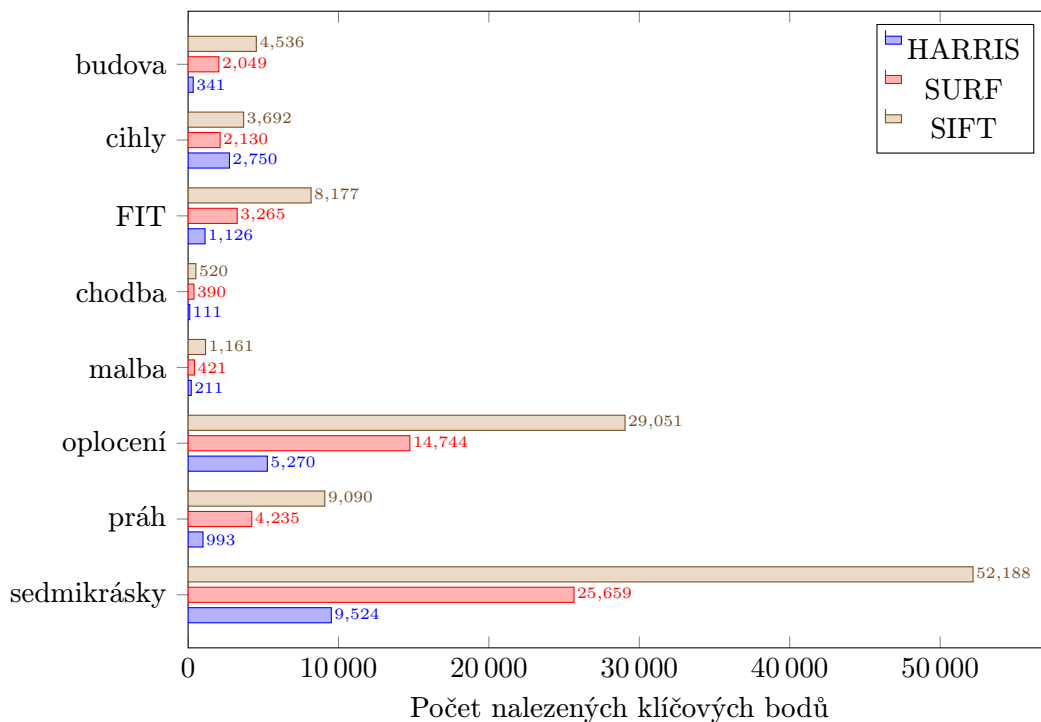


(f) SURF, 100 nej. bodů

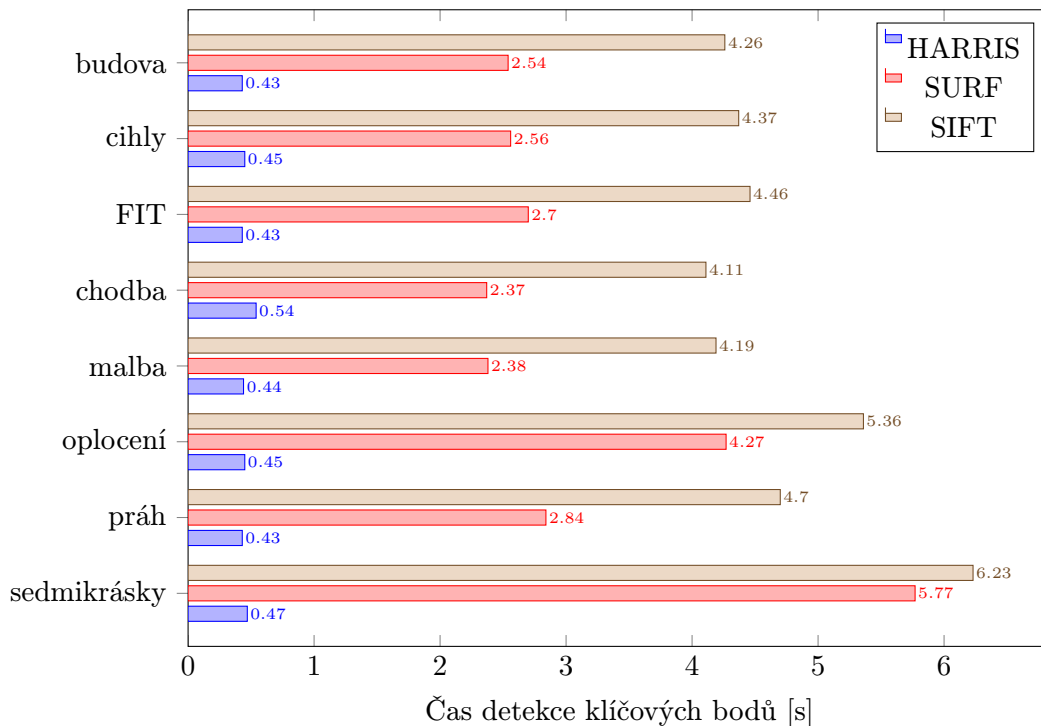
Obrázek 4.2: Rozpoložení nalezených klíčových bodů v obrázku velikosti 384 x 512 px získaných za pomoci detektorů klíčových bodů. Ve spodní řadě je znázorněno rozpoložení 100 nejsilnějších klíčových bodů.

	HARRIS			SURF			SIFT		
	počet	čas [s]	t_1 [ms]	počet	čas [s]	t_1 [ms]	počet	čas [s]	t_1 [ms]
budova	341	0,43	1,26	2049	2,54	1,24	4536	4,26	0,94
cihly	2750	0,45	0,16	2130	2,56	1,20	3692	4,37	1,18
FIT	1126	0,43	0,38	3265	2,7	0,83	8177	4,46	0,55
chodba	111	0,54	4,86	390	2,37	6,08	520	4,11	7,90
malba	211	0,44	2,09	421	2,38	5,65	1161	4,19	3,61
oplocení	5270	0,45	0,09	14744	4,27	0,29	29051	5,36	0,18
práh	993	0,43	0,43	4235	2,84	0,67	9090	4,7	0,52
sedmikráska	9524	0,47	0,05	25659	5,77	0,22	52188	6,23	0,12
průměr	2541	0,46	1,17	6612	3,18	2,02	13552	4,71	1,88

Tabulka 4.2: Dosažené výsledky detekování všech klíčových bodů pro jednotlivé detektory. Hodnota t_1 značí průměrnou dobu potřebnou pro detekování jednoho klíčového bodu.



Obrázek 4.3: Srovnání počtu všech detekovaných klíčových bodů pro jednotlivé detektory klíčových bodů.



Obrázek 4.4: Srovnání doby potřebné pro detekci všech klíčových bodů v obraze.

4.2 Porovnání metod pro selekci klíčových bodů

Z tabulek 4.2 a 4.3 lze vyčíst, že práce s velkým počtem klíčových bodů je velmi časově náročná, a proto se využívá selekce klíčových bodů, která snižuje celkový počet klíčových bodů. Množství bodů, které se vybírají pro další zpracování, závisí především na rozlišení obrázku. Vhodný počet selektovaných klíčových bodů je okolo 300–500, ale v případě menších rozlišení obrázků stačí uchovávat bodů i méně. Množství vybíraných bodů je velmi důležité pro to, aby byla co největší šance na úspěšné spojení dvou obrázků. Nejdůležitější při volbě vhodného přístupu je, aby v obou sousedících obrázcích byly nalezeny sobě korespondující body. Doba potřebná pro získání konkrétního počtu vhodných bodů se mezi jednotlivými přístupy nijak zvláště neliší, avšak vždy je závislá na počátečním počtu detekovaných klíčových bodů.

Na obrázku 4.5 je vidět, že jednotlivé metody se mezi sebou liší v rozpození finálně zvolených klíčových bodů. Nejlepšího pokrytí v obrázku dosahuje sada náhodně vybraných bodů, avšak použitelnost tohoto přístupu je naprosto nulová, jelikož mezi těmito body jsou často "slabé" body a také je zde minimální šance, že v sousedním obrázku budou náhodně vybrány jejich odpovídající klíčové body. Využití sady N-nejsilnějších klíčových bodů dosahovalo lepších výsledků, ale často se stávalo, že nejsilnější body nebyly na okrajích obrázku, a proto nedošlo k výslednému spojení. Pro zlepšení šance na úspěch, avšak na úkor času, by v tomto případě bylo zvětšit sadu vybíraných bodů. Nejlepších výsledků dosahovaly sady, na které byla aplikovaná Non-Maximal Suppression, jelikož získané body byly často rozesety po větší části obrazu než výběr N-nejsilnějších.

Problém se zajištěním dostatečného počtu korespondujících bodů může být také vyřešen tím, že se určí regiony vyhledávání, uvnitř kterých budou dané body detekovány a selektovány. Těmito regiony pak mohou být okraje obrázku, ve kterých je mnohem větší šance na to, že budou vybrány body, které se nacházejí také v jiném obrázku.

4.3 Porovnání deskriptorů klíčových bodů

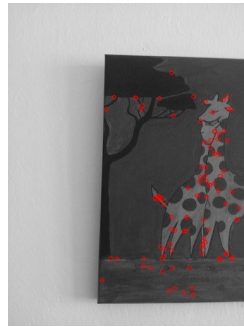
Přístupy pro popis klíčových bodů pomocí deskriptorů se liší především v rychlosti, kterou jsou jednotlivé body popisovány. Metody byly testovány na stejné sadě fotografií, jako detektory klíčových bodů. Testovanými deskriptory klíčových bodů jsou SIFT deskriptor 2.1.3 a SURF deskriptor 2.1.4. Testování popisu klíčových bodů pomocí těchto deskriptorů je implementováno v souboru `src/descriptors.cpp`. Výsledky testování jsou zaznamenány v tabulce 4.3 a vyobrazeny na grafu 4.7.

Jak již bylo uvedeno výše, tak SIFT popisuje okolí klíčového bodu pomocí vektoru 128 hodnot na rozdíl metody SURF, která pro popis používá vektor pouze 64 hodnot. Tento rozdíl v ukládání potřebných informací z okolí klíčových bodů pak způsobuje velké rozdíly v rychlosti zpracování panoramatu a také v paměťové náročnosti daného přístupu. Jak lze vidět na grafu 4.7, tak SURF dosahoval 4x rychlejšího výsledku při popisování klíčových bodů než metoda SIFT. Je zde taky patrné, že je vhodné pro rychlejší zpracování panoramat redukovat počet klíčových bodů s kterými se pracuje, protože popis všech klíčových bodů obrázku trval i desítky sekund.

SIFT i SURF dosahují vysoké invariance vůči deformaci obrazu, avšak SIFT deskriptory jsou díky svému detailnějšímu popisu nepatrně lepší. Na obrázku 4.6 lze vidět invariance deskriptorů klíčových bodů vůči otočení obrazu, která je zobrazená pomocí spojení korespondujících bodů.



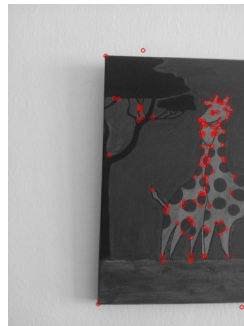
(a) Všechny klíčové body



(b) 100 náhodných bodů



(c) 100 nejsilnějších bodů



(d) 100 bodů za pomoci NMS

Obrázek 4.5: Rozpoložení selektovaných klíčových bodů v obrázku velikosti 384 x 512 px pomocí výběru nejsilnějších klíčových bodů, výběru náhodných klíčových bodů a výběru s využitím Non-Maximal Suppression.

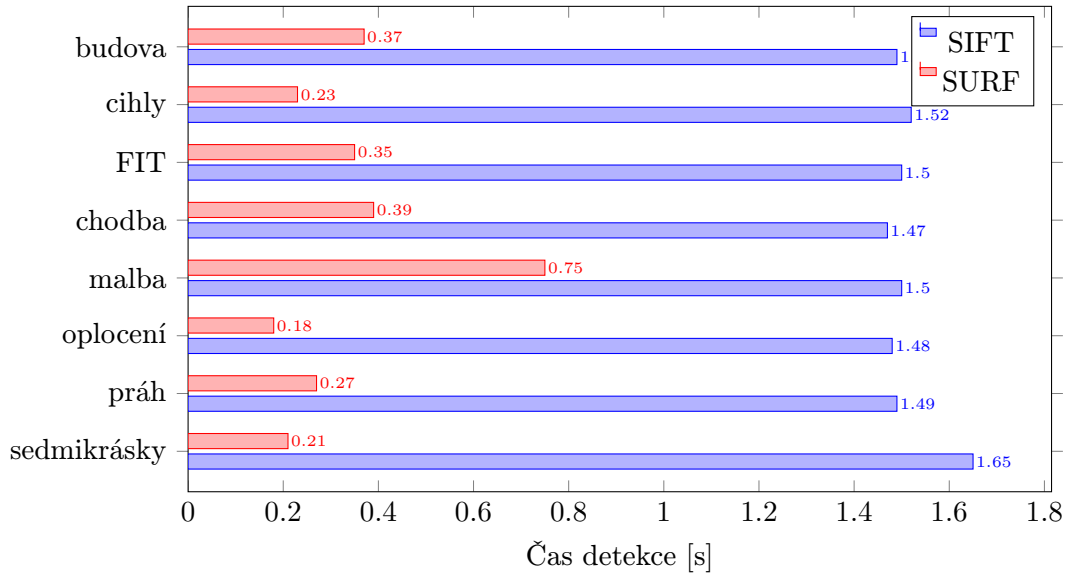


(a) Spojené korespondující body



(b) Výsledné spojení obrazu

Obrázek 4.6: Ukázka invariance SURF deskriptorů vůči rotaci obrazu.



Obrázek 4.7: Srovnání doby potřebné pro popis 300 nejsilnějších klíčových bodů

obrázek	počet deskriptorů	SIFT		SURF	
		čas [s]	t_{300} [s]	čas [s]	t_{300} [s]
budova	2049	6,43	1,49	2,47	0,37
cihly	2130	7,14	1,52	1,52	0,23
FIT	3265	9,89	1,50	3,18	0,35
chodba	390	1,83	1,47	0,48	0,39
malba	421	1,98	1,50	0,87	0,75
oplocení	14744	41,76	1,48	10,22	0,18
práh	4235	12,79	1,49	2,77	0,27
sedmikráska	25659	73,11	1,65	16,93	0,21
průměr	6612	19,37	1,51	4,81	0,34

Tabulka 4.3: Dosažené hodnoty deskripce všech klíčových bodů v obraze. Hodnota t_{300} značí dobu potřebnou pro popis 300 nejsilnějších klíčových bodů.

4.4 Porovnání přístupů pro smíchání obrazu

Pokud chceme co nejlepší výsledek, tak smíchání obrazu je nejdůležitějším krokem hned za samotným spojením. I nedostatky ve spojení dvou obrázků se dají odstranit díky jejich smíchání. V této sekci budou porovnávány přístupy smíchání sousedících obrázků již spojeného panoramatu. U porovnávání je nejdůležitější výsledný optický vjem, který daný přístup vytváří. U panoramat je totiž velice důležité, aby v obraze nebyly viditelné jakékoliv chyby, vzniklé pořizováním fotografií či vytvářením panoramatu. Není to však podmínkou, aby metoda pro smíchávání odstranila všechny tyto chyby, ale pokud tomu tak nenastane, projeví se to na celkovém výsledku. Hlavní podmínkou pro tento krok tvorby je, aby panoráma bylo spojeno s co největší přesností, protože pokud je panoráma spojeno nepřesně, tak po smíchání obrazu může být výsledná realita hodně zkreslená.

Důležitým bodem samotného smíchání je odstranění viditelných hran mezi obrázky (obrázky 4.8a, 4.9a), které vzniknou změnou světelných podmínek a kontrastu mezi jednotli-

vými fotografiemi. Problémem pak však může být, pokud je tato změna příliš velká. V tomto případě, i po smíchání panoramatu, se takovéto oblasti jeví tak, že v určité části obrazu je mnohem větší tma než v jeho okolí (obrázky 4.8b, 4.8c, 4.9b, 4.9c). Aby i tento problém byl co nejvíce potlačen, musí se využít "gain compensation", který vyrovná hodnoty intenzit v jednotlivých obrázcích (obrázky 4.8d, 4.9d).

Dalším úkazem, který se snaží smíchávání odstranit je výskyt tzv. duchů v obraze, kteří se objevují především při použité smíchání pomocí lineárního přechodu (obrázek 4.10a). Jsou to polopropustné části obrazu, které vzniknou při fotografování objektu, který je v pohybu. Tyto části mohou tvořit také stojící objekty díky nepřesnosti homografie, pokud bude velká změna pozorovacího úhlu mezi fotografiemi, nebo pokud objekty které jsou při fotografování hodně blízko u fotoaparátu. Problém polopropustných částí částečně odstraňuje multi-band smíchání (obrázek 4.10b). Plně problém pak řeší Graph cuts, jelikož každý pixel výsledného spojení je vždy závislý na hodnotě pixelu z pouze jednoho původního obrázku.

Stejně jako kvalita výsledného optického vjemu je také důležitá rychlost zpracování samotného smíchání. U obyčejného průměru intenzit nebo lineárního přechodu se pro každý pixel vypočítá jeho výsledná hodnota pouze na základě jednoho vzorce, a proto jsou velmi rychlé na zpracování. Kvalita výsledků těchto přístupů však není většinou dostačující z důvodů uvedených v tabulce 4.4, proto se na úkor času potřebného pro zpracování panoramatu používá metoda multi-band, která dosahuje mnohem lepších výsledků. Její čas zpracování je však vysoký z důvodů vytváření Laplaceových pyramid. Metoda Graph cuts stejně jako multi-band podává velmi kvalitní výsledky a její čas potřebný ke smíchání dvou obrázků zpomaluje záplavová segmentace obrazu a následné vytvoření řezu.

Pro vybrání správného přístupu je důležité, na co bude výsledný program využíván. Pokud je potřeba rychlého zpracování, tak smíchání pomocí lineárního přechodu má ideální poměr rychlosti a kvality výsledku. Pokud by tvorba měla být zaměřena na kvalitu výsledku, poté se musí využít graph cuts nebo multi-band smíchání nejlépe s gain compensation pro co nejlepší výsledek.

	odstranění hran	odstranění duchů	rychlost zpracování
průměr intenzit	ne	ne	vysoká
prolnutí přechodem	částečně	ne	vysoká
multi-band	částečně	částečně	velmi nízká
multiband-band s gain com.	ano	částečně	velmi nízká
graph cuts	částečně	ano	nízká

Tabulka 4.4: Porovnání metod pro smíchání obrazu.



(a) Bez smíchání



(b) Lineární přechod



(c) Multi-band



(d) Multi-band s využitím gain compensation

Obrázek 4.8: Srovnání metod využívaných pro smíchání obrazu nad fotografiemi křižovatky.



(a) Bez smíchání



(b) Lineární přechod



(c) Multi-band

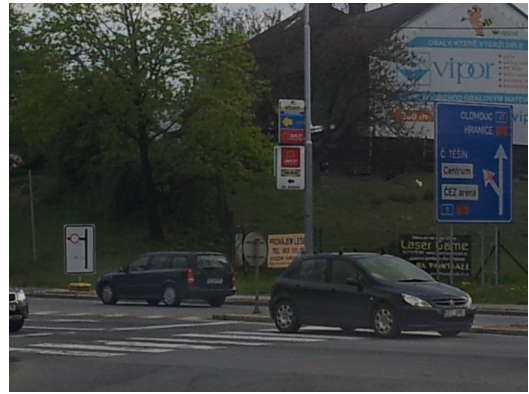


(d) Multi-band s využitím gain compensation

Obrázek 4.9: Srovnání metod využívaných pro smíchání obrazu nad fotografiemi Fakulty informačních technologií VUT v Brně.



(a) Lineární přechod



(b) Multi-band

Obrázek 4.10: Přítomnost polopropustných částí v obraze při smíchání pomocí lineárního přechodu. V případě multi-band smíchání se již tyto části nevyskytují.

Kapitola 5

Závěr

Cílem této práce bylo zaměřit se na tvorbu panoramat a také na srovnání metod, které jsou pro tuto problematiku využívány. V práci je vytvořen náhled nad skládáním panoramatických obrázků z několika dílčích obrázků. Celý postup tvorby panoramat je zde podrobně sepsán do po sobě navazujících kroků. Pro každý krok tvorby bylo uvedeno několik z často používaných přístupů, které tyto dílčí sekce řeší.

Porovnání jednotlivých metod bylo prováděno na pořízené datové sadě za pomoci implementací získaných z online zdrojů i vlastně vytvořených programů. V práci jsou uvedeny důležité vlastnosti, výhody a nevýhody jednotlivých sepsaných metod. Při řešení bylo zjištěno, že je velký rozdíl mezi rychlostí zpracování a kvalitou spojení výsledného panoramatu. Z dosažených výsledků srovnávání lze usoudit, že výběr nejvhodnějšího algoritmu pro tvorbu panoramat je závislý na požadavku na výslednou rychlost a kvalitu zpracování.

Dalším vývojem v tvorbě panoramatických snímků by bylo experimentování s dalšími metodami pro detekci a deskripci klíčových bodů, jelikož existují detektory i deskriptory, které dokáží mnohem rychleji zpracovávat klíčové body než mnou uvedené metody. Dalším důležitým bodem rozvoje pro zrychlení tvorby panoramat by bylo se zaměřit na selekci vhodných klíčových bodů pro spojování, kdy jde experimentovat s vyhledáváním klíčových bodů uvnitř velmi malých oblastí obrazu.

Literatura

- [1] *Object Recognition from Local Scale-Invariant Features, ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, Washington, DC, USA: IEEE Computer Society, 1999, ISBN 0-7695-0164-8.
- [2] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, ročník 110, č. 3, Červen 2008: s. 346–359, ISSN 1077-3142, doi:10.1016/j.cviu.2007.09.014.
URL <http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- [3] Bay, H.; Tuytelaars, T.; Van Gool, L.: SURF: Speeded Up Robust Features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.
- [4] Beis, J. S.; Lowe, D. G.: Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, Washington, DC, USA: IEEE Computer Society, 1997, ISBN 0-8186-7822-4, s. 1000–.
URL <http://dl.acm.org/citation.cfm?id=794189.794431>
- [5] Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [6] Brown, M.; Lowe, D.: Autostitch. [software], 2004.
URL <http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html>
- [7] Brown, M.; Lowe, D. G.: Recognising Panoramas. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03*, Washington, DC, USA: IEEE Computer Society, 2003, ISBN 0-7695-1950-4, s. 1218–.
URL <http://dl.acm.org/citation.cfm?id=946247.946772>
- [8] Brown, M.; Lowe, D. G.: Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, ročník 74, č. 1, 2007: s. 59–73, ISSN 0920-5691, doi:10.1007/s11263-006-0002-3.
URL <http://dx.doi.org/10.1007/s11263-006-0002-3>
- [9] Brown, M.; Szeliski, R.; Winder, S.: Multi-Scale Oriented Patches. Technická Zpráva MSR-TR-2004-133, Microsoft Research, December 2004.
URL <http://research.microsoft.com/apps/pubs/default.aspx?id=70120>
- [10] Brown, M.; Szeliski, R.; Winder, S.: Multi-Image Matching Using Multi-Scale Oriented Patches. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, Washington, DC, USA: IEEE Computer Society, 2005, ISBN

- 0-7695-2372-2, s. 510–517, doi:10.1109/CVPR.2005.235.
 URL <http://dx.doi.org/10.1109/CVPR.2005.235>
- [11] Burt, P. J.; Adelson, E. H.: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. kapitola The Laplacian Pyramid As a Compact Image Code, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987, ISBN 0-934613-33-8, s. 671–679.
 URL <http://dl.acm.org/citation.cfm?id=33517.33571>
- [12] Chen, J.; Benzeroual, K.; Allison, R.: Robust homography for real-time image un-distortion. In *3D Imaging (IC3D)*, 2013 International Conference on, Dec 2013, s. 1–8, doi:10.1109/IC3D.2013.6732075.
- [13] Cheng, Z.; Devarajan, D.; Radke, R. J.: Determining Vision Graphs for Distributed Camera Networks Using Feature Digests. *EURASIP J. Appl. Signal Process.*, ročník 2007, č. 1, Leden 2007: s. 220–220, ISSN 1110-8657, doi:10.1155/2007/57034.
 URL <http://dx.doi.org/10.1155/2007/57034>
- [14] Edelman, S.; Intrator, N.; Poggio, T.: Complex cells and Object Recognition, 1997.
 URL <http://cogprints.org/561/>
- [15] Fischler, M. A.; Bolles, R. C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, ročník 24, č. 6, Červen 1981: s. 381–395, ISSN 0001-0782, doi:10.1145/358669.358692.
 URL <http://doi.acm.org/10.1145/358669.358692>
- [16] Harris, C.; Stephens, M.: A Combined Corner and Edge Detector. In *Proceedings of the Alvey Vision Conference*, Alvey Vision Club, 1988, s. 23.1–23.6, doi:10.5244/C.2.23.
- [17] Juan, L.; Gwun, O.: SURF applied in panorama image stitching. In *Image Processing Theory Tools and Applications (IPTA)*, 2010 2nd International Conference on, July 2010, ISSN 2154-5111, s. 495–499, doi:10.1109/IPTA.2010.5586723.
- [18] Karthik, R.; AnnisFathima, A.; Vaidehi, V.: Panoramic view creation using invariant moments and SURF features. In *Recent Trends in Information Technology (ICRTIT)*, 2013 International Conference on, July 2013, s. 376–382, doi:10.1109/ICRTIT.2013.6844233.
- [19] Kwatra, V.; Schodl, A.; Essa, I.; aj.: Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, ročník 22, č. 3, July 2003: s. 277–286.
- [20] Lindeberg, T.: Feature Detection with Automatic Scale Selection. *Int. J. Comput. Vision*, ročník 30, č. 2, Listopad 1998: s. 79–116, ISSN 0920-5691, doi:10.1023/A:1008045108935.
 URL <http://dx.doi.org/10.1023/A:1008045108935>
- [21] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, ročník 60, č. 2, Listopad 2004: s. 91–110, ISSN 0920-5691, doi:10.1023/B:VISI.0000029664.99615.94.
 URL <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>

- [22] Mikolajczyk, K.; Schmid, C.: Scale & Affine Invariant Interest Point Detectors. *Int. J. Comput. Vision*, ročník 60, č. 1, Říjen 2004: s. 63–86, ISSN 0920-5691, doi:10.1023/B:VISI.0000027790.02288.f2.
URL <http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2>
- [23] Moravec, H.: Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. In *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University*, CMU-RI-TR-80-03, September 1980.
- [24] Nyman, P.: Image Stitching using Watersheds and Graph Cuts.
URL <http://www.maths.lth.se/matematiklth/personal/petter/rapporter/panorama2.pdf>
- [25] Příbyl, B.; Chalmers, A.; Zemčík, P.: Feature Point Detection Under Extreme Lighting Conditions. In *Proceedings of the 28th Spring Conference on Computer Graphics, SCCG '12*, New York, NY, USA: ACM, 2012, ISBN 978-1-4503-1977-5, s. 143–150, doi:10.1145/2448531.2448550.
URL <http://doi.acm.org/10.1145/2448531.2448550>
- [26] Souza, C.: Accord.NET Sample Applications. [software], 2009-2014.
URL <http://accord-framework.net/samples>
- [27] Szeliski, R.: Image Alignment and Stitching: A Tutorial. *Found. Trends. Comput. Graph. Vis.*, ročník 2, č. 1, Leden 2006: s. 1–104, ISSN 1572-2740, doi:10.1561/0600000009.
URL <http://dx.doi.org/10.1561/0600000009>

Příloha A

Obsah CD

- /bp.pdf – text bakalářské práce
- /datova_sada – datová sada testování tvorby panoramat
- /palkat.pdf – plakát pro prezentaci bakalářské práce
- /smichani_obrazu – dosažené výsledky smíchání panoramat
- /src – zdrojové kódy pro testování tvorby panoramat
- /text – zdrojové kódy textu bakalářské práce