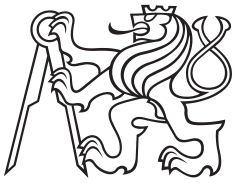
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Detekce klíčových bodů pomocí metody SIFT

Václav Hudeček

Vedoucí: Mgr. Lukáš Adam, Ph.D.
Obor: Otevřená informatika
Květen 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hudeček** Jméno: **Václav** Osobní číslo: **492072**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra měření**

Studijní program: **Otevřená informatika**

Specializace: **Internet věcí**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Detekce klíčových bodů pomocí metody SIFT

Název bakalářské práce anglicky:

Detection Keypoint by the SIFT method

Pokyny pro vypracování:

Tématem práce je detekce klíčových bodů na obrázku.
1. Vysvětlete, k čemu tyto metody jsou, a seznamte se se základními algoritmy.
2. Dále se zaměřte na metodu scale-invariant feature transform (SIFT). Tuto metodu podrobně popište a proveďte její (zjednodušenou) implementaci.
3. Empiricky ukažte rozdíly mezi Vaší a OpenCV implementací.
4. Dále ukažte, že je metoda schopná rozpoznat stejný objekt na různých fotografiích.
5. Nakonec naimplementujte metodu na slepení více fotografií do módu panorama.

Seznam doporučené literatury:

[1] Lowe, David G.: Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision 2, 1150–1157 (1999)
[2] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60:2, 91–110 (2004)
[3] Forsyth, D., Ponce, J.: Computer vision: A modern approach. Prentice Hall, (2011)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Mgr. Lukáš Adam, Ph.D. centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **08.02.2022** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:
do konce letního semestru 2022/2023

Mgr. Lukáš Adam, Ph.D.		prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____ _____

Datum převzetí zadání Podpis studenta

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 18. května 2022

Abstrakt

Práce se zaobírá popisem SIFT algoritmu a jeho průběhem krok po kroku. Následuje využití SIFTu při tvorbě panorama, kde samotný proces tvorby panorama, pro dva až tři obrázky, je zde na teoretické rovině popsán. Výsledky obou algoritmů jsou na konci práce popsány a zdokumentovány, je na nich ukázáno, co oba algoritmy umí, jaké jsou jejich výhody a nedostatky.

Klíčová slova: SIFT, detekce, klíčové body, panorama, homografie, RANSAC

Abstract

The work deals with the description of the SIFT algorithm and its course step by step. This is followed by the use of SIFT in the creation of a panorama, where the very process of creating a panorama, for two to three images, is described here on a theoretical level. The results of both algorithms are described and documented at the end of the work, they show what both algorithms can do, and what are their advantages and disadvantages.

Keywords: SIFT, detection, keypoints, panorama, homography, RANSAC

Obsah

1 Úvod 1

2 Scale-Invariant Feature Transform 3

2.1 Nalezení kandidátů na klíčové body 3

2.1.1 Výpočet počtu oktáv 3

2.1.2 Rozmazání na oktávě 4

2.1.3 Výpočet DOG 4

2.1.4 Detekce lokálních extrémů . . . 5

2.2 Lokalizace klíčových bodů 5

2.2.1 Určení pozice klíčových bodů . 6

2.2.2 Eliminace hran a ploch 6

2.3 Přiřazení orientace 8

2.4 Vytvoření deskriptoru klíčového bodu 9

3 Spojení obrázků (Panorama) 11

3.1 Transformace obrazu 11

3.2 Homogenní souřadnice 12

3.3 Afinní transformace 12

3.4 Projektivní transformace 13

3.5 Homografie 13

3.5.1 Výpočet homografie 14

3.6 RANSAC 16

3.6.1 Průběh RANSAC algoritmu . 16

4 Výsledky 19

4.1 SIFT algoritmus 19

4.1.1 Průběh nalezení klíčových bodů 19

4.1.2 Hledání objektu na obrázku . 22

4.1.3 Porovnání s OpenCV implementací SIFT 23

4.2 Tvorba panorama 24

4.2.1 Tvorba panorama ze dvou obrázků 24

4.2.2 Tvorba panorama ze tří obrázků 26

5 Závěr 29

Literatura 31

Obrázky

2.1 Výpočet DOG z rozmazaných obrazů [Lowe, 2004] 5

2.2 Určení lokálního extrému [Lowe, 2004] 5

2.3 Převod orientace pixelů v okolí na deskriptor [Lowe, 2004] 9

3.1 Princip správného spojení obrázků [Collins et al., 2017] 11

3.2 Rozdíl mezi projektivní a afinní transformací [GraphicsMill] 13

3.3 Ukázka homografie [Qureshi, 2021] 14

3.4 Ukázka několika iterací RANSAC algoritmu, metoda: line fitting [Collins, 2007] 17

4.1 Vybraný obrázek 19

4.2 Kandidáti na klíčové body, počet nalezených kandidátů je 1304 20

4.3 Lokalizované klíčové body, počet klíčových bodů je 293 20

4.4 Zoom na výsledné deskriptory klíčových bodů 21

4.6 Výsledek hledání objektu vpravo na obrázku vlevo 22

4.7 Klíčové body s deskriptory nalezené za pomoci OpenCV 23

4.8 Klíčové body s deskriptory nalezené za pomoci OpenCV 24

4.9 Obrázky ke spojení 25

4.10 Transformovaný obrázek 4.9*b* . 25

4.11 Výsledné spojení obrázků (panorama) 26

4.12 Obrázky ke spojení 27

4.13 Krok 1: Transformace 4.12*c* a spojení s 4.12*b* 28

4.14 Krok 2: Transformace nově vzniklého 4.13*b* a finální spojení s 4.12*a* 28

Kapitola 1

Úvod

Zpracování obrazu slouží strojům jako primární zdroj informace o určitém obrazu či okolním světu. Při zpracování obrazu je důležité určit, které části či konkrétní body daný obraz definují, jenže to se neurčuje jednoduše. Tento problém pomáhá řešit SIFT algoritmus. SIFT neboli „Scale-Invariant Feature Transform“, patří k algoritmům počítačového vidění, které se využívají na rozpoznávání objektů, spojování obrazů či rozpoznávání gest. Principem je hledání stabilních klíčových bodů na určitém obrázku nebo ploše obrázku. Na nalezené klíčové body se dá nahlížet, jako na určitý základní popis obrazu, skrze který lze daný obraz rozeznat i přes různé transformace, které ho normálně vizuální liší od původního obrazu. Tyto klíčové body jsou invariantní vůči velikosti (měřítku) a orientaci (transformaci).

Výsledky vrácené tímto algoritmem se dají využít pro spousty aplikací, jako je třeba rozpoznání objektů nebo panorama. Obě tyto aplikace sdílejí stejný základ, který zahrnuji porovnání klíčových bodů dvou obrázků na základě jejich vzdálenosti. Panorama jde o několik kroků dál a zkoumá vztah mezi dvěma obrázky za pomoci nalezené společné části, která pak poskytuje informace k nalezení provedené transformace.

Tato práce se nejdříve zabývá SIFT algoritmem a popisem jeho teoretické funkčnosti, dále se popisuje jeho následné využití při tvorbě panoramatu. Samotný koncept tvorby panoramatu je zde rozepsán od nalezení společných bodů, přes metody, které přesněji určí proběhlou transformaci, až po finální spojení obrázků. V další části je ukázána výsledná funkčnost naimplementovaného SIFT algoritmu a jeho porovnání s knihovním SIFT od OpenCV. A posledně jsou ukázány výsledky vytvořeného algoritmu pro tvorbu panoramatu pro dva až tři obrázky.

Kapitola 2

Scale-Invariant Feature Transform

Uvažujme obrázek I, na kterém chceme najít body, které nejlépe určují daný obrázek. Cheme, aby tyto body zůstaly nezměněny i při změně velikosti nebo po provedení transformace na obrázku I. Tyto body se nazývají stabilní klíčové body a jsou výsledkem aplikace SIFT algoritmu na obrázek I.

2.1 Nalezení kandidátů na klíčové body

Prvním krokem k nalezení stabilních bodů je nalezení lokálních extrémů uvnitř množiny obrázků, která se vytvoří z originálního obrázku. Tato množina se nazývá scale-space. Daný scale-space se vytvoří z originálního obrázku, který je dvakrát zvětšen, to zaručí nalezní alespoň o čtyřikrát více bodů. Tento zvětšený obrázek se nazývá základní obrázek.

2.1.1 Výpočet počtu oktáv

Počet vrstev neboli oktáv, udává kolikrát lze základní obrázek zmenšit, za předpokladu, že finální obrázek bude mít velikost strany alespoň 1, tedy splňuje nerovnost

$$1 \leq \frac{l}{2^n}, \tag{2.1}$$

kde l je nejkratší strana a n je počet oktáv. Dále se základní obrázek zmenší n-krát, tím vznikne nová množina obrázků, kde na každé oktávě se provede opakovaně konvoluce.

2.1.2 Rozmazání na oktávě

Množina o velikosti počtu oktáv skládající se z obrázků, kde na každé oktávě je obrázek o polovinu menší než na oktávě předchozí, se nazývá scale-space. Každou oktávu uvnitř scale-space je třeba rozmnožit o stejný počet stejných obrázků, tím v každé oktávě vznikne několik vrstev. Následně je důležité pro každou oktávu rozmazat každý obrázek přes Gaussův filtr G

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

kde x a y jsou souřadnice bodu na obrázku, který použitím filtru změní svoji hodnotu, neboli se rozmaže.

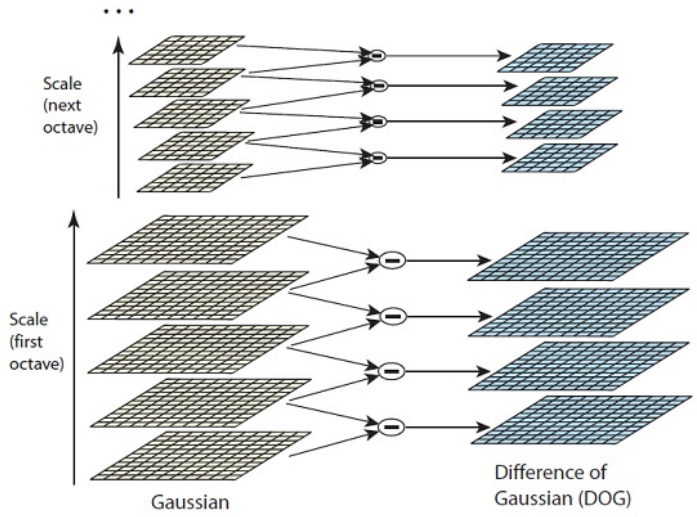
To se aplikuje na celý obrázek

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.3}$$

kde L je výsledný rozmazaný obrázek, G je operátor Gaussovského filtru, I je obrázek, na který filtr aplikujeme a $*$ je konvoluční operace v x a y. Důležitou hodnotou je zde sigma σ, která udává, jak moc se bod rozmaže, čím vyšší, tím větší je rozmazání. Je nutné si vytvořit množinu hodnot σ, které se opakovně aplikují na každou oktávu.

2.1.3 Výpočet DOG

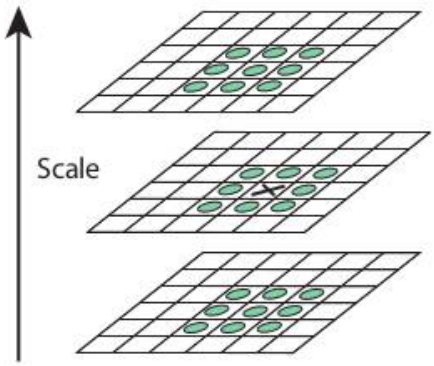
Vytvořením množiny obrázků, rozmazaných skrze Gaussův filtr, nám umožňuje vytvořit obrázky, na kterých budeme hledat lokální extrémy. Ty se vytvářejí přes DOG „Difference of Gaussian“. Jde o rozdíl mezi dvěma sousedními obrázky na dané oktávě. Vždy se odečte obázek s obrázkem na vrstvě vyšší a nižší. Princip této metody je vidět na Obrázek 2.1.



Obrázek 2.1: Výpočet DOG z rozmazaných obrazů [Lowe, 2004]

2.1.4 Detekce lokálních extrémů

Aby se dalo říct, že daný bod je lokální minimum či maximum, tak se musí porovnat se sousedy na svém 3×3 okolí a také na 3×3 okolí s obrázkem o vrstvu nižší a vyšší. Bod, který se porovnává, je středem dané 3×3 matice. Bod, jehož hodnota je nižší, než určitá hranice je přeskočen, protože poskytuje příliš malou informační hodnotu. Porovnání je ukázáno na Obrázek 2.2.



Obrázek 2.2: Určení lokálního extrému [Lowe, 2004]

2.2 Lokalizace klíčových bodů

Jakmile jsou kandidáti vybráni skrze porovnání vůči sousedům je nutné je lepé zařadit do prostoru na jejich skutečnou pozici. Během tohoto procesu se vyřadí body s nízkým kontrastem např. šum vzniklý z rozmazaní nebo body lokalizované kolem okrajů obrázku, které jsou nežádoucí pro další práci.

2.2.1 Určení pozice klíčových bodů

Kvůli možnosti, že vybraný kandidát neleží na pozici skutečného extrému, je třeba pozici tohoto hledaného extrému aproximovat. Pozice klíčového bodu (hledaného extrému), z nalezeného kandidáta, se určuje za pomoci Taylorova polynomu druhého řádu funkce $D(x, y, \sigma)$, za jejíž počátek se považuje pozice kandidáta, použitím vzorce

$$D(\mathbf{x}) = D + \frac{\partial D^\intercal}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^\intercal \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}, \tag{2.4}$$

kde funkce D a její derivace jsou vyhodnoceny ve zkoumaném bodě (kandidátovi) a $\mathbf{x} = (x, y, \sigma)^\intercal$ pozice kandidáta.

Pozice hledaného extrému, $\hat{\mathbf{x}}$, je určena pomocí derivace funkce D podle $\mathbf{x}$ položené rovno nule

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}}. \tag{2.5}$$

Pokud vzdálenost $\hat{\mathbf{x}}$ od $\mathbf{x}$ je větší než 0.5 v jakémkoliv směru, znamená to, že nalezený extrém leží blíže jinému bodu. V tomto případě změníme vyšetřovaný bod a v nalezeném bodě se provede interpolace.

Funkční hodnota nově nalezeného extrému $D(\hat{\mathbf{x}})$ je užitečná pro odmítání nestabilních extrémů s nízkým kontrastem. Této hodnoty docílíme dosazením 2.5 do 2.4 a tím získáme

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^\intercal}{\partial \mathbf{x}}\hat{\mathbf{x}}. \tag{2.6}$$

Pokud tedy funkční hodnota $D(\hat{\mathbf{x}})$ je menší než zvolené minimum, daný bod se zahodí.

2.2.2 Eliminace hran a ploch

Nestačí pouze vyřadit body s nízkým kontrastem, jelikož body kolem okraje obrázku budou mít vysoký kontrast, a tudíž projdou testem. Tyto body nesou malé množství informace a proto je třeba je odstranit. Pomocí Harrisova detektoru určíme tři případy:

1. *Rovná plocha* - navzájem kolmé směrové derivace jsou malé, skoro nulové, to značí minální růst v obou směrech, takovýto bod je nežádoucí, protože se může jevit jako shoda pro spousty rozdílných bodů.
2. *Okraj obrázku* - jedna ze dvou směrových derivací je výrazně menší, to značí malý růst v jednom směru, nachází se tedy v okolí okraje obrázku

3. *Roh* - obě směrové derivace jsou velké, toto je velice užitečná informace a patří k nejlepší klíčovým bodům na obrázku

První dvě možnosti jsou nežádoucí a takové body se musí odstranit z množiny. K tomu, abychom určili o jakou možnost jde, je potřeba Hessova matice $\mathbf{H}$, která je spočítaná následovně z funkce D

$$\mathbf{H} = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix}. \tag{2.7}$$

Pomocí vlastních čísel matice $\mathbf{H}$ určíme danou situaci. Vlastní čísla jsou však výpočetně náročná, naštěstí se tomu dá vyhnout díky již vypočítaným derivacím funkce D a tomu, že je pro nás důležitější poměr mezi vlastními čísly než samotná hodnota vlastních čísel. Tudíž je snažší spočítat stopu a determinant, kvůli jejich vztahu s vlastními čísly. Nechť je α největší vlastní hodnota v absolutní hodnotě a β druhá vlastní hodnota, pak

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta, \tag{2.8}$$

$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \tag{2.9}$$

Nechť r je poměr mezi dvěma vlastními čísly, že $\alpha = r\beta$. Pak

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}, \tag{2.10}$$

což ukazuje, že poměr mezi dvěma vlastními čísly je pro nás užitečnější než jejich přesné velikosti. Tudíž, když chceme zjistit, zda je poměr pro libovolné $\mathbf{H}$ pod určitou konstantní hodnotou r, tak stačí kontrolovat

$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}. \tag{2.11}$$

2.3 Přiřazení orientace

V tomto bodě jsou nalezené již všechny adekvátní klíčové body, je tedy nutné jim přiřadit orientaci, která přidává rotační invarianci klíčovému bodu. To se provádí nasbíráním směru gradientů a jejich velikostí kolem každého klíčového bodu. To nám pomůže vybrat nejlepší orientaci či orientace v daném okolí, které se pak přiřadí klíčovému bodu.

Výpočet velikosti gradientu m a orientace θ se provádím pomocí vztahů ve tvarech

$$m\left(x,y\right) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \tag{2.12}$$

$$\theta\left(x,y\right) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right), \tag{2.13}$$

vše se to počítá jako rozdíl pixelů v okolí klíčového bodu (x, y) na Gaussovsky rozmazaném obrázku L, který náleží danému σ.

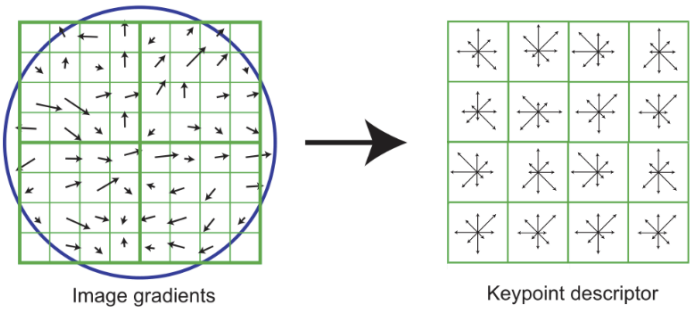
Výsledné hodnoty se uloží do histogramu s 36 třídami, pro rozdělení 360 stupňů orientace. Podle hodnoty orientace se do daného bloku histogramu přidá hodnota úměrná velikosti gradientu v daném bodě, to nám utvoří vrcholy v histogramu. Jakýkoliv vrchol, který je větší než 80% nejvyššího vrcholu, se převede na nový klíčový bod. Díky tomu můžeme dostat více výsledných klíčových bodů, než jsme měli, což je dobře, protože více bodů znamená přesnější informace.

2.4 Vytvoření deskriptoru klíčového bodu

Nakonec se vytvoří jednotlivým klíčovým bodům jejich deskriptor, který zakóduje informace o okolí klíčového bodu a dovolí nám tak lepší porovnání s ostatními klíčovými body.

Proces tvorby deskriptoru spočívá ve znovuvytvoření histogramu gradientu orientací. Použije se čtvercové okolí klíčového bodu, ale teď dané okolí o rotujeme o úhel klíčového bodu. Z tohoto okolí vybereme sloupce a řádky, které nám indikují, kde každý pixel v okolí leží. Dále pro každý pixel se spočítá velikost gradientu a jeho orientace. Místo uložení do histogramu s 36 bloky, tentokrát bude histogram mít pouze 8 bloků na pokrytí 360 stupňů.

V dalším kroce, z našeho 2D okolí, utvoříme vektor o délce 36. Orientace asociované s každým pixelem se roztřídí do daného vektoru, ale na jejich pozici nyní ještě přidáme váženou velikost gradientu pro daný pixel. Tím vznikne 3D okolí, které je třeba zploštit, aby vznikl vektor deskriptoru o velikosti 128. Tento vektor se musí ve finále normalizovat, a tím se získá finální deskriptor popisující dané okolí a identifikuje klíčový bod.

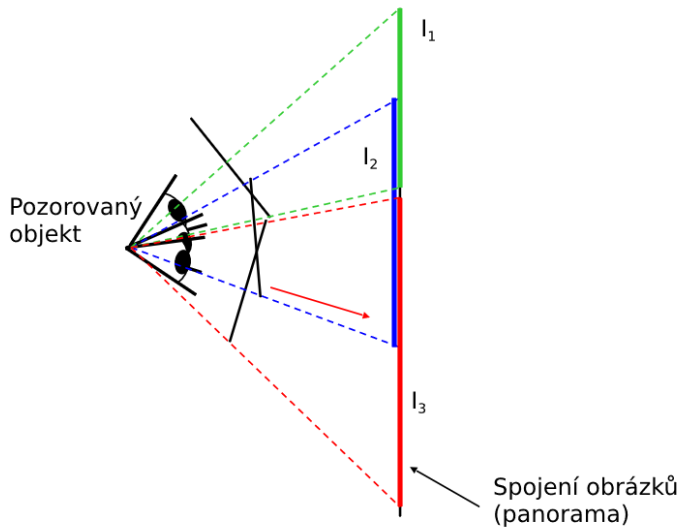


Obrázek 2.3: Převod orientace pixelů v okolí na deskriptor [Lowe, 2004]

Kapitola 3

Spojení obrázků (Panorama)

Chceme spojit dva a více obrázků do jednoho panorama. Pro jejich spojení je důležité zjistit jejich společnou část. Toho se docílí pomocí SIFT algoritmu, který je popsán v předchozí kapitole. Po nalezení společné části využijeme klíčové body obrázku I_1 a I_2 ke zjištění provedené transformace.



Obrázek 3.1: Princip správného spojení obrázků [Collins et al., 2017]

3.1 Transformace obrazu

K nalezení transformace, aplikované na obrázek I_2, je potřeba si uvědomit, jaké transformace umožňuje dosavadní popis obrázku I_2. Obrázek I_2 je popsán množinou bodů $\mathbf{p}$, kde $\mathbf{p} \in \mathbb{R}^2$, se souřadnicemi (x, y). To omezuje transformace na sadu jednoduchých lineárních zobrazení jako je např. rotace, zvětšení či zrcadlení, které mají tento tvar

$$\mathbf{p}_2 = \mathbf{T}\mathbf{p}_1 \qquad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \qquad \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \quad (3.1)$$

Důležitá transformace, která ale tímto způsobem definovat nelze je posun obrázku o konstantu t_x a t_y, protože soustava

$$x_2 = x_1 + t_x \qquad y_2 = y_1 + t_y, \quad (3.2)$$

nemá řešení pro matici $\mathbf{T}$ velikosti 2×2. To se dá vyřešit převedením $\mathbf{p}$ do homogenních souřadnic.

3.2 Homogenní souřadnice

Homogenní neboli projektivní souřadnice, je souřadnicový systém, který se používá pro projektivní geometrii. Jejich výhodou je, že souřadnice všech bodů, včetně bodů v nekonečnu, mohou být reprezentovány konečnými souřadnicemi. Převod bodů $\mathbf{p}$, obrázku I_2, z euklidovské roviny do projektivní roviny, se provede přidáním jedné souřadnice, tím vznikne $\tilde{\mathbf{p}} = (x, y, 1)$. Vzniklé souřadnice jsou invariantní vůči škálování, jelikož bod $(x, y, 1) = (\tilde{z}x, \tilde{z}y, \tilde{z})$, pro $\tilde{z} \neq 0$. Bod (0, 0, 0) je nepovolený a bod $(0, 0, 1)$ je počátek homogenní souřadnicové soustavy. Převod zpět do kartézských je proveden následnovně

$$(\tilde{z}x, \tilde{z}y, \tilde{z}) \Leftrightarrow (\frac{x}{\tilde{z}}, \frac{y}{\tilde{z}}) \quad (3.3)$$

Nyní je soustava rovnic pro posun řešitelná a dostáváme

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \quad (3.4)$$

3.3 Afinní transformace

Lineární transformace společně s posunem tvoří afinní transformace. Afinní tranforamce jsou speciální případ projektivní transformace, kde poslední řádek matice transformace má tvar $\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$. Afinní transformace mají tyto vlastnisti

1. Úsečky se mapují na úsečky.
2. Parelelní úsečky zůstanou paralelní.

3. Počátek se nemusí mapovat na počátek.
4. Zachovávají poměr délek dvou paralelních úseček.
5. Umožňují jen šest stupňů volnosti.

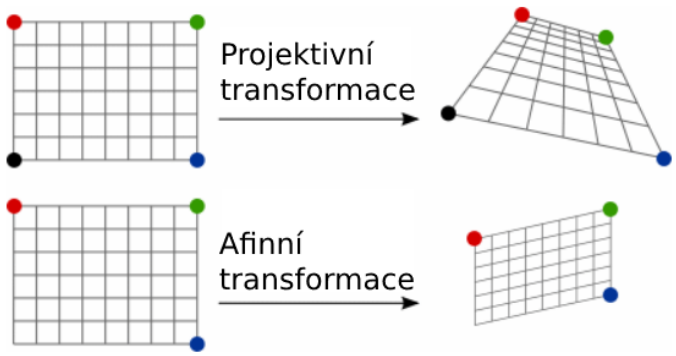
3.4 Projektivní transformace

Projektivní transformace poukazuje na to, jak se mění vnímání objektů, když se mění úhel, pod kterým je objekt pozorován. Samotné afinní transformace jsou speciálním případem projektivní transformace. Matice projektivní transformace je tvaru

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & 1 \end{pmatrix} \tag{3.5}$$

Kde $\mathbf{A}_{2\times2}$ je matice, která určuje jaká transformace se provádí např. rotace či zrcadlení. $\mathbf{B}_{2\times1}$ je matice posunu, určující posun obrázku v určitém směru. Poslední částí je matice projekce $\mathbf{C}_{1\times2}$, která je pro afinní transformace vždy rovna $\mathbf{0}$.

Na rozdíl od afinní transformace, projektivní transformace nezachovává paralelitu úseček a ani poměr jejich délek, dále z 3.5 je poznat, že nabízí osm stupňů volnosti.

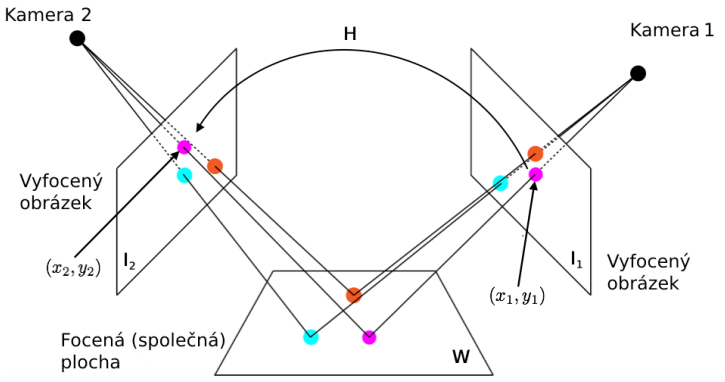


Obrázek 3.2: Rozdíl mezi projektivní a afinní transformací [GraphicsMill]

3.5 Homografie

Pokud se odstraní podmínka, které říká, že poslední řádek pro afinní transformace je roven $\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$, tak vznikne matice homografie. Homografie je druh transformace, která mapuje rovinu na jinou rovinu skrze určitý bod projekce. Máme tedy obrázky I_1 a I_2, které jsou zobrazením objektu W pod různými

úhly pozorování, ale skrze stejný bod projekce, pak tedy I_1 a I_2 jsou spojené homografií $\mathbf{H}$.



Obrázek 3.3: Ukázka homografie [Qureshi, 2021]

Matice homografie $\mathbf{H}_{3\times3}$ nemá zafixován poslední prvek na jedničku a její tvar je tedy

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{3.6}$$

I přesto kvůli podmínce $\sqrt{\sum(h_{ij})^2} = 1$ je zde opět osm stupňů volnosti.

3.5.1 Výpočet homografie

Ke zjištění homografie $\mathbf{H}$ mezi dvěma body $\mathbf{p}_1$ a $\mathbf{p}_2$ je třeba vyřešit

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z} \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \tag{3.7}$$

kde hodnoty x_2 a y_2 se vyjádří jako

$$x_2 = \frac{\tilde{x}_2}{\tilde{z}} = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \tag{3.8}$$

$$y_2 = \frac{\tilde{y}_2}{\tilde{z}} = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \tag{3.9}$$

Po přerovnání vznikne

$$x_2(h_{31}x_1 + h_{32}y_1 + h_{33}) = h_{11}x_1 + h_{12}y_1 + h_{13} \tag{3.10}$$

$$y_2(h_{21}x_1 + h_{22}y_1 + h_{23}) = h_{11}x_1 + h_{12}y_1 + h_{13} \tag{3.11}$$

A z tohoto vyjádření se řeší rovnice

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_2x_1 & -x_2y_1 & -x_2 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_2x_1 & -y_2y_1 & -y_2 \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.12}$$

Toto se opakuje pro všechny body z I_1 a I_2, které byly vyhodnoceny jako shodné dvojice za pomoci SIFT. Vzniká tedy soustava

$$\mathbf{Ah} = \mathbf{0} \tag{3.13}$$

kde $\mathbf{A}$ je matice známých bodů, $\mathbf{h}$ je matice neznamých, pro které se hledá řešení, a to celé za podmínky $\|\mathbf{h}\|_2^2 = 1$.

Kvůli dané podmínce se musí přistoupit k metodě řešení přes podmíněné nejmenší čtverce, z toho vziká tento problém

$$\hat{\mathbf{h}} = \underset{\|\mathbf{h}\|_2=1}{\arg\min} \|\mathbf{Ah}\|_2^2 \tag{3.14}$$

Zadefinuje se tedy účelová fuknce $L(\mathbf{h}, \lambda)$

$$L(\mathbf{h}, \lambda) = \mathbf{h}^T\mathbf{A}^T\mathbf{Ah} - \lambda(\mathbf{h}^T\mathbf{h} - 1) \tag{3.15}$$

která vrací $\mathbf{A}^T\mathbf{Ah} = \lambda\mathbf{h}$. Řešením 3.14 je tedy vlastní vektor $\mathbf{h}$ nejmenšího vlastního čísla λ matice $\mathbf{A}^T\mathbf{A}$.

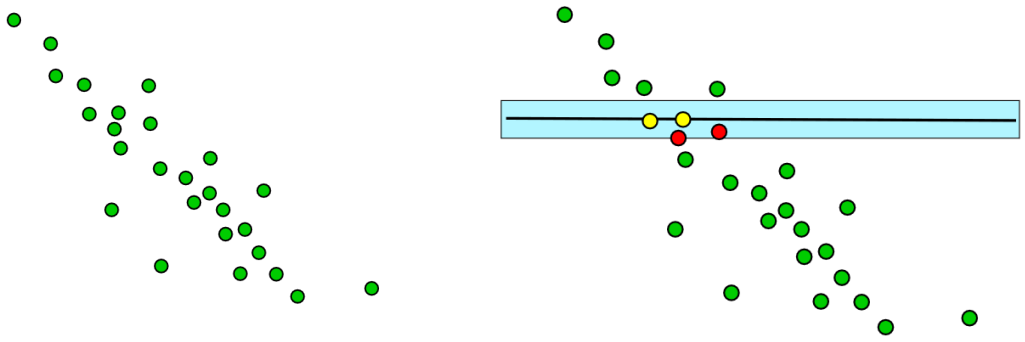
3.6 RANSAC

Největší problém při výpočtu homografie je, že je závislá na množině dvojic bodů vzniklých za pomoci SIFT, kde mohlo dojít k chybám a body, které se tvářily vzdáleností jako vhodní kandidáti, tak jsou ve skutečnosti naprosto rozdílné body. Pokud těchto chybných bodů je méně než 50% z celkového počtu bodů, tak se využije RANSAC metody. Ta tyto chybné body pomáhá odstranit, aby při výpočtu homografie nezpůsobyly chybné výsledky.

3.6.1 Průběh RANSAC algoritmu

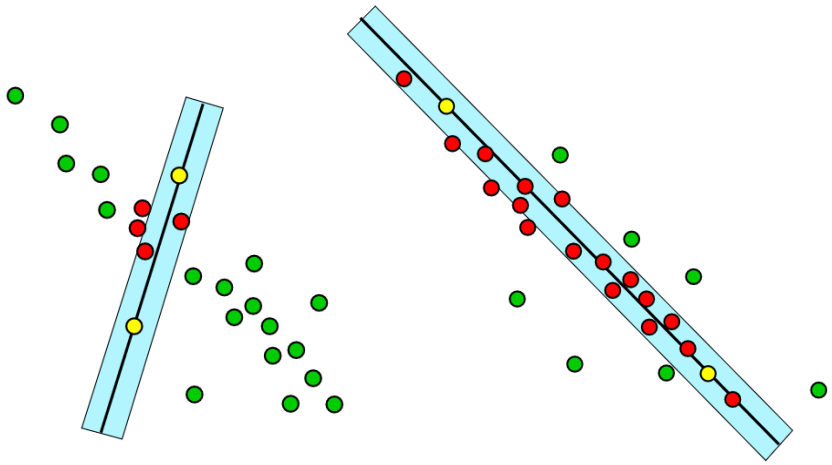
1. Náhodně vybere s prvků, kde s je nejmenší požadované množství pro daný model. Pro homografii je $s = 4$, protože to je minimun dvojic potřebných k výpočtu homografie.
2. Vypočítá daný model s náhodně vybranými daty, v tomto případě homografii.
3. Spočítá počet bodů M, které splňují daný model, neboli porovná vzdálenosti výsledných bodů s body na obrázku I_1. Pokud je vzdálenost menší než akceptovatelná chyba ε, tak daný bod splňuje model.
4. Krok 1-3 se opakuje N-krát.
5. Po skončení se vybere model s největším počtem M.

Na konec se vypočte nová homografie, obrázek I_1 se spojí s výsledným I_3, který vznikl aplikací homografie na I_2.



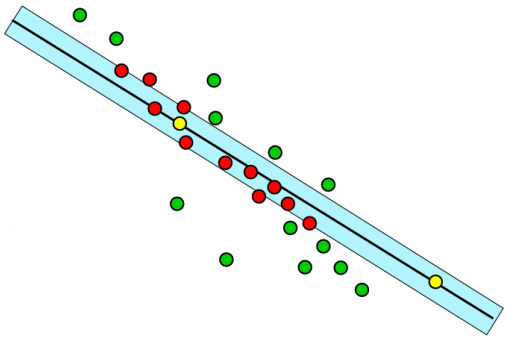
(a) : Množina bodů

(b) : První iterace: M = 4



(c) : Druhá iterace: M = 6

(d) : Třetí iterace: M = 19



(e) : Čtvrtá iterace: M = 13

Obrázek 3.4: Ukázka několika iterací RANSAC algoritmu, metoda: line fitting [Collins, 2007]

Kapitola 4

Výsledky

V této kapitole se prvně předvede SIFT algoritmus a jeho průběh nalezení klíčových bodů na zvoleném obrázku I_1, a poté shoda těchto klíčových bodů s body z obrázku I_2. Výsledky porovnání budou srovnány s implementací SIFT z knihovny OpenCV. Následně bude ukázán průběh tvorby panaromatu za pomoci SIFT algoritmu, projektivní transformace a homografie.

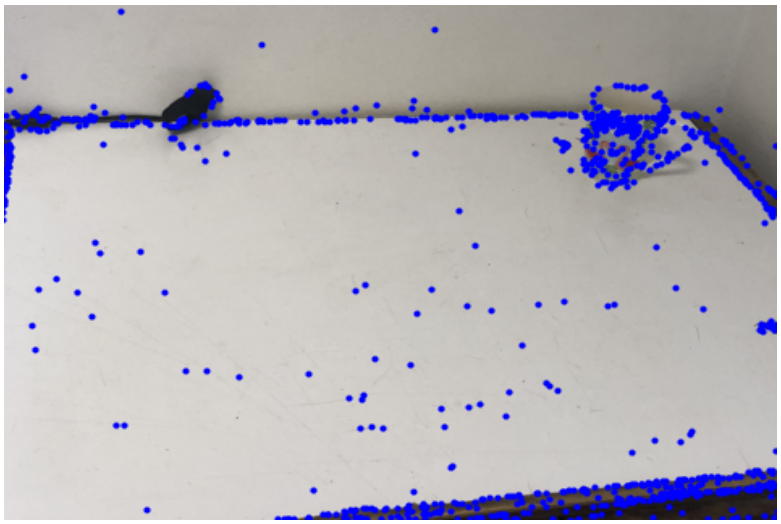
4.1 SIFT algoritmus

Obrázek 4.1: Vybraný obrázek

4.1.1 Průběh nalezení klíčových bodů

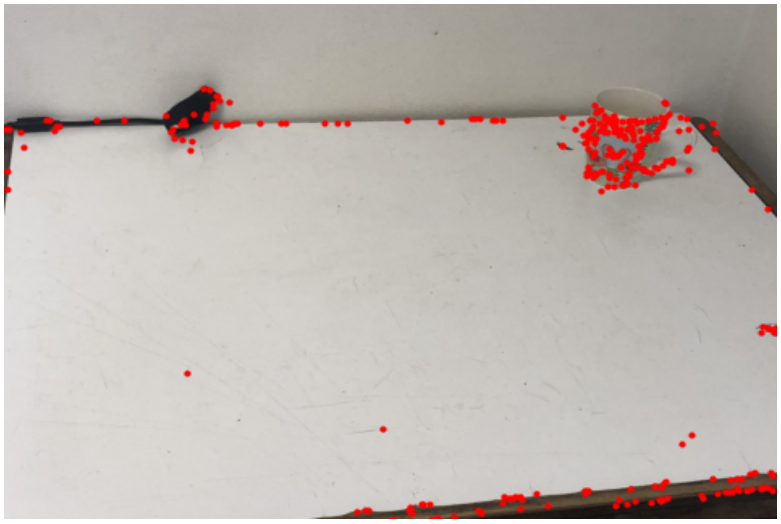
Po zvolení základního obrázku se provede, dle principu 2.1, výpočet oktáv, aplikace Gaussova filtru a nakonec DOG. Na této vzniklé množině hledáme

vhodné kandidáty na klíčové body, 2.1.4, viz Obrázek 4.2. Vzhledem k spoustě rovných ploch budeme očekávat velké množství nerelevantních kandidátů, kteří budou v dalších krocích odstraněni.



Obrázek 4.2: Kandidáti na klíčové body, počet nalezených kandidátů je 1304

Po získání množiny kandidátů je třeba určit skutečnou pozici klíčových bodů, 2.2, odstranit body s nízkým kontrastem a body kolem okrajů obrázku. Tím nám vznikne množina klíčových bodů, které jsou invariantní vůči změně velikosti. Na Obrázek 4.3 je vidět, že výsledný počet lokalizovaných klíčových bodů je o dost menší než byl původní počet kandidátů, především díky odstranění bodů na hranách a na desce stolu (rovná plocha).



Obrázek 4.3: Lokalizované klíčové body, počet klíčových bodů je 293

Dále je třeba přiřadit klíčovým bodům jejich orientaci, 2.3, abychom si zajistili invarianci vůči transformaci. Pozice klíčových bodů zůstává stejná jako na

Obrázek 4.3, a to proto, že nové klíčové body nemají jinou pozici, ale pouze různou orientaci. Proto tedy na jedné pozici může být více klíčových bodů lišících se pouze orientací nikoliv svojí pozicí. V tomto případě se počet zvedl z 293 na 335.

V poslední fázi je třeba vytvořit deskriptory, 2.4, výsledný přiblížený obrázek s deskriptory je vidět na Obrázek 4.4. V ideálním případě bychom na obrázku chtěli vidět co nejvíce deskriptorů s malým okolím, kvůli maximalizaci přesnosti.

Obrázek 4.4: Zoom na výsledné deskriptory klíčových bodů

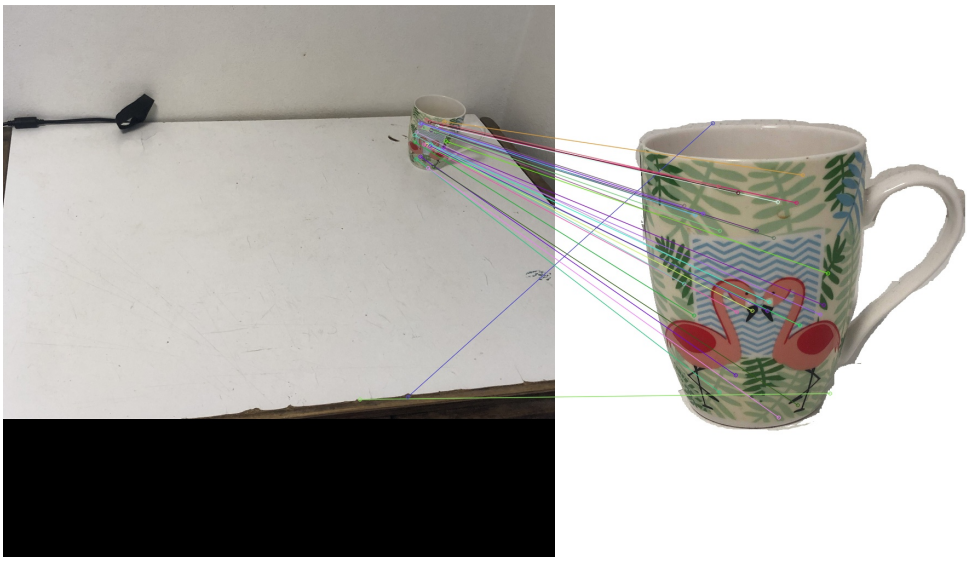
4.1.2 Hledání objektu na obrázku

V této části zkoušíme schopnost SIFT algoritmu, při hledání objektu na obrázku. Porovnání probíhá na základě poměření vzdáleností deskriptorů klíčových bodů mezi Obrázek 4.5*b* a Obrázek 4.4. Je nutno podotknout, že krátká vzdálenost mezi deskriptory neimplikuje shodu, pouze zvyšuje šanci na shodu, tudíž můžeme mít mnoho rozdílných bodů, které se budou tvářit jako stejné.

(a) : Hledaný objekt

(b) : Výsledné deskriptory klíčových bodů hledaného objektu

Výsledky porovnání klíčových bodů na Obrázek 4.6 ukazují přesnost tohoto algoritumu. Malý počet bodů se zde namapoval na nesprávný objekt a to z důvodu nepřesného oříznutí hledaného objektu a mylných shod vzdáleností.



Obrázek 4.6: Výsledek hledání objektu vpravo na obrázku vlevo

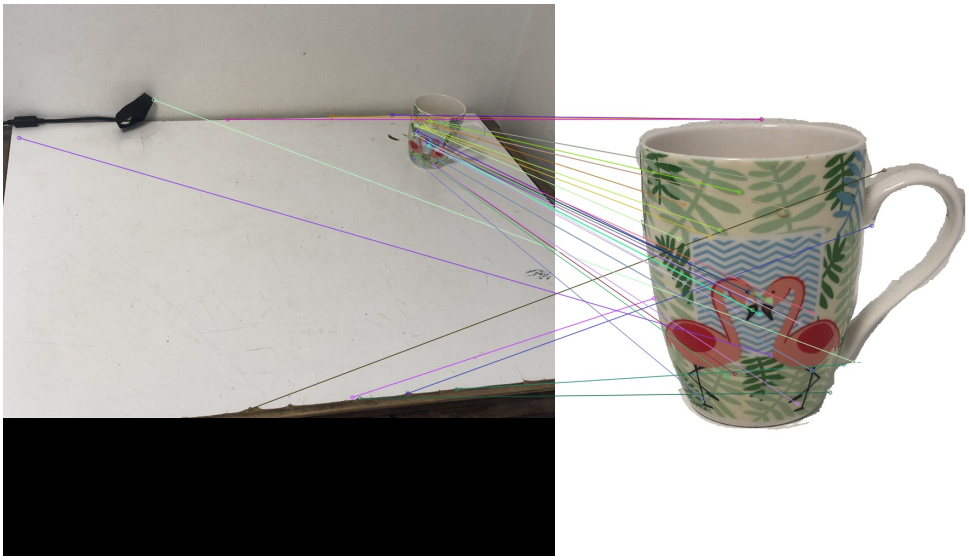
4.1.3 Porovnání s OpenCV implementací SIFT

Tato část zkoumá správné fungování implementovaného SIFT algoritmu oproti knihovnímu SIFT algoritmu od OpenCV.

	Počet klíčových bodů		Nalezené shody
	Obrázek 1	Obrázek 2	
Naimplementovaný SIFT	333	1253	32
OpenCV SIFT	263	1023	38

Lehce odlišné hodnoty ve výsledcích vypovídají o přesnějším určování klíčových bodů na straně OpenCV knihovny a také lepším popisu deskriptoru, to je zase vidět na vyšším počtu shod. Celkově jsou výslekdy ale velice podobné, což znaží o správném fungování SIFT algoritmu implementovaného v této práci.

Obrázek 4.7: Klíčové body s deskriptory nalezené za pomoci OpenCV



Obrázek 4.8: Klíčové body s deskriptory nalezené za pomoci OpenCV

4.2 Tvorba panorama

Panorama je jedna z aplikací využívající SIFT algoritmu. Zde se konkrétně využívá při výpočtu homografie pro tranformaci obrázku, přesněji při určování společné části mezi dvěma obrázky a následné tvorby matice transformace z nalezených bodů.

4.2.1 Tvorba panorama ze dvou obrázků

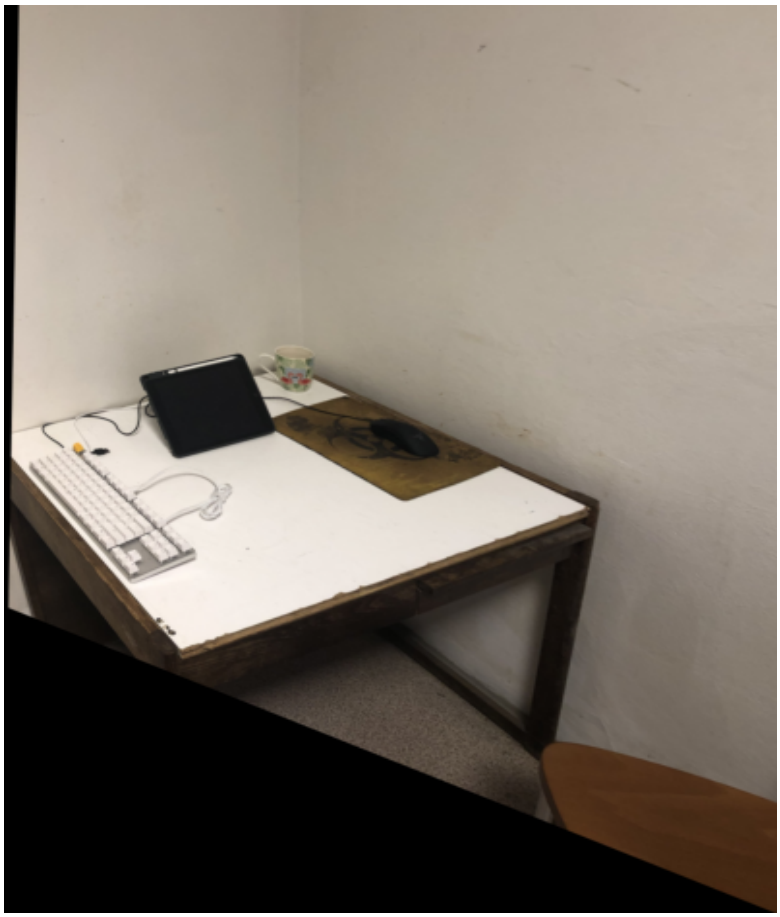
Prvním krokem spojení dvou obrázků, Obrázek 4.9*a* a Obrázek 4.9*b*, do panoramatu je nalezení klíčových bodů za pomoci SIFT algoritmu, ze kterých utvoříme algoritmem na nalezení dobré shody, dle vzdálenosti klíčových bodů, množinu shodných klíčových bodů. Body z obou obrázků, uvnitř této množiny, použijeme pro výpočet matice homografie **H**.

Budeme spojovat Obrázek 4.9*b* s 4.9*a* tak, že aplikujeme matici **H** na Obrázek 4.9*b*. Výsledkem je Obrázek 4.10, transformovaný obrázek v nových souřadnicích. Poslední krok je přiložení Obrázek 4.9*a* zleva k nově transformovanému obrázku.

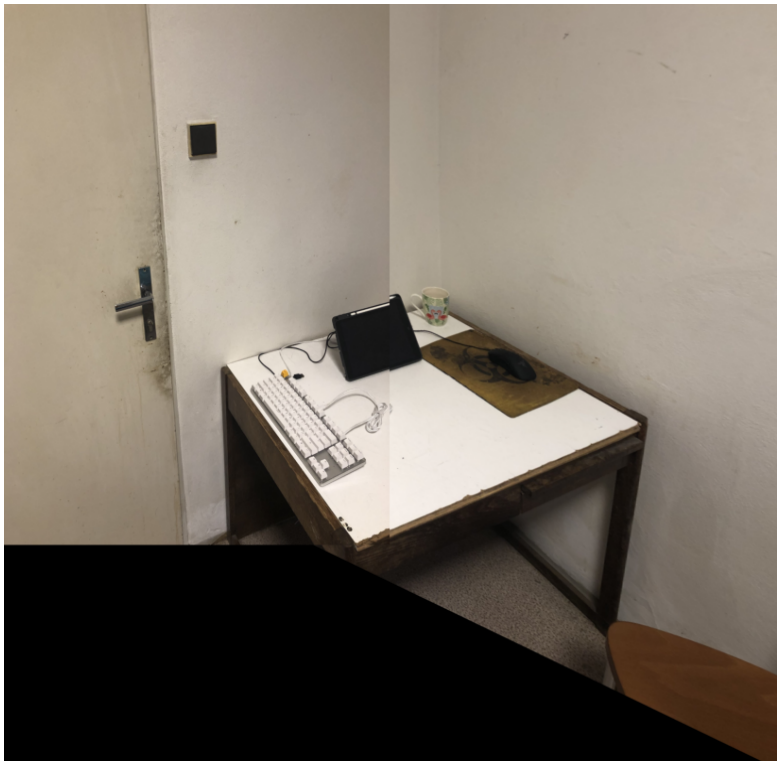
(a) : Levý obrázek

(b) : Pravý obrázek

Obrázek 4.9: Obrázky ke spojení



Obrázek 4.10: Transformovaný obrázek 4.9*b*

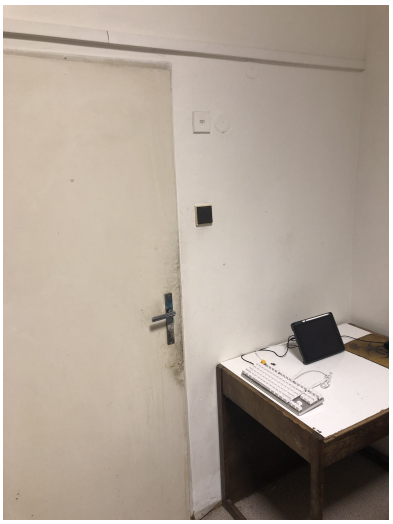


Obrázek 4.11: Výsledné spojení obrázků (panorama)

4.2.2 Tvorba panorama ze tří obrázků

Spojení tří či více obrázků probíhá podobně jako spojení dvou. Prvním rozdílem je porovnání shod mezi všemi obrázky, čímž určíme, které dva se spojí nejdříve, to se opakuje dokud se nespojí všechny obrázky. Aplikace této metody je nestabilní a největší úspěšnost je pouze pro dva obrázky, čím více obázků je součástí panaromatu, tím větší chyba vznikne. Chyba vzniká z důvodu komplikovaného tvaru nově vzniklých obrázků po transformaci a spojení, načež matice **H** není přesná při výpočtu nových souřadnic a obrázek se více deformuje.

(a) : Levý obrázek



(b) : Prostřední obrázek

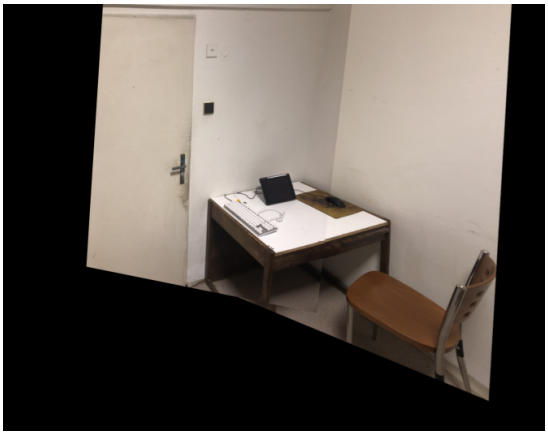
(c) : Pravý obrázek

Obrázek 4.12: Obrázky ke spojení

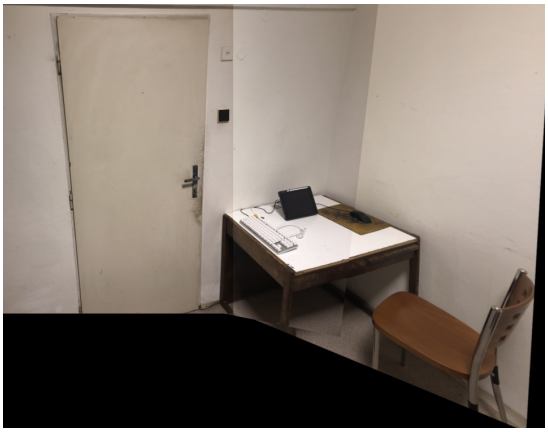
(a) : Transformovaný 4.12*c*

(b) : Spojení s 4.12*b*

Obrázek 4.13: Krok 1: Transformace 4.12*c* a spojení s 4.12*b*



(a) : Transformace 4.13*b*



(b) : Finální panorama

Obrázek 4.14: Krok 2: Transformace nově vzniklého 4.13*b* a finální spojení s 4.12*a*

Kapitola 5

Závěr

Různé velikosti a úhly natočení zkoumaného obrázku tvoří spousty problémů pro algoritmy rozpoznávání obrazu. Tento problém částečně řeší SIFT algoritmus, i když přilíš velká transformace o proti původnímu obrázku je problém i pro SIFT, přesto možnosti, které SIFT nabízí jsou velice přínosné pro počítačové vidění, jelikož umožňuje správnou shodu klíčového bodu s jiným, který je vybrán z velkého množství klíčových bodů. To je dosaženo díky deskriptoru klíčového bodu, který reprezentuje vlastnosti daného bodu a vlastnosti jeho okolí. Klíčový bod je navíc sám o sobě velice robustní, jelikož při jeho volbě se dbá na jeho invarianci vůči velikosti a jeho orientaci. Právě tyto vlastnosti dělají SIFT tak významným algoritmem.

Využitelnost SIFTu zde byla ukázána hlavně při tvorbě panoramatu, kde sloužil pro nalezení společné části dvou obrázků. Samotný popis algoritmu pro tvorbu panoramatu zde byl popsán od nalezní společné části až po určení správné homografie mezi nimi. Byla při tom hlavně využita projektivní transformace, která je velice dostatečná pro spojení dvou obrázků, ale pro větší množství bude akumulovat chybu v tvaru výsledného obrázku, je nutné tedy, pro vyšší počet obrázků, doplnit tento algoritmus o další metody, které tuto chybu minimalizují.

Výsledky obou naimplementovaných algoritmů byly ukázány v předchozí kapitole a i přes jejich zmíněné nedostatky byly uspokojivé a oba algoritmy dosáhly toho, co bylo jejich cílem. Tato práce nadále nabízí možnosti pro další zlepšení a samotné rožšíření schopností obou algoritmů.

Literatura

[Lowe, 2004] Lowe, David G., 2004, *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, **1**, 60(2):91–110.

[Lowe, 1999] Lowe, David G., 1999, *Object recognition from local scale-invariant features*, In International Conference on Computer Vision, Corfu, Greece, **2**, pp. 1150–1157.

[Witkin, 1983] Witkin, A.P., 1983, *Scale-space filtering. In International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, **3**, 1019–1022.

[Koenderink, 1984] Koenderink, J.J., 1984, *The structure of images*, Biological Cyber-netics, **4**, 50:363–396.

[Lindeberg, 1994] Lindeberg, T., 1994, *Scale-space theory: A basic tool for analysing structures at different scales.*, Journal of Applied Statistics, **5**, 21(2):224–270.

[Harris and Stephens, 1988] Harris, C. and Stephens, M., 1988, *A combined corner and edge detector*, In Fourth Alvey Vision Conference, Manchester, UK, **6**, pp. 147–151.

[Brown and Lowe, 2002] Brown, M. and Lowe, D.G., 2002, *nvariant features from interest point groups*, In British Machine Vision Conference, Cardiff, Wales, **7**, pp. 656–665.

[Schmid and Mohr, 1997] Schmid, C. and Mohr, R., 1997, *Local grayvalue invariants for image retrieval*, IEEE Trans. on Pattern Analysis and Machine Intelligence, **8**, 19(5):530–534.

[GraphicsMill] *Transformations Difference*, `https://www.graphicsmill.com/docs/gm/TransformationsDifference.png`, **9**.

[Edelman et al., 1997] Edelman, S., Intrator, N., and Poggio, T., 1997, *Complex cells and object recognition*, **10**.

[Schiele and Crowley, 2000] Schiele, B. and Crowley, J.L., 2000, *Recognition without correspondence using multidimensional receptive field histograms*, International Journal of Computer Vision, **11**, 36(1):31–50.

[Beis and Lowe, 1997] Beis, J. and Lowe, D.G., 1997, *Shape indexing using approximate nearest-neighbour search in high-dimensional spaces*, In Conference on Computer Vision and Pattern Recognition, Puerto Rico, **12**, pp. 1000–1006.

[Schiele and Crowley, 2002] Mikolajczyk, K. and Schmid, C., 2002, *An affine invariant interest point detector*, In European Conference on Computer Vision (ECCV), Copenhagen, Denmark, **13**, pp. 128–142.

[Bloomenthal and Rokne, 1994] Bloomenthal, J. and Rokne, J., 1994, *Homogeneous Coordinates*, The Visual Computer 11, **14**, pp. 15-26.

[Brown and Lowe, 2007] Brown, M., Lowe, D.G., 2007, *Automatic Panoramic Image Stitching using Invariant Features*, Int J Comput Vision 74, **15**, pp. 59–73.

[Agarwal et al., 2005] Agarwal, A., Jawahar, C. V., Narayanan, P. J., 2005, *A survey of planar homography estimation techniques*,Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12, **16**.

[Weisstein, 2004] Weisstein, E. W., 2004, *Affine transformation*, `https://mathworld.wolfram.com`, **17**.

[Qureshi, 2021] Qureshi, F., 2021, *Homogoraphy: A Case Study in Model Fitting*, `http://csundergrad.science.uoit.ca/courses/cv-notes/notebooks/19-homography.html`, **18**.

[Šír, 2020] Šír, Z., 2020, *Projektivní zobrazení a jejich aplikace*,`https://www2.karlin.mff.cuni.cz/~sir/soubory/ProjZobrazeniAaplikaceTEMP.pdf`, **19**.

[Collins et al., 2017] Collins, R., et al., 2017, *Homography, Transforms, Mosaics*, `http://www.cs.columbia.edu/~allen/F17/NOTES/homography_pka.pdf`, **20**.

[Collins, 2007] Collins, R., 2007, *Lecture 15 Robust Estimation : RANSAC*, `https://www.cse.psu.edu/~rtc12/CSE486/lecture15.pdf`, **21**.

[Derpanis, 2010] Derpanis, K. G., 2010, *Overview of the RANSAC Algorithm*, Image Rochester NY, 4(1), **22**, pp. 2-3.