

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

PLZEŇ, 2022

BC. VOJTĚCH ŠAFRÁNEK

PROHLÁŠENÍ

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

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V PLZNI

.....

Anotace

V diplomové práci se zabývám návrhem a tvorbou aplikace fotoaparátu pro mobilní zařízení s operačním systémem Android. Aplikace bude využívat klíčové body ve snímku pro různé typy úloh. Aplikace bude odstraňovat pohyb objektů z popředí, provádět super kompozici objektů z více scén do jedné a nakonec filtr obrazové sekvence, která ze záznamu sekvence snímků vyfiltruje rozmazané a opakující se snímky.

Klíčová slova

klíčové body, mobilní aplikace, odstranění objektů, kompozice, homografie, fotoaparát, filtr rozmazaných snímků, filtr opakujících se snímků, nové řešení

Annotation

In my diploma thesis I deal with the design and develop of a camera application for mobile devices with the Android operating system. The application will use the key points in the slides for different types of tasks. The application will remove the objects in movement from the foreground, perform super compositions of objects from multiple scenes into one, and finally filter of image sequences, which will remove blurred and repetitive images from the recorded images.

Keywords

key-points, mobile application, remove objects, composition, homography, camera, blurred image filter, repetitive image filter, new solution

Obsah

1 Úvod	7
1.1 Úvod do problematiky	7
1.2 Představení řešení	7
2 Metody a techniky	9
2.1 Významné body v obrazu	9
2.2 SIFT	9
2.2.1 Detekce extrému	10
2.2.2 Zpřesnění polohy významných bodů	12
2.2.3 Přiřazení orientace významným bodům	13
2.2.4 Sestavení deskriptoru	14
2.3 SURF	15
2.4 ORB	16
2.4.1 Extrakce charakteristických bodů	16
2.4.2 Sestavení deskriptoru	19
2.5 BRISK	19
2.5.1 Detekce významných bodů	20
2.5.2 Deskriptor klíčových bodů	21
2.6 AKAZE	23
2.6.1 Rychlá explicitní difúze	25
2.6.2 Detekce významných bodů	26
2.6.3 Deskriptor klíčových bodů	26
2.7 Porovnání technik	27
2.8 Homografie	31
3 Odstranění pohybu	34
3.1 Motivace	34
3.2 Homografie v Odstranění objektů	35
3.3 Odečítání pozadí	36
3.3.1 Časový průměr	37
3.3.2 Model Gaussovských směsí	37
3.3.3 Aplikace MoG	38
3.4 Rekonstrukce masky popředí	38
3.4.1 Matematická morfologie	38
3.4.2 Aplikace matematické morfologie	40
3.5 Kompozice	41

3.6	Optický tok	44
3.7	Vyhodnocení	46
3.7.1	Objektivní vyhodnocení	46
3.7.2	Porovnání podle MSE	46
3.7.3	Subjektivní validace na testovacích sadách dat	47
4	Super kompozice	51
4.1	Motivace	51
4.1.1	Širokoúhlý objektiv	51
4.1.2	Panorama	53
4.1.3	Problém sešívání obrazů	53
4.2	Kompozice	54
4.3	Porovnání metod	58
4.4	Výsledné snímky super kompozice	59
4.4.1	Kompozice vs. panorama	64
5	Filtr obrazové sekvence	66
5.1	Motivace	66
5.2	Filtr rozostřených fotek	66
5.3	Filtr opakujících se fotek	70
5.4	Implementace do aplikace	73
6	Závěr	74

Obrazová příloha

1	Diskrétní škálovaný prostor. [17]	10
2	Konstrukce Gaussovy pyramidy. [22]	11
3	Detekce lokálního extrému. [18]	11
4	Gradienty spočtené v okolí významného bodu.	14
5	Deskriptor metody SIFT. [23]	15
6	FAST algoritmus výběr okolí. [9]	17
7	Pyramida FAST algoritmu. [10]	18
8	Ilustrace orientace thety [11]	19
9	Detekce významného bodu v pyramidovém prostoru [27]	20
10	Vzorkovací maska [28]	22
11	Porovnání mezi gaussovským a nelineárním difúzním měřítkovým prostorem[32]	24
12	Porovnání LDB vs M-LDB [34]	27
13	Příklad obrázků na porovnání technik pro hledání klíčových snímků	28
14	Nalezené klíčové body ve snímku parku pro jednotlivé metody	29
15	Nalezené klíčové body ve snímku s billboardem pro jednotlivé metody	30
16	Nalezené klíčové body ve snímku katedrály pro jednotlivé metody	30
17	Vizualizace homografie [8].	31
18	Ukázka homografie [8].	33
19	Příklad snímků pro odstranění pohybujících se objektů.	35
20	Vykreslení nalezených klíčových bodů.	36
21	Vykreslení nalezených zarovnaných obrázků.	36
22	Vizualizace metody odečtu pozadí [39].	37
23	Vykreslení masky odečtení pozadí	38
24	Eroze vs. dilatace	39
25	Otevření vs. uzavření	40
26	Vykreslení masky odečtení pozadí	40
27	Vykreslení masky odečtení pozadí	41
28	Vykreslení masky odečtení pozadí	42
29	Záplata pro testovací obrázek	43
30	Finální snímek po odstranění pohybu	43
31	Posunutí objektu [36].	44
32	Detekce pohybu pomocí optického toku	45
33	Porovnání výstupního snímku se snímkem upraveným	47
34	Porovnání snímků před a po úpravě	48
35	Porovnání snímků před a po úpravě	48

36	Porovnání snímků před a po úpravě	49
37	Porovnání snímků před a po úpravě	49
38	Porovnání snímků před a po úpravě	50
39	Porovnání snímků před a po úpravě	50
40	Typy distorzí.	51
41	Soudkovitá distorze před opravou [41].	52
42	Opravená poduškovitá distorze [41].	53
43	Snímky pro super kompozici obrazu.	54
44	Levý a pravý obrázek pro spojení.	55
45	Levý a pravý obrázek po spojení.	56
46	Shodné klíčové body.	56
47	Perspektivní transformace.	57
48	Výsledný obrázek po sloučení dvou snímků.	57
49	Odpovídající výstupní obrázek.	58
50	Příklad super kompozice 1.	59
51	Příklad super kompozice 2.	60
52	Příklad super kompozice 3.	61
53	Příklad super kompozice 4.	62
54	Příklad super kompozice 5.	63
55	Příklad super kompozice 6.	64
56	Rozdíl super kompozice cs. panorama.	64
57	Rozdíl super kompozice cs. panorama.	65
58	Rozdíl super kompozice cs. panorama.	65
59	Obrazová data před vyfiltrováním.	68
60	Výsledek po aplikaci filtru na rozostření snímky.	70
61	Výsledek po aplikaci filtru na opakované snímky.	72
62	Grafické rozhraní aplikace.	73

Tabulková příloha

1	Počet nalezených klíčových bodů v jednotlivých obrázcích	28
2	Čas uběhlý při získávání klíčových bodů	29
3	Průměrný čas potřebný na nalezení klíčového bodu	29
4	Čas potřebný na nalezené klíčových bodů ve všech snímcích.	35
5	Porovnání rychlostí	45
6	MSE skóre	47
7	Porovnání přesnosti technik	58
8	Příklad na počet klíčových bodů ve snímku.	67
9	Příklad na výsledky po filtraci.	67
10	F1 skóre pro různé nastavení prahu u jednotlivých technik.	69
11	Vybrané snímky, co mají být vyfiltrovány.	71
12	F1 skóre pro různé nastavení prahu u jednotlivých technik.	71

Seznam zkratek

- ang. angličtina
- č. číslo
- obr. obrázek
- SIFT Scale-Invariant Feature Transform
- SURF Speeded Up Robust Feature
- FAST Features from Accelerated and Segments Test
- BRIEF Binary robust independent elementary feature
- rBRIEF BRIEF aware rotation
- ORB Oriented FAST and Rotated BRIEF
- BRISK Binary Robust Invariant Scalable Keypoints
- AGAST Adaptive and Generic Accelerated Segment Test
- FED Fast Explicit Diffusion
- API Application Programming Interface

1 Úvod

1.1 Úvod do problematiky

Na trhu existuje mnoho aplikací, které jsou schopny zdatně zpracovat snímky pořízené fotoaparátem mobilního telefonu.

Jako konkurenční aplikace z hlediska přístupu k odstraňování objektů ve scéně v prostředí Android jsou na trhu k dispozici aplikace, které se snaží pomocí umělé inteligence dokreslit scény po ruční detekci a vymazání objektů ze scény. Příkladem tomu může například být aplikace nesoucí název *Retouch* [3], která má demo verzi nebo placenou plnou verzi.

Pro kompozice snímku do jednoho existují alternativy v podání nativní funkce panorama, kterým disponuje většina chytrých telefonů na trhu. Tato funkce má však nevýhodu v tom, že dokáže zaznamenat pouze horizontální scénu nebo vertikální. Existují aplikace, které tuto funkci rozšiřují a jsou schopny zaznamenat pruh scény v rozsahu 360 ° kolem uživatele příkladem může být aplikace *Panorama360Camera* [4] nebo může být zmíněna aplikace pro sférické fotografie nabízející přímo společností Google stojící za operačním systémem Android *GoogleStreetView* [5].

V neposlední řadě pro pořizování sekvencí fotek je možné považovat sportovní režim v nativních aplikacích fotoaparátů nebo v aplikacích třetích stran jako je tvorba snímků z videa, kterou nabízí aplikace *PhotosToVideo* [6]. Ta umožní uživateli pořídit sekvenci fotek, kterou pak následně uživateli rovnou uloží. Nevýhoda takového přístupu spočívá v tom, že uživatel může pořídit spoustu nekvalitních fotek z důvodu rozostření nebo nedostatku obrazové informace pro uživatele, jinými slovy může být pořízena sekvence snímků stejné scény beze změny v popředí.

1.2 Představení řešení

Cílem práce je navrhnout a implementovat nové metody automatické analýzy digitalizovaného obrazu scény z časové sekvence pomocí určování klíčových bodů ve snímku za pomoci stávajících metod.

V práci budou rozebírány celkem tři různé přístupy k využití sekvence snímků, které následně budou implementovány do samostatné mobilní aplikace v prostředí Android, která bude jednak zaznamenávat sekvence obrazů a následně je zpracuje a uloží ve formátu JPG. V diplomové práci budou představeny k přestaveny jednotlivé módy, které budou využívat klíčové body ve snímku. Veškeré výpočty v rámci vývoje jednotlivých přístupů byly prováděny na zařízení Apple MacBook Air s M1 čipem, všechny výsledné hodnoty porovnávacích technik jsou proto brány pouze jako relativní hodnoty na porovnání mezi sebou.

- Odstranění pohybu V prvním přístupu bude moci uživatel pomocí pořízené sekvence fo-

tografií odstranit pohyblivé objekty ze scény. V části diplomové práce se porovnají stávající přístupy k detekci pohyblivých objektů ve scéně a zároveň dojde k využití nejvíce vhodného řešení na praktických příkladech v reálných situacích, kdy uživatel bude moci pořídít snímek například bez pohybučících se aut či osob.

- Super kompozice V druhém přístupu budou klíčové body snímku využity pro kompozici sekvence obrázků, kdy uživateli vznikne konečný snímek složený pořízené sekvence. Tento mód najde využití v případě, kdy uživatel bude potřebovat zaznamenat scénu, která se nevejde do zorného pole objektivu a místo toho scénu složí z dílčích snímků.
- Filtr obrazové sekvence V tomto přístupu bude mobilní zařízení zaznamenávat sekvenci snímků do té doby, dokud se uživatel rozhodne sekvenci zastavit. Následně dostane na výběr snímku k uložení čerstvě pořízené sekvence snímků. V tomto režimu aplikace využije klíčové body k vyfiltrování rozmazaných snímků.

Samotná aplikace bude rozdělená do dvou částí, kdy každá bude napsaná v jiném jazyce v Javě a Pythonu. Grafické rozhraní aplikace a samotné rozhraní kamery fotoaparátu budou napsaná v Javě využívající vývojové prostředí Android Studio [2]. Samotné zpracování přístupu ke klíčovým bodům ve snímku a jejich využití pro jednotlivé funkce budou jako samostatně spustitelné soubory v Pythonu. Pro spuštění kódu v Pythonu v Android prostředí se využije rozhraní pro programování aplikace Chaquopy [1].

2 Metody a techniky

2.1 Významné body v obrazu

V počítačovém vidění a ve zpracování obrazu se detektory a deskriptory významných bodů staly důležitými algoritmy. Jsou široce používány v mnoha aplikacích založených na vidění, jako je reprezentace obrazu, klasifikace obrazu, rozpoznávání objektů, 3D modelování, sledování objektů nebo biometrické systémy. V této práci budou klíčové body využívány především pro perspektivní transformaci, kde jsou klíčové body použity pro tvorbu transformační matice H , viz kapitola 2.8.

Takové aplikace vyžadují robustní body v obrazu, které jsou nejen reprezentativní, ale také invariantní vůči šumu, změně měřítka nebo změně jasu. Proto je detekce a extrahování významných bodů z obrázků nezbytnou součástí těchto aplikací.

V počítačovém vidění pojem významný bod odkazuje na část informace, která představuje vlastnosti obrazu. Tento koncept je obecně stejný jako funkce ve strojovém učení a rozpoznávání vzorů, i když obrazová data mají velmi sofistikovanou sbírku funkcí. Významné body lze považovat za zajímavé části obrazu. Vzhledem k tomu, že významné body se používají jako výchozí bod a je to základ pro následné algoritmy, bude celkový algoritmus často jen tak dobrý, jak dobrý jsou klíčové body ve snímku se, kterými se pracuje. Obecně lze prvky obrázku kategorizovat jako hrany, rohy.

Detekce významných bodů je metoda pro výpočet extrakce obrazových informací a přijímání místních rozhodnutí v každém obrazovém bodě, zda v daném pixelu existuje obrazová informace daného typu nebo ne. Detekce takovýchto bodů je nízkourovňová operace zpracování obrazu. To znamená, že se obvykle provádí jako první operace na obrázku a zkoumá každý pixel, aby se zjistilo, zda je v tomto pixelu přítomen informativní obrazový prvek. Po detekci klíčových bodů v obrázku je potřeba mít metodu k popisu lokálních vlastností obrázku v těchto bodech, odtud deskriptor obrazového bodu. Tyto algoritmy extrahují zajímavé informace z obrazových dat v detekovaných klíčových bodech. Běžnou praxí pro organizování informací poskytovaných těmito algoritmy pro popis vlastností je zakódování jednotlivých vlastností do jednoho vektoru, který je běžně označován jako příznakový vektor. Sada všech možných příznakových vektorů tvoří příznakový prostor.

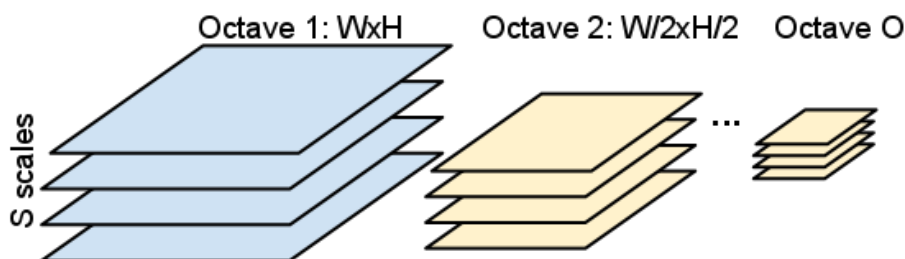
V následujících částech bude poskytnut přehled o známých technikách detekce významných bodů obrazu a jejich popisu, které jsou využívány v této práci.

2.2 SIFT

Metoda SIFT (Scale-Invariant Feature Transform) je algoritmus počítačového vidění pro detekci, popis a shodu lokálních příznaků v obrazech, který vynalezl David Lowe v roce 1999 [?].

2.2.1 Detekce extrému

Detaily v obraze jsou tvořeny strukturami s malým měřítkem, pokud se měřítko postupně zvětšuje, zůstanou v obraze pouze struktury s vyšším měřítkem. Diskrétní škálovaný prostor je trojrozměrnou reprezentací obrazu o několika vrstvách, kde první vrstva odpovídá původnímu obrazu a každá další je tvořena obrazem s vyšším měřítkem. Pohledem frekvenční analýzy jsou detaily v obraze reprezentovány nejvyššími frekvencemi. Spolu se zvyšováním měřítka tedy dochází k filtraci typu dolní propust a v obraze jsou tak zachovány jen frekvence nižší. Takto škálovaný prostor je rozčleněn na oktávy skládající se z vrstev o stejném rozměru, přičemž nejvyšší vrstva má dvojnásobné měřítko oproti první vrstvě. Základem následující oktávy je poslední vrstva oktávy předešlé, jež je podvzorkována na poloviční rozměr, viz obrázek 1). Škálovaný prostor je rovněž také označován jako měřítkově nezávislá forma obrazu [21].



Obrázek č. 1: Diskrétní škálovaný prostor. [17]

Během tohoto kroku metody SIFT jsou nalezeny potencionálně významné body, kandidáti, jenž zůstávají stabilní i během změny měřítka obrazu. Je tedy sestavena funkce, která postupně převádí vstupní obraz do různých měřítek a v takto vzniklých obrazech následně dochází k lokalizaci významných bodů. Jak bylo dříve dokázáno, jedinou možnou funkcí splňující tento účel je funkce Gaussova. Je tedy zkonstruována tzv. Gaussova pyramida skládající se z obrazů $L(x, y, \sigma)$, které vznikají konvolucí Gaussova jádra $G(x, y, \sigma)$ o různém měřítku σ (šířka jádra) se vstupním obrazem $I(x, y)$:

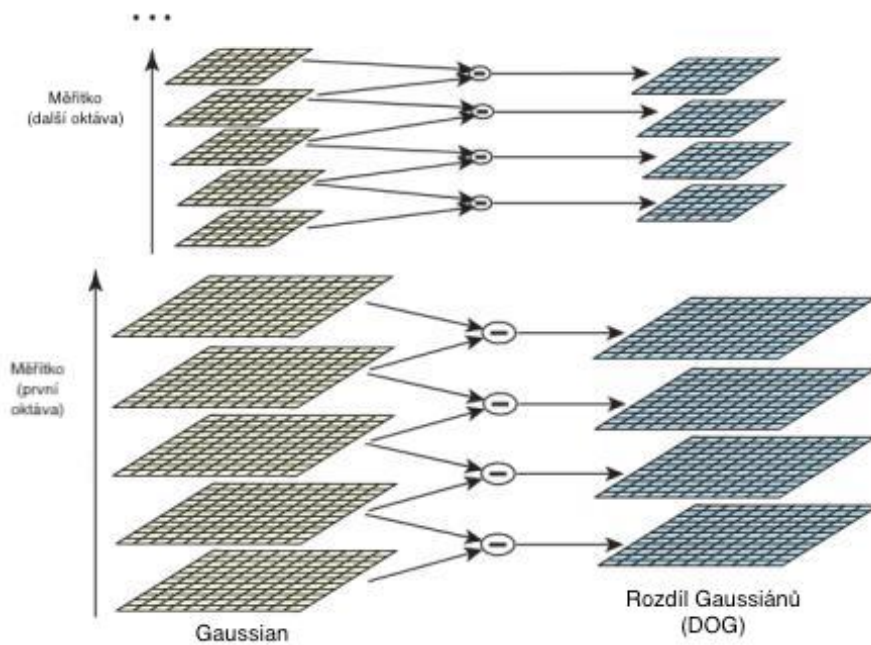
$$L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y), \quad \text{kde} \quad G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (1)$$

Při sestavování Gaussovy pyramidy se využívá vlastnosti soběpodobnosti Gaussovy funkce. To znamená, že vyšší vrstva je získána opakovanou aplikací konvoluce na vrstvu předešlou bez nutnosti neustálého zvětšování šířky konvolučního jádra[21].

Samotný škálovaný prostor, v níž jsou následně hledány extrémy, je sestavena z rozdílových obrazů $D(x, y, \sigma)$, které vznikají rozdílem dvou sousedních obrazů $L(x, y, \sigma)$ v Gaussově pyramidě. Je zřejmé, že sousední obrazy mají rozdílné měřítko, které vyjadřuje koeficient k :

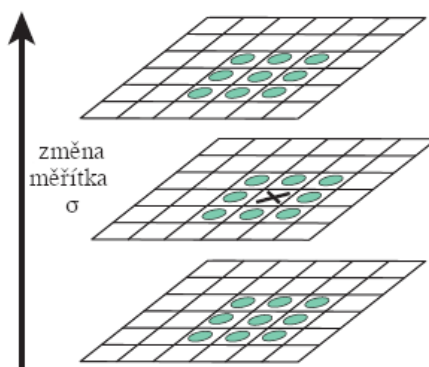
$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (2)$$

Obraz $D(x, y, \sigma)$ se nazývá rozdíl Gaussových funkcí. Konstrukce Gaussovy pyramidy a tedy škálovaného prostoru pomocí rozdílových obrazů je znázorněna následujícím obrázkem 2.



Obrázek č. 2: Konstrukce Gaussovy pyramidy. [22]

Lokální maxima a minima v $D(x, y, \sigma)$ jsou nalezeny prozkoumáním 26-okolí každého bodu obrazu, viz obrázek 3, kde pixel (označený X v obrázku) je porovnáván s pixely ve svém 26-okolí. Pokud takový bod má nejmenší či největší hodnotu ve svém okolí, stává se kandidátem na významný bod objektu [19].



Obrázek č. 3: Detekce lokálního extrému. [18]

2.2.2 Zpřesnění polohy významných bodů

Kandidáti nalezení v předchozím kroku metody jsou charakterizovány svou souřadnicí a měřítkem. Pro tyto body je vhodné přesněji určit jejich polohu a naopak z množiny nalezených kandidátů vypustit takové body, které mají nízký kontrast, jelikož právě tyto body jsou citlivé na šum, nebo se nacházejí v blízkosti hran v obraze. Body ležící v blízkosti hran jsou špatně lokalizovatelné a jako takové, musejí být dodatečně odfiltrovány.

Poloha kandidáta, či extrému, je aproximována kvadratickou funkcí. Za tímto účelem byla vyvinuta metoda, která pracuje na základě Taylorova rozvoje obrazu $D(x, y, \delta)$, jehož počátek je posunut do zkoumaného bodu \mathbf{x} :

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \cdot \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}, \quad (3)$$

kde D je hodnota derivace ve zkoumaném bodě $\mathbf{x} = (x, y, \sigma)^T$. Přesná pozice kandidáta je určena posunutím $\hat{\mathbf{x}}$, které odpovídá hodnotě extrému Taylorova rozvoje. Ten je spočten pomocí derivace funkce $D(\mathbf{x})$ podle \mathbf{x} a následným položením rovno nule

$$\hat{\mathbf{x}} = \frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \cdot \frac{\partial D}{\partial \mathbf{x}}. \quad (4)$$

Posunutí $\hat{\mathbf{x}}$ dosahující větších hodnot než 0,5 v kterékoliv dimenzi, značí skutečnost, že se extrém nachází blíže k sousednímu bodu. V takovém případě je zkoumaný bod nahrazen bodem sousedníma a celý výpočet se opakuje. Pokud $\hat{\mathbf{x}}$ nepřesahuje hodnotu 0,5 je výsledné posunutí přičteno k souřadnicím zkoumaného bodu, čímž je získána sub-měřítková poloha extrému [20].

2.2.2.1 Odstranění bodů v blízkosti hran

Ačkoliv rozdíl Gaussových funkcí $D(x, y, \sigma)$ vykazuje velkou odezvu v blízkosti hran obrazu, není vhodné takové pixely řadit do množiny významných bodů. Tyto body jsou totiž špatně lokalizovatelné a tudíž nestabilní.

Špatně definované extrémy funkce $D(x, y, \sigma)$ vykazují velkou hlavní křivost podél hran, avšak malou křivost v ortogonálním směru. Hlavní křivost v okolí zkoumaného bodu lze vyjádřit pomocí vlastních čísel Hessiány matice:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (5)$$

spočtení vlastních čísel je ovšem výpočetně náročná operace a proto se místo konkrétní hodnoty těchto čísel hledá jejich poměr. Pokud se totiž zkoumaný bod nachází v blízkosti hrany, je jeho jedno vlastní číslo, označené α , větší než to druhé, označen β , a současně platí, že:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta, \quad (6)$$

$$\text{Det}(H) = D_{xx} \cdot D_{yy} - D_{xy}^2 = \alpha \cdot \beta, \quad (7)$$

kde $\text{Tr}(H)$ je stopa Hessianovy matice a $\text{Det}(H)$ jeho determinant. Nyní se zavede r jako jeho poměr mezi vlastními čísly, musí tehdy platit rovnice $\alpha = r \cdot \beta$. Jednoduchou úpravou výše uvedených rovnic a dosazením lze získat tvar:

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r+1)^2}{r}. \quad (8)$$

Extrém funkce $D(x, y, \sigma)$ je zachován v množině významných bodů jen v případě, že je výše uvedená nerovnost splněna. Práh r je vstupním parametrem metody SIFT [20].

2.2.3 Přiřazení orientace významným bodům

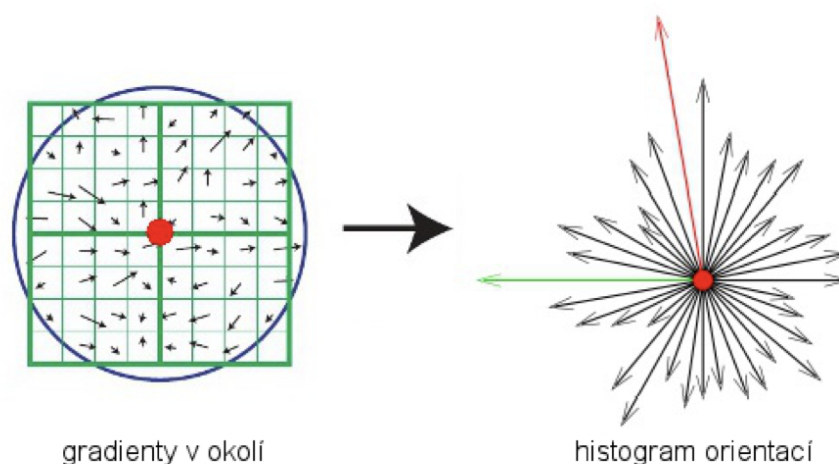
Body, které vzešly z předchozích dvou kroků metody SIFT, jsou nyní považovány za významné body. Jsou to body jednoznačně lokalizované ve škálovaném prostoru a je tedy invariantní vůči změně velikosti obrázku. Aby mohli být významné body považovány za invariantní vůči rotaci, musím jim být přiřazena jejich dominantní orientace.

Proces přiřazení orientace probíhá tak, že nejdříve je ke zkoumanému bodu $\mathbf{x} = (x, y, \sigma)^T$, $\mathbf{x} \in D(x, y, \sigma)$, nalezen měřítkově nejbližší obraz $L(x, y, \sigma)$ nacházející se v již dříve sestrojené Gaussově pyramidě. Tímto přístupem je zajištěna měřítková nezávislost. Pro každý bod obrazu $L(x, y, \sigma)$, daného měřítka σ , je pak spočtena pomocí rozdílů jasových hodnot pixelů velikost gradientu $m(x, y)$ a orientace gradientu $\Theta(x, y)$:

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}, \quad (9)$$

$$\Theta(x, y) = \arctan \left[\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right]. \quad (10)$$

Následně je zkonstruován histogram orientací sestávající se z gradientů spočtených v okolí významného bodu. Histogram orientací má 36 binů pokrývající rozsah rotace o 360° . K binům histogramu s odpovídající orientací jsou přičteny hodnoty vážené velikostmi gradientů a koeficienty Gaussova kruhového okna, viz následující obrázek 4



Obrázek č. 4: Gradienty spočtené v okolí významného bodu.

Jak je možné vidět na obrázku 4, kružnice v levém okně značí vážení gradientů v okolí bodů Gaussovým oknem a sestavený histogram napravo vyobrazuje šípkami více dominantních orientací.

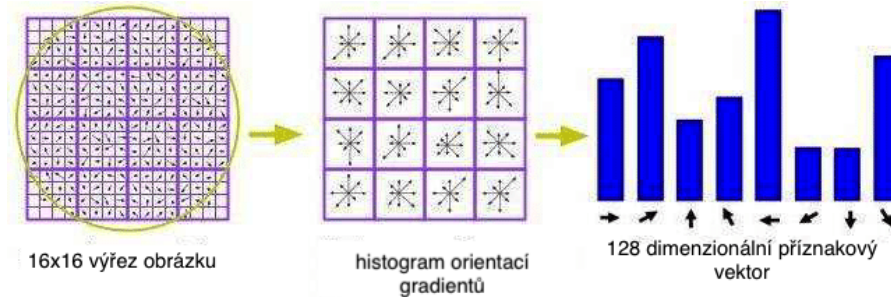
Dominantní orientace významného bodu je poté určena globálním maximem v sestaveném histogramu orientací. Pokud se ovšem v histogramu vyskytují lokální maxima, které dosahují alespoň 80% hodnoty globálního, označuje se takový histogram za histogram s více dominantními orientacemi. V takovém případě je na stejných souřadnicích vytvořen další významný bod s orientací odpovídající další nejvíce zastoupené orientaci v histogramu. Toto platí i v případě, že je v histogramu existuje i více než dvě dominantní orientace [21].

2.2.4 Sestavení deskriptoru

V předchozích krocích byla nalezena množina kandidátů na významné body v obraze. Z této množiny byly postupně odstraněny body vykazující nestabilitu a u zbývajících byla přesně definována jejich poloha. Dále byla jednotlivým významným bodům přiřazena dominantní orientace pomocí konstrukce histogramu orientací v jejich okolí. Nyní do celého procesu vstupuje tzv. deskriptor. Jeho úkolem je popsat okolí jednotlivých významných bodů, tak aby tento popis byl nezávislý na geometrických a jasových transformacích obrazu.

Deskriptor se sestavuje z gradientů v blízkém okolí významného bodu. Je tedy možné pro jeho konstrukci využít výsledky z předešlých kroků metody. Nejprve je významnému bodu vybrán nejbližší obraz $L(x, y)$ z Gaussovy pyramidy, tak aby byla opět zachována nezávislost na měřítku. Okolí zkoumaného bodu je rozděleno na 8 stejně velkých čtvercových oblastí a pro každou oblast je sestaven 8-binový histogram orientací. Tyto histogramy jsou poté natočeny podle určené dominantní orientace významného bodu, čímž je zajištěna nezávislost na rotaci. Rozčlenění okolí do několika oblastí má výhodu v tom, že výsledný deskriptor je odolný vůči malým posunům obrazu. Za účelem zesílení této odolnosti SIFT navíc používá trilineární interpolaci, která výslednou hod-

notu gradientu rozkládá mezi sousední biny v histogramu orientací. Výsledný deskriptor se skládá z jednotlivých histogramů nacházejících se v okolí významného bodu a je označován jako 128-binový deskriptor, kdy okolí je složeno ze 4x4 oblastí, kde každá obsahuje právě 8-binový histogram [21].



Obrázek č. 5: Deskriptor metody SIFT. [23]

Pro nezávislost deskriptoru na transformacích jasové funkce obrazu stačí provést normalizaci vektoru deskriptoru. Při změně kontrastu dojde k přenásobení hodnot jasu v obraze určitou konstantou, dojde tedy k nárůstu velikosti gradientu. Získaný deskriptor metody SIFT je po normalizaci odolný i vůči jasovým transformacím obrazu [21].

2.3 SURF

SURF (Speeded Up Robust Features) je částečně inspirován deskriptorem SIFT. Standardní verze SURF je několikanásobně rychlejší než SIFT a její autoři tvrdí, že je odolnější vůči různým transformacím obrazu než SIFT. SURF poprvé představili Herbert Bay a kol., v roce 2006 na Evropské konferenci o počítačovém vidění [24].

Aplikace tohoto algoritmu je patentována ve Spojených státech v důsledku těchto patentů nebude nikterak využita pro zpracování obrazu v této práci.

Jak bylo v předchozí kapitole 2.2 zmíněno SIFT škálování prostor aproximuje pomocí rozdílů Gaussových funkcí. Ty ovšem mají velké odezvy i v okolí hran obrazu, které je potřeba z množiny významných bodů dodatečně odstranit pomocí Hessianu. Metoda SURF tyto dva kroky slučuje a významné body detekuje přímo pomocí determinantu Hessianovy matice. Rychlý výpočet determinantu je umožněn aplikací integrálního obrazu. Popisuje rozložení Haarových vlnových odezev v okolí významného bodu. Integrální obrazy se používají vzhledem rychlosti a využívá se pouze 64 dimenzí, což zkracuje čas na výpočet prvků a shodu. Krok indexování je založen na laplaciánském znaménku, což zvyšuje rychlost párování a robustnost deskriptoru. Důležité zvýšení rychlosti je způsobeno použitím integrálních obrázků, které významně snižují počet operací pro jednoduché více tří rozměrné konvoluce, nezávisle na zvoleném měřítku. I bez specializovaných optimalizací je možný výpočet téměř v reálném čase bez ztráty výkonu, což představuje důležitou výhodu pro mnoho aplikací počítačového vidění. SURF je rychlejší než SIFT z hlediska doby provedení. Pokud

však rychlost není kritická, SIFT stále překonává SURF v kvalitě detekce [25].

2.4 ORB

ORB (Oriented FAST and Rotated BRIEF) je algoritmus pro nalezení a popsání významných bodů ve snímku. Tento algoritmus byl vyvinut v roce 2011 jako alternativa k tehdejší patentovaným metodám SIFT a SURF, které byly rovněž využívány jako detektory významných bodů ve snímku [13].

ORB algoritmus pro porovnávání obrazu je obecně rozdělen do tří kroků:

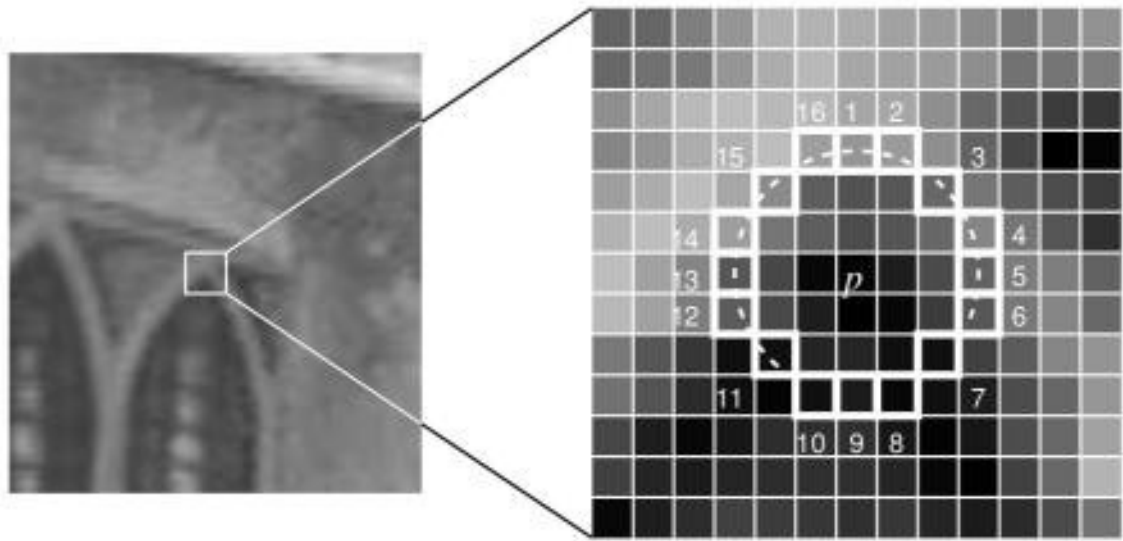
- extrakce charakteristických bodů
- generování deskriptorů charakteristických bodů
- přiřazení charakteristických bodů

2.4.1 Extrakce charakteristických bodů

K extrakci bodů ORB algoritmu slouží FAST (Features from Accelerated and Segments Test) algoritmus k detekci bodů prvků. Myšlenka je taková, že pokud se pixel výrazně liší od sousedních pixelů, je pravděpodobnější, že se bude jednat o rohový bod. Proces detekce je následující:

2.4.1.1 Detekce významných bodů obrazu

Nejprve se vybere pixel p na snímku a předpokládá se, že jeho jas je I_p . Nastaví se práh jasu T . Poté se vezme pixel p jako střed, vybere se 16 okolních pixelů na kruhu o poloměru 3 pixely a porovná se hodnota šedotónového obrazu mezi pixelem p s ostatními pixely v kruhu. Pokud je jas po sobě jdoucích N bodů na vybraném kruhu větší než $I_p + T$ nebo menší než $I_p - T$, lze pixel p považovat za hlavní bod.[9]



Obrázek č. 6: FAST algoritmus výběr okolí. [9]

2.4.1.2 Screening hlavních bodů

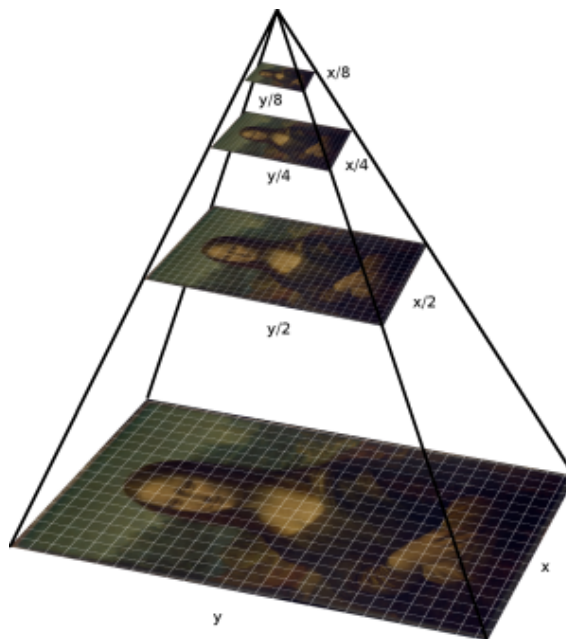
Vzhledem k tomu, že výpočet rohového bodu FAST slouží pouze k porovnání rozdílu v jasu mezi pixely, je toto číslo velké a nejisté a neexistuje žádná informace o směru. Proto algoritmus ORB vylepšuje původní FAST algoritmus, který vypočítává hodnoty Harrisovy matice odezvy pro původní FAST rohové body a třídí je podle šedotónové hodnoty a bere se prvních N bodů. Vzorec pro výpočet hodnoty Harrisovy odezvy je uveden v rovnici 11:

$$R = \det(H) - k(\text{Tr}(H))^2 \quad (11)$$

kde $\text{Tr}(H)$ je stopa Hessiánské matice, viz. (6) a $\det(H)$, viz. (7), jeho determinant a R je hodnota Harrisovy odezvy, H je matice 2×2 , k se pohybuje od 0,04 do 0,06, $w(x, y)$ je funkce okna obrázku, I_x je variace funkčního bodu v horizontálním směru a I_y je variace funkčního bodu ve vertikálním směru [12].

2.4.1.3 Konstrukce pyramid

Problém je, že funkce FAST není invariantní vůči orientaci nebo změně měřítka. ORB algoritmus tedy používá pyramidu s mnoha měřítky. Obrazová pyramida je víceúrovňová reprezentace jednoho obrazu, která se skládá ze sekvencí obrazů, z nichž všechny jsou verzemi obrazu v různých rozlišeních. Každá úroveň v pyramidě obsahuje převzorkovanou verzi obrázku než předchozí úroveň. Jakmile ORB vytvoří pyramidu, použije rychlý algoritmus k detekci významných bodů v obraze. Tímto způsobem je ORB invariantní v částečném rozsahu měřítka [9].



Obrázek č. 7: Pyramida FAST algoritmu. [10]

2.4.1.4 Určení směru významného bodu

Aby získané významné body byly invariantní vůči otočení, směr prvků se získá pomocí metody Intensity Centroid. Nejdříve, v malém obrazovém bloku B je moment obrazového bloku definován jako:

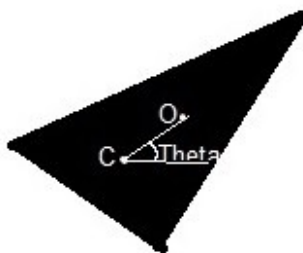
$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x, y), \quad p, q = 0, 1, \quad (12)$$

kde x a y jsou souřadnice pixelu a $I(x, y)$ je hodnota pixelu v šedotónovém obraze. Tím je možné najít centroid daného bloku podle momentu:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right), \quad (13)$$

kde 0. moment (m_{00}) je váha obrazového bloku a 1. moment (m_{10}, m_{01}) je centroid obrazového bloku. Nakonec je geometrický střed O a centroid C obrazového bloku připojen, aby se získal směrový vektor \vec{OC} a směr významného bodu je definován jako:

$$\Theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \quad (14)$$



Obrázek č. 8: Ilustrace orientace thety [11]

Prostřednictvím výše uvedených kroků jsou FAST významné body invariantní vůči změně měřítka a rotace, což výrazně zlepšuje jejich robustnost.

2.4.2 Sestavení deskriptoru

BRIEF (Binary robust independent elementary feature) vezme všechny významné body nalezené FAST algoritmem a převede je na binární příznakový vektor, takže společně mohou reprezentovat objekt. Binární příznakový vektor, známý také jako deskriptor, je vektor příznaků, který obsahuje pouze 1 a 0, každý významný bod je tedy popsán vektorem prvku, který je 128–512 bitovým řetězcem [15].

BRIEF začíná vyhlazováním obrazu pomocí Gaussova jádra, aby se zabránilo citlivosti deskriptoru na vysokofrekvenční šum. Poté krátce vybere náhodný pár pixelů v definovaném sousedství kolem tohoto významného bodu. Definované okolí kolem pixelu je známé jako patch, což je čtverec o dané šířce a výšce pixelu. První pixel v náhodném páru je vybrán ze středu gaussovské distribuce kolem významného bodu s odchylkou sigma. Druhý pixel v náhodném páru je vybrán z Gaussova rozložení se středem kolem prvního pixelu se standardní s odchylkou sigma půl. Pokud je nyní první pixel jasnější než druhý, přiřadí hodnotu 1 odpovídajícímu bitu, jinak 0[15].

BRIEF opět vyberte náhodný pár a přiřadí jim hodnotu. U 128bitového vektoru BRIEF opakujte tento proces 128krát pro konkrétní významný bod. Algoritmus zopakuje tento postup pro každý významný bod v obrázku. BRIEF ovšem také není invariantní vůči rotaci, takže ORB využívá rBRIEF (BRIEF aware rotation) [14].

2.5 BRISK

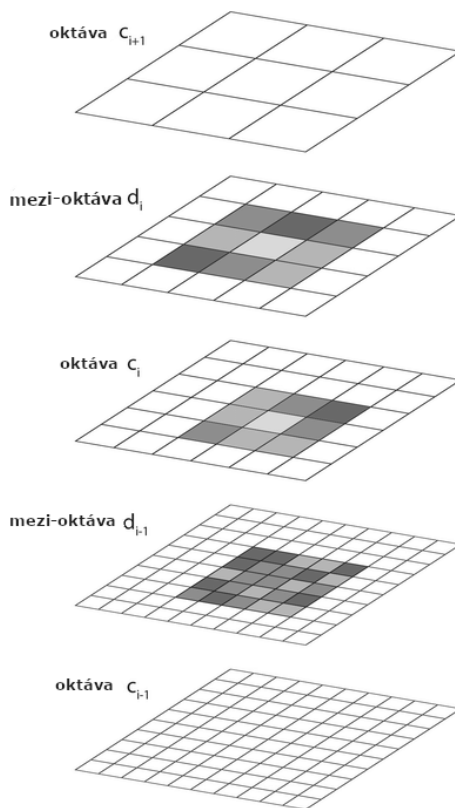
Metoda BRISK (Binary Robust Invariant Scalable Keypoints) je jednou z dalších nových metod pro vysoce kvalitní, rychlou detekci, popis a párování klíčových bodů. Jak již název napovídá, metoda je do značné míry invariantní rotací i měřítkem, čímž dosahuje výkonu srovnatelného se současným stavem techniky a zároveň dramaticky snižuje výpočetní náklady. [29]

2.5.1 Detekce významných bodů

Se zaměřením na efektivitu výpočtů je metodologie detekce inspirována metodou AGAST (Adaptive and Generic Accelerated Segment Test), ta je v podstatě rozšířením pro zrychlený výkon již zmíněného FAST v předchozí kapitole 2.4.1, který se ukázal jako velmi účinný základ pro extrakci klíčových bodů. S cílem dosáhnout invariance vůči měřítku, která je klíčová pro nalezení vysoce kvalitních klíčových bodů ve snímku, v případě BRISK jde o krok dále tím, že se hledá maximum nejen v obrazové rovině, ale také v ale rovněž napříč měřítkovou pyramidou pomocí FAST skóre s jako měřítko pro význačnost. Navzdory diskretizaci osy měřítka v hrubších intervalech než u alternativních výkonných detektorů, detektor BRISK odhaduje skutečné měřítko každého klíčového bodu ve spojitém měřítku.

V rámci BRISK se škálovací pyramidy skládají z n oktáv c_i a n mezi-oktáv (z ang. *intra – octave*) d_i , pro $i = 0, 1, \dots, n - 1$ a typicky $n = 4$. Oktávy jsou vytvořené progresivním polovičním vzorkováním původního obrazu (odpovídající c_0), kdy Každá mezi-oktáva d_i je umístěna mezi vrstvami c_i a c_{i+1} , jak je znázorněno na následujícím obrázku 9.

První mezi-oktáva d_0 se získá převzorkováním původního obrazu c_0 faktorem 1,5, zatímco zbytek mezi-oktávových vrstev je odvozen postupným polovičním vzorkováním. Pokud tedy t označuje měřítko, pak $t(c_i) = 2^i$ a $t(d_i) = 2^i \cdot 1,5$.



Obrázek č. 9: Detekce významného bodu v pyramidovém prostoru [27]

Zde je důležité poznamenat, že FAST poskytují různé alternativy tvarů masky pro detekci

klíčových bodů. V BRISK se většinou používá maska 9-16 pixelů, která v podstatě vyžaduje alespoň 9 po sobě jdoucích pixelů v 16 pixelovém kruhu, aby byly dostatečně jasově výraznější nebo tmavší než centrální pixel, aby bylo splněno kritérium FAST [26].

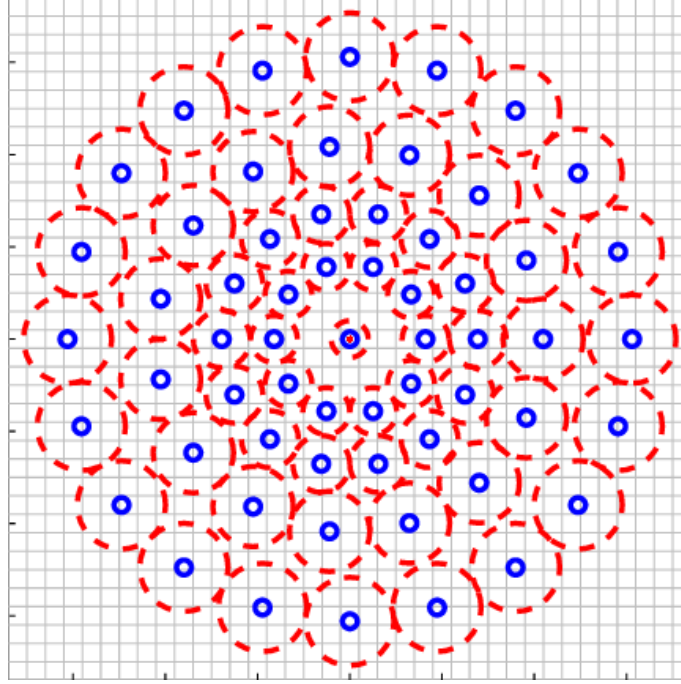
Zpočátku je detektor FAST 9-16 aplikován na každou oktávu a mezi-oktávu samostatně s použitím stejného prahu T k identifikaci potenciálních oblastí zájmu. Dále jsou body patřící do těchto regionů podrobeny nemaximálnímu potlačení v škálovaném prostoru: za prvé, dotyčný bod musí splňovat maximální podmínku s ohledem na jeho 8 sousedních skóre s získaným FAST algoritmem ve stejné vrstvě. Skóre s je definováno jako maximální práh rozhodující zda je významný bod považován za roh nebo ne. Za druhé, skóre ve vrstvě nad a pod bude muset být také nižší. Zkontroluje se vnitřní stejně velké čtvercové záplaty, kdy délka strany je zvolena tak, aby její velikost byla 2 pixely ve vrstvě s předpokládaným maximem. Vzhledem k tomu, že sousední vrstvy (a tedy jejich FAST skóre) jsou reprezentovány odlišnou diskretizací, je na hranicích pole aplikována určitá interpolace [26].

Detekce maxim napříč osou škálovaného prostoru v oktávě c_0 je speciální případ, kdy se získá skóre FAST pro virtuální mezi-oktávu d_1 pod c_0 , aplikuje se maska FAST 5 – 8px na c_0 . Avšak skóre v políčku d_1 v tomto případě nemusí být nižší než skóre zkoumaného bodu v oktávě c_0 [26].

Vezme-li se v úvahu nápadnost obrazu jako spojitou veličinu nejen napříč obrazem, ale také podél osy škálovaného prostoru, provádí se zjemnění subpixelů a spojitých měřítek pro každé detekované maximum. Aby se omezila složitost procesu upřesňování, nejprve se přizpůsobí 2D kvadratická funkce ve smyslu nejmenších čtverců každé ze tří záplat skóre (tak jak byla získána ve vrstvě klíčového bodu, jednu výše a jednu níže) výsledkem jsou tři subpixelové zjemněné maxima význačnosti. Aby se vyhnulo převzorkování, uvažuje se o záplatě 3 x 3 na každé vrstvě. Dále se takovéto skóre použijí k přizpůsobení 1D paraboly podél osy měřítka, čímž se získá konečný odhad skóre a odhad měřítka na svém maximu. Jako poslední krok se znovu interpolují souřadnice obrazu mezi záplatami ve vrstvách vedle určeného měřítka [26].

2.5.2 Deskriptor klíčových bodů

Vzhledem k sadě klíčových bodů (skládajících se ze subpixelových upřesněných umístění obrazu a souvisejících hodnot stupnice s plovoucí desetinnou čárkou) je deskriptor BRISK složen jako binární řetězec zřetěžením výsledků jednoduchých testů porovnání jasu. V BRISK se identifikuje charakteristický směr každého klíčového bodu, aby byla umožněna orientace normalizovaných deskriptorů, a tím bylo dosaženo rotační invariance, která je klíčem k obecné robustnosti. Rovněž je také potřeba správně vybrat jasy ke srovnání se zaměřením na maximální informativnost [26].



Obrázek č. 10: Vzorkovací maska [28]

2.5.2.1 Vzorkovací maska a odhad natočení

Klíčový koncept BRISK deskriptoru využívá vzorkovací masku pro navzorkování okolí klíčového bodu. Vzor znázorněný na obrázku 10 definuje N umístění rovnoměrně rozmístěných v kruzích soustředných s klíčovým bodem. Na obrázku je maska pro vzorkování s $N = 60$ body, kdy malé modré kroužky označují místa vzorkování, větší červené čárkované kruhy jsou nakresleny v poloměru δ odpovídajícím standardní odchylce Gaussova jádra použitého k vyhlazení hodnot intenzity v bodech vzorkování. Zobrazený vzor platí pro stupnici $t = 1$ [26].

V takovém to případě je kladen důraz na vyhnutí se aliasingovým efektům při vzorkování jasové intenzity obrazu bodu p_i v masce, zde se aplikuje Gaussovo vyhlazování se směrodatnou odchylkou σ_i úměrnou vzdálenosti mezi body na příslušné kružnici. Umístěním a změnou měřítka vzoru pro konkrétní klíčový bod k v obrázku se uvažuje jeden z $N \cdot (N - 1) / 2$ párů vzorkovacích bodů (p_i, p_j) . Vyhlazené jasové hodnoty v bodech $I(p_i, \sigma_i)$ a $I(p_j, \sigma_j)$, tyto hodnoty slouží k odhadu lokálního gradientu $g(p_i, p_j)$ podle:

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2}. \quad (15)$$

Dále se vytvoří set \mathcal{A} složený z navzorkovaných párů:

$$\mathcal{A} = (p_i, p_j) \in \mathbb{R}^2 \cdot \mathbb{R}^2 | i < N \wedge j < i \wedge i, j \in \mathbb{N}. \quad (16)$$

Definuje se podmnožina párů na krátké vzdálenosti \mathcal{S} a další podmnožina párů \mathcal{L} na dlouhé

vzdálenosti \mathcal{L} :

$$\mathcal{S} = (p_i, p_j) \in \mathcal{A} \quad \|p_j - p_i\| < \delta_{max} \subseteq \mathcal{A} \quad (17)$$

$$\mathcal{L} = (p_i, p_j) \in \mathcal{A} \quad \|p_j - p_i\| > \delta_{min} \subseteq \mathcal{A} \quad (18)$$

Vyhlazené hodnoty prahových bodů se nastaví na $\delta_{max} = 9,75t$ a $\delta_{min} = 13,67t$, kde t je stupnice k . Při iteraci přes dvojice bodů v \mathcal{L} se odhadne celkový směr charakteristického vzoru klíčového bodu k jako:

$$g = [g_{x,y}]^T = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in \mathcal{L}} g(p_i, p_j). \quad (19)$$

Pro tento výpočet jsou použity vzdálené páry na základě předpokladu, že lokální gradienty se navzájem eliminují a nejsou tak nutné při určování globálního gradientu – to bylo potvrzeno i experimentem s variací prahu vzdálenosti δ_{min} [26].

2.5.2.2 Sestavení deskriptoru

Pro vytvoření deskriptoru invariantního na rotaci a změnu měřítka se využije vzorkovací maska otočená o $\alpha = \arctan 2(g_y, g_x)$ kolem klíčového bodu k . Deskriptor bitového vektoru d_k je sestaven využitím všech porovnání intenzity blízkých párů bodů $(p_i^\alpha, p_j^\alpha) \in \mathcal{S}$, takže každý bit b tedy odpovídá:

$$b = \begin{cases} 1, & (P_j^\alpha, \delta_j) > I(p_i^\alpha), \\ 0 & \forall (p_i^\alpha, p_j^\alpha) \in \mathcal{S} \end{cases} \quad (20)$$

Zatímco deskriptor BRIEF, popsáný v předešlé kapitole 2.4.2, je sestaven pomocí porovnávání jasu, BRISK má některé zásadní rozdíly kromě zřejmého předškálování a předtočení vzorkovací masky. Za prvé, BRISK používá deterministickou vzorkovací masku, jejíž výsledkem je jednotná hustota vzorkovacích bodů na daném poloměru kolem významného bodu. V důsledku toho přizpůsobené gaussovské vyhlazování náhodně nezkrusí informační obsah jasu rozmazáním dvou blízkých vzorkovacích bodů. BRISK navíc používá výrazně méně vzorkovacích bodů než párová srovnání, jeden bod se tedy účastní více srovnání, což omezuje složitost vyhledávání hodnot intenzity. A jako poslední, srovnání jsou zde prostorově omezena, takže změny jasu musí být pouze lokálně konzistentní. Se vzorkovací maskou a prahovými hodnotami vzdálenosti, jak je ukázáno výše, je získán bitový řetězec délky 512. Verze bitového řetězce BRIEF64 bitů také obsahuje 512 bitů, takže párování pro pár deskriptorů bude podle definice provedeno stejně rychle [26].

2.6 AKAZE

Dosavadně představené přístupy k detekci a popisu víceškálových prvků jako jsou SIFT a SURF, přičemž SURF je méně výpočetně náročný než SIFT. Takovéto přístupy využívají gaussovsky škálovaného prostoru, buď konstrukcí gaussovského škálovaného prostoru v pyramidě, jako je SIFT,

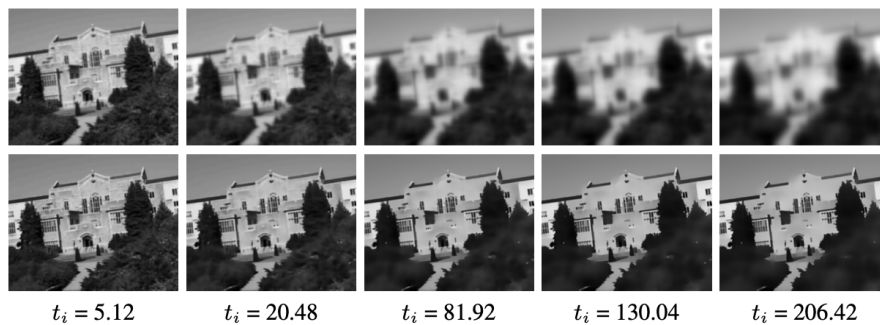
nebo aproximací gaussovských derivací přes krabicové filtry jako v SURF. To má značné nevýhody, protože gaussovské rozostření nezachovává hranice objektů a ve stejném rozsahu vyhlazuje detaily a šum ve všech změnách měřítka, což snižuje přesnost lokalizace a rozlišovací schopnost.

V roce 2012 byla předložena alternativa k patentovaným SIFT a SURF pod názvem KAZE, která byla založena na nelineárním škálovacím měřítku pomocí nelineární difúzní (z ang. Non-linear diffusion) filtrace. Díky tomu se rozostření v obrazech lokálně přizpůsobilo hlavním bodům, čímž se snížil šum a současně se zachovaly hranice oblastí na snímcích předmětu. Detektor KAZE je založen na měřítku normalizovaném determinantu Hessiianovy matice, který je počítán na více úrovních měřítka. Maxima odezvy detektoru se zachycují jako příznakové body pomocí pohyblivého okénka. Nalezením dominantní orientace v kruhovém okolí kolem každého detekovaného bodu je dosažena invariance na rotaci. KAZE je tedy invariantní vůči rotaci, změně velikosti obrázku a mají větší rozlišovací schopnost v různých měřítkách s náklady na prodloužení doby výpočtu. Následná rovnice ukazuje standardní vzorec nelineární difúze: [29]

$$\frac{\partial L}{\partial t} = \text{div}(c(\mathbf{x}, \mathbf{y}, t) \cdot \nabla L), \quad (21)$$

Kde c je funkce vodivosti, div je divergence, ∇ je gradientový operátor a L je jas obrazu.

Na následujícím obrázku ?? je porovnání mezi gaussovským a nelineárním difúzním měřítkovým prostorem pro několik evolučních časů t_i . První řada: Gaussův škálovaný prostor a tedy lineární difúze. Škálový prostor je tvořen konvolucí původního obrázku s Gaussovým jádrem s rostoucí směrodatnou odchylkou. Druhá řada: Nelineární difúzní škálovaný prostor.



Obrázek č. 11: Porovnání mezi gaussovským a nelineárním difúzním měřítkovým prostorem[32]

Algoritmus AKAZE (Accelerated-KAZE) vycházející z KAZE je založen na nelineárním difúzním filtrování jako KAZE, ale jeho nelineární škálovaný prostor je zkonstruován pomocí výpočetně efektivního algoritmu zvaného rychlá explicitní difúze FED (Fast Explicit Diffusion).

2.6.1 Rychlá explicitní difúze

FED kombinuje výhody explicitních a poloimplicitních funkcí a zároveň se vyhýbá jejich nedostatkům. Filtrace FED je motivována dekompozicí krabicových filtrů ve smyslu explicitních funkcí. Iterované krabicové filtry se s dobrou kvalitou přibližují gaussovským jádrům a jsou snadno implementovatelné. Hlavní myšlenkou je provést M cyklů n explicitních difúzních kroků s různými velikostmi kroku τ_j , které pocházejí z faktorizace krabicového filtru:

$$\tau_j = \frac{\tau_{max}}{2 \cos^2\left(\pi \cdot \frac{2j+1}{4n+2}\right)}, \quad (22)$$

kde τ_{max} je maximální velikost kroku, která neporušuje podmínku stability explicitního systému. Odpovídající doba zastavení Θ_n jednoho cyklu FED se získá jako:

$$\Theta_n = \sum_{j=0}^{n-1} \tau_j = \tau_{max} \cdot \frac{n^2 + n}{3}. \quad (23)$$

Některé kroky velikostí τ_j z rovnice 23 mohou porušovat podmínky stability.

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, t) \cdot \nabla L) \quad (24)$$

Diskretizace nelineární difúze, na které je založena metoda KAZE, ?? pomocí explicitního schématu může být vyjádřen ve vektorové maticové notaci jako:

$$\frac{L^{i+1} - L^i}{\tau} = A(L^i)L^i, \quad (25)$$

kde $A(L^i)$ je matice, která kóduje vodivosti pro obraz a τ je konstantní velikost časového kroku taková, že $\tau < \tau_{max}$ dodržuje podmínky stability. V explicitním schématu je řešení L^{i+1} vypočítáno přímo z řešení na předchozí evoluční úrovni L^i a vodivosti obrazu $A(L^i)$:

$$L^{i+1} = (I + \tau A(L^i))L^i, \quad (26)$$

kde I je matice identity. S ohledem na apriorní odhad $L^{i+1,0} = L^i$, a FED cyklus s n proměnným krokem o velikosti τ_j je získán tak:

$$L^{i+1,j+1} = (I + \tau_j A(L^i))L^{i+1,j}, \quad \text{kde } j = 0, \dots, n-1. \quad (27)$$

Zde je důležité poznamenat, že nelinearity z matice $A(L^i)$ jsou udržovány konstantní během celého cyklu FED. Jakmile je cyklus FED hotový, vypočítáme nové hodnoty matice $A(L^i)$.

2.6.2 Detekce významných bodů

V této části bude popsána metoda detekce a popisu nových významných bodů. Využívá se při tom FED pro budování prostoru nelineárního měřítka uvažujícího anizotropní difúzi. Aby byla urychlena konstrukce prostoru nelineárního měřítka, vloží se schéma FED do pyramidálního rámce. Pyramidální strategie a FED schémata umožňují rychlou konstrukci prostoru v nelineárním měřítku, vhodnou pro robustní detekci a popis prvků.

Pro nalezení klíčových bodů se vypočítá determinant Hessianu pro každý z filtrovaných obrázků L_i v nelineárním měřítku. Sada diferenciálních víceškálových operátorů je normalizována s ohledem na změnu měřítka, které bere v úvahu oktávu každého konkrétního obrazu v prostoru nelineárního měřítka, znamená: $\sigma_{i,norm} = \sigma_i/2^{\sigma^i}$,

$$L_{Hessian}^i = \sigma_{i,norm}^2 (L_{xx}^i L_{yy}^i - L_{xy}^i L_{xy}^i). \quad (28)$$

Pro výpočet derivací druhého řádu se využívají zřetězené Scharrovy filtry s velikostí kroku $\sigma_{i,norm}$. Scharrovy filtry aproximují rotační invarianci lépe než jiné filtry nebo centrální diferenciace rozdílů. Nejprve se vyhledají maxima odezvy detektoru v prostorovém umístění. Na každé evoluční úrovni i se kontroluje, že odezva detektoru je vyšší než předem definovaný práh a je to maximum v okně 3×3 pixelů. To se provádí za účelem rychlého odstranění nemaximálních odezev. Poté pro každé z potenciálních maxim se zkontroluje, že odezva je maximem vzhledem k ostatním klíčovým bodům z úrovně $i + 1$ a $i - 1$, respektive přímo nad a přímo pod v okně o velikosti $\sigma_i x \sigma_i$ pixelů. Nakonec je 2D poloha klíčového bodu odhadnuta se subpixelovou přesností přizpůsobením 2D kvadratické funkce determinantu Hessovy odezvy v okolí 3×3 pixelů a nalezením jejího maxima.[34]

2.6.3 Deskriptor klíčových bodů

Binární deskriptor, jako jsou ty používané v BRIEF, ORB a BRISK, se aplikuje i případě AKAZE, protože je lze velmi efektivně vypočítat a spárovat. Navzdory tomu byl teprve nedávno získán náhled na to, jak tyto binární prvky skutečně fungují. Ziegler a kol. [21] ukázali, že namísto odhadu gradientů obrázku, jak bylo navrženo dříve, jsou tyto rysy hašováním citlivým na lokalitu (LSH) Kendallovou tau metriky. Zavedli také deskriptor LUCID, řazení pixelů záplaty získané pomocí stabilního třídícího algoritmu. Výpočet Kendallové vzdálenosti τ je však operace $O(n \log n)$, takže lepším řešením se naskýtá použít Hammingovu vzdálenost. Výsledky LUCID jsou o něco horší než výsledky BRIEF, což naznačuje, že snížení rozměrů provedené hašováním zlepšuje výkon.



Obrázek č. 12: Porovnání LDB vs M-LDB [34]

Modified-Local Difference Binary (M-LDB), který využívá informace o gradientu a intenzitě z prostoru nelineárních měřítek. Deskriptor LDB byl představen je představen v [33] a řídí se stejným principem jako BRIEF, ale používá binární testy mezi průměrem oblastí namísto jednotlivých pixelů pro dodatečnou robustnost. Kromě hodnot intenzity se používá průměr horizontálních a vertikálních derivací v porovnávaných oblastech, což vede k 3 bitům na srovnání. LDB navrhuje použití různých mřížek jemnějších kroků, rozdělení záplaty na mřížky 2 x 2, 3 x 3, 4 x 4 atd., jak je znázorněno na obr. 12a. Průměry těchto pododdílů lze velmi rychle vypočítat pomocí integrálních obrázků. Při zvažování rotace klíčových bodů však nelze použít integrální obrazy a návštěva všech bodů v otočeném pododdílu může být relativně nákladná na výpočetní čas. Invariance rotace se získá odhadem hlavní orientace klíčového bodu jako v KAZE a podle toho se natočí mřížka LDB. Namísto použití průměru všech pixelů uvnitř každé podčásti mřížky se podvzorkují mřížky v krocích, které jsou funkcí měřítka σ prvku. Tato aproximace průměru funguje dobře v našich experimentech. Vzorkování závislé na měřítku zase činí deskriptor odolný vůči změnám měřítka. Tento proces je znázorněn na obr.12b. M-LDB používá deriváty vypočítané v kroku detekce příznaků, čímž snižuje počet operací potřebných k vytvoření deskriptoru.

Vzhledem k tomu, že M-LDB počítá aproximaci průměru stejných oblastí v obrázcích intenzity a gradientu, nejsou booleovské hodnoty, které jsou výsledkem srovnání, na sobě nezávislé. Očekává se, že snížení velikosti deskriptoru výběrem náhodné podmnožiny bitů nebo propracovanější metodou, zlepší výsledky nebo přinejmenším sníží výpočetní zátěž bez klesající výkon. [34]

2.7 Porovnání technik

V této části dojde k náležitému porovnání technik pro nalezení klíčových bodů ve snímku, které byly představeny v předchozích částech 2.2 až 2.6, budou zde porovnány tři rozdílné snímky, zároveň pro všechny snímky platí, že jejich velikost je 4032 x 3024 px.



(a) Park



(b) Billboard



(c) Katedrála

Obrázek č. 13: Příklad obrázků na porovnání technik pro hledání klíčových snímků

V následující tabulce 1 je uveden počet klíčových bodů jednotlivě pro každý snímek a každou metodu, které byly nalezeny. V tabulce je možno vidět, že pro přírodní scénérii 13a je nejméně vhodná metoda AKAZE, kde ve snímku našla pouhých 7012 bodů, zatímco jako nejlepší metoda se jeví ORB s 84864, což je 12x lepší výsledek.

Avšak metoda AKAZE z hlediska počtu nalezených klíčových bodů i v případě billboardu 13b a stavby 13c si vedla nejhůře, zatímco ve všech případech, co se týče počtu nalezených klíčových bodů triumfuje ORB.

	SIFT	ORB	BRISK	AKAZE
park	38737	84864	34364	7012
plakat	23317	79238	24490	16326
katedrála	88773	193795	92745	41508

Tabulka č. 1: Počet nalezených klíčových bodů v jednotlivých obrázcích

V následující tabulce 2 je k vidění porovnání časů potřebných k nalezení konkrétního počtu klíčových bodů. V tomto případě není důležitý samotný čas, jelikož ten je závislý na výkonu výpočetního zařízení, důležitým faktorem v tomto bodě je rozdíl daných časů. Tento rozdíl indikuje, jak jsou metody různě rychlé oproti ostatním.

Nejvíce výpočetního času si zabrala metoda SIFT, která je porovnáním s nejrychleji proběhlou metodou ORB až desetkrát pomalejší. Rovněž je vidět, že i AKAZE s porovnáním s metodou ORB je minimálně pětkrát pomalejší.

	SIFT	ORB	BRISK	AKAZE
park	2.804 s	0.284 s	0.435 s	1.476 s
plakat	2.701 s	0.304 s	0.373 s	1.499 s
katedrála	3.198 s	0.457 s	1.100 s	1.341 s

Tabulka č. 2: Čas uběhlý při získávání klíčových bodů

Poslední tabulka 3 indikuje rychlost nalezení a popsání jednoho významného bodu. Je tedy nejlepší ukázkou rychlosti jednotlivých metod.

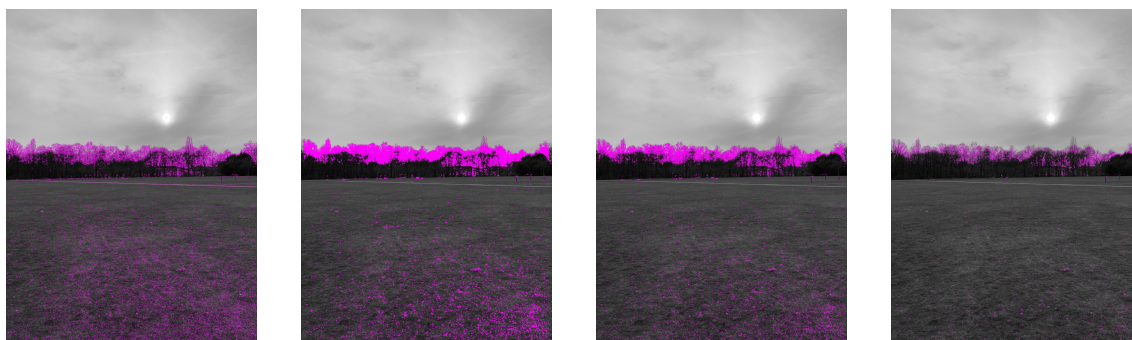
Ukazuje se, že metoda SIFT je obecně z vybraných metod tou nejpomalejší, avšak její průměrná rychlost nalezení jednoho klíčového bodu se snižuje vzrůstajícím počtem nalezených bodů. V případě hledání velkého množství klíčových bodů, tedy 100 000 a více, metoda SIFT se co do rychlosti vyrovná AKAZE.

	SIFT	ORB	BRISK	AKAZE
park	0.0723 ms	0.00334 ms	0.0127 ms	0.210 ms
plakat	0.116 ms	0.00384 ms	0.0152 ms	0.0918 ms
katedrála	0.0360 ms	0.00236 ms	0.0118 ms	0.0323 ms

Tabulka č. 3: Průměrný čas potřebný na nalezení klíčového bodu

Na následujících sériích obrázků 14,15,16 je možno porovnat jaká metoda našla klíčové body a kde ve snímku jich našla nejvíce. Na šedotónových obrázcích jsou klíčové body zobrazeny jako fialové body.

Na obrázcích s parkem si nejlépe vedla metoda SIFT, která se kromě stromů v pozadí dokázala zdatně poradit i s trávnikem v popředí, což se metodám AKAZE a BRISK nepovedlo a je tedy možné usoudit, že tyto metody nejsou úplně vhodné pro detekci bodů v takovémto typu obrazů.



(a) Park SIFT

(b) Park ORB

(c) Park BRISK

(d) Park AKAZE

Obrázek č. 14: Nalezené klíčové body ve snímku parku pro jednotlivé metody

Na příkladu s billboardem je možné vidět, jak si jednotlivé techniky dokážou poradit s jednoduchým textem. V textu je již od pohledu vidět, že nejvíce klíčových bodů našla metoda AKAZE, která

si s textem poradila lépe než zbylé metody. Nejhůře si vedla metoda SIFT o které je tedy možné prohlásit, že detekci klíčových bodů v textu není příliš vhodná. Velmi dobře v porovnání k metodě SIFT si vedl metoda ORB, která obecně detekuje dostatek více klíčových bodů ve snímku než ostatní metody.



(a) Billboard SIFT

(b) Billboard ORB

(c) Billboard BRISK

(d) Billboard AKAZE

Obrázek č. 15: Nalezené klíčové body ve snímku s billboardem pro jednotlivé metody

Na posledních obrázcích 16 byly metody vystaveny snímku se sakrální stavbou, kde nejvhodnější metodou se jeví metoda SIFT, která si našla klíčové body jak v textuře chodníku, tak i v kamenné struktuře samotné katedrály. ORB a BRISK se hodně vážaly na různé hrany a římsy stavby a ne tak dobře samotnou texturu stavby.



(a) Katedrála SIFT

(b) Katedrála ORB

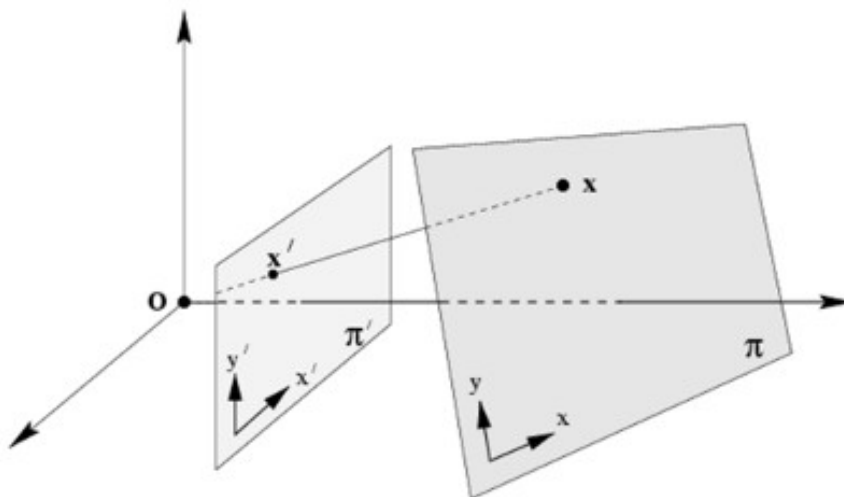
(c) Katedrála BRISK

(d) Katedrála AKAZE

Obrázek č. 16: Nalezené klíčové body ve snímku katedrály pro jednotlivé metody

2.8 Homografie

Homografie, nebo také projektivita, je transformace, která mapuje body z jedné roviny na body v jiné rovině. Je reprezentována transformační maticí 3×3 v homogenním souřadnicovém prostoru.



Obrázek č. 17: Vizualizace homografie [8].

Matematicky je homografie matice reprezentována jako:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (29)$$

Projektivní transformace:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = H \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (30)$$

Jedná se o lineární transformace homogenního vektoru nesingulární 3×3 maticí H , kdy matice H má 8 stupňů volnosti. Matice H je tedy jednoznačně určena čtveřicí sobě korespondujících bodů nebo přímek v obecné pozici. [7]

Kolineární body jsou opět transformovány na kolineární body, z toho vyplývá, že i několik různoběžných přímek se společným průsečíkem jsou transformovány opět na různoběžné přímky s

jedním společným průsečíkem. Současně je i pořadí jednotlivých kolineárních bodů zachováno. Pro bod x platí, že je transformován na bod x' tak, že:

$$x' = Hx. \quad (31)$$

2D homografie je dána množinou bodů x_i v prostoru \mathbb{P}^2 a množinou korespondujících bodů ve stejném \mathbb{P}^2 . Nalezením takovéto transformace z $x_i \iff x'_i$ znamená určit matici H tak, že platí

$$Hx = x'_i, \forall i. \quad (32)$$

Jelikož má 1 bod dva stupně volnosti, tedy souřadnice (x,y) a hledaná matice H má 8 stupňů volnosti, jsou tedy potřeba 4 body a 4 korespondenti. Pro jeden korespondující pár (vztah (32)) se určí soustava 3 lineárních rovnic. Následně se může soustava přepsat jako vektorový součin $x_i \times Hx_i = 0$ a pak se získá:

$$x'_i \times Hx_i = \begin{bmatrix} y'h^{3T}x_i - w'h^{2T}x_i \\ w'h^{1T}x_i - x'h^{3T}x_i \\ x'h^{2T}x_i - y'h^{1T}x_i \end{bmatrix}, \quad (33)$$

kde h^{1T} je vektor odpovídající prvnímu řádku matice H . Transformovaný bod je v homogenních souřadnicích značen $x'_i = (x'_i, y'_i, w'_i)$.

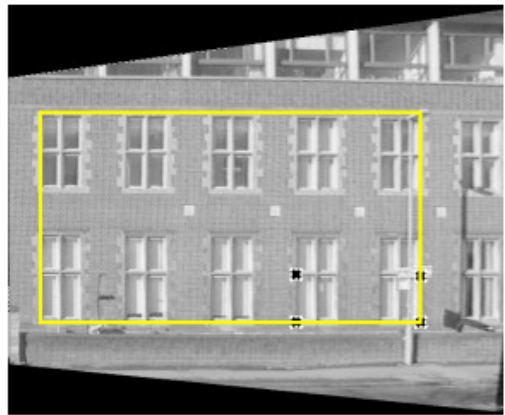
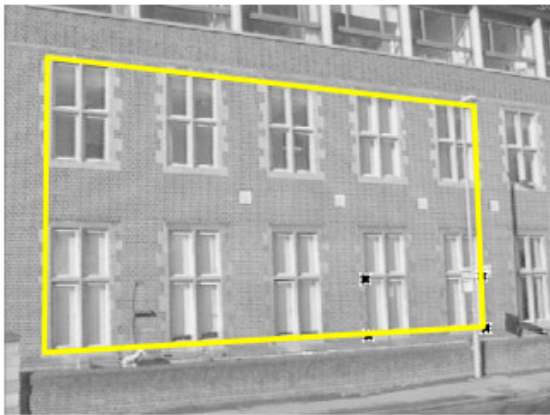
Maticový zápis soustavy lineárních rovnic pak po přepisu získá tvar:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (34)$$

pro bod i se označí soustava jako $A_i h = 0$, kde vektor h je sloupcový vektor 9×1 složený ze třech řádků matice H . Jelikož jsou ze 3 rovnic pouze jen 2 lineárně nezávislé, protože 3. rovnice je lineární kombinace první a druhé rovnice.

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (35)$$

Třetí homogenní souřadnice promítnutého bodu w'_i může být zvolena $w'_i = 1$. Při řešení soustavy rovnic $Ah = 0$ v případě pro 4 body, kdy je hodnota matice $A = 8$ existuje jedno řešení odpovídající pravému nulovému prostoru.



from Hartley & Zisserman

Obrázek č. 18: Ukázka homografie [8].

3 Odstranění pohybu

3.1 Motivace

Zadaným úkolem bylo vytvořit aplikaci fotoaparátu do telefonu, která bude schopna pořídit fotku scény, aniž by zaznamenala pohybující se objekty v popředí, například: osoby, zvířata, auta či jiné dopravní prostředky pohybující se ve scéně před uživatelem, a zachovat přitom pozadí, tedy vše, co se nepohybuje. Aplikace bude schopna tyto objekty ze scény automaticky odstranit při pořízení snímku. Aby mohla aplikace odstranit objekty ze scény je zapotřebí, aby objekty zcela opustily místo, kde se nacházely, když bylo snímání sekvence fotek zahájeno, jinými slovy objekt se musí pohnout natolik, aby aplikace zaznamenala prostor za ním.

Jelikož se jedná o reálnou úlohu, kdy uživatel pořizuje sekvenci běžným způsobem, kdy nastává přirozené chvění ruky je nezbytné jednotlivé snímky zarovnat na sebe tak, aby jednotlivé klíčové body pro všechny snímky odpovídaly stejným souřadnicím. Algoritmus, způsob jak toho docílit bude popsán v dalších kapitolách. To je nezbytný krok protože, aby bylo možné odstranit pohybující se objekty, je nezbytné rozlišit popředí od pozadí a následně z každého obrázku vyextrahovat konkrétní pozadí. Vzniklé obrázky budou mít v místě pohybujícího se objektu díru, tedy oblast, kde pixely budou mít nedefinovanou hodnotu pro všechny tři barevné kanály barevného obrázku. Díky zarovnání klíčových bodů snímků bude možné nahradit prázdná místa z následujících snímků, aby vznikl jeden konečný snímek bez děr, tedy bez černých oblastí.

K tomu aby bylo možné nalézt vytvořit konečný snímek, je potřeba rozlišit popředí od pozadí. K tomu poslouží algoritmus odečítání pozadí (background subtraction) ??, který rozpozná popředí od pozadí a následně vytvoří binární masku popředí, která umožní vyextrahovat popředí z obrázku a odečíst ho z obrázku samotného. Metody odečítání pozadí budou popsány v následujících kapitolách.

Jelikož bude docházet ke kombinaci několika různých snímků, je nezbytné mít pod kontrolou ostření fotoaparátu. Fotoaparát při pořízení fotek musí právě u prvního snímku zafixovat zaostření kamery na scénu, kterou pořizujeme a toto zaostření neměnit. Zároveň je nezbytné, aby nedošlo ke změně barev ve snímcích, je potřeba zafixovat parametry expozice, nastavené pro první snímek. Při splnění těchto předpokladů dojde ke zlepšení ve smyslu zachování stejné kvality jednotlivých snímků.

Na následující sérii obrázků 19 bude postupně ilustrován průběžný postup odstranění objektů ze snímku.



Obrázek č. 19: Příklad snímků pro odstranění pohybujících se objektů.

3.2 Homografie v Odstranění objektů

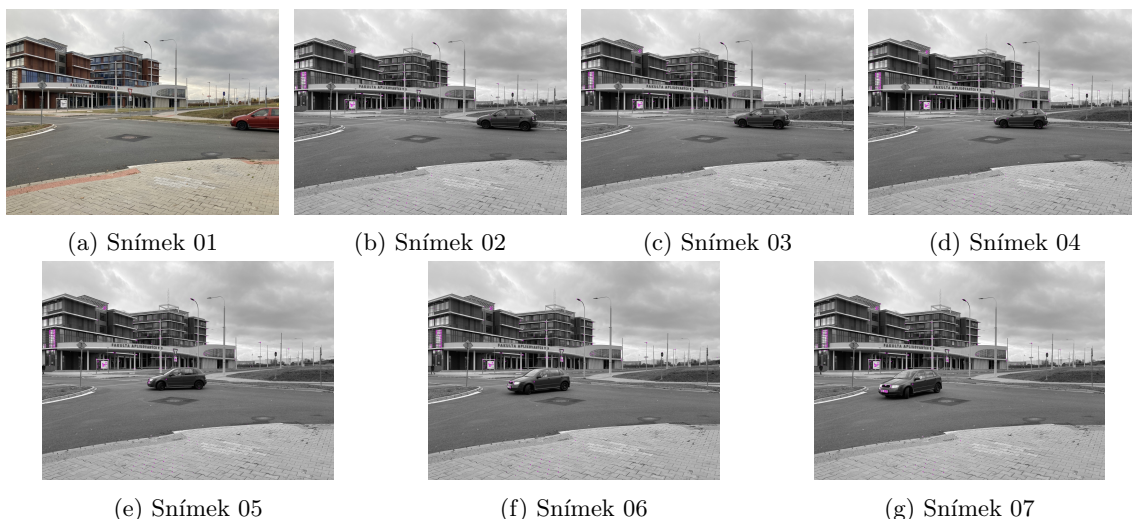
Jak bylo řečeno v úvodu této kapitoly 3.1, na sérii pořázených snímků je patřičně vhodné aplikovat principy homografie, viz kapitola 2.8. Pro provedení homografie je nezbytné nalézt klíčové body ve snímcích. K tomu se využije algoritmů představené v předchozí kapitole 2.1. Je důležité si připomenout, že stěžejním bodem pro uživatelsky přívětivou aplikaci je rychlost porřízení finálního snímku. K této podmínce se tedy přizpůsobí i výběr techniky pro nalezení klíčových bodů ve snímku. Při experimentálním měření bylo zjištěno, že na zarovnání obrázků na sebe stačí najít 1000 klíčových bodů při rozměru snímku 3024 x 4032 px. V následující tabulce 4 je možné vidět rozdíl časů potřebných k nalezení klíčových bodů. Metrikou pro nejlépe zvolenou metodu bude sloužit procentuální podíl času hledání klíčových bodů ku běhu celého algoritmu Z hlediska časů je tedy nejlepší zvolit z uvedených technik právě ORB.

	SIFT	ORB	BRISK *	AKAZE *
Homografie	15.289 s	1.082 s	160.245 s	35.117 s
Celková doba běhu	17.813 s	3.900 s	163.204	37.798 s
Procentuální poměr	85.33 %	27.75 %	98.19%	92.90 %

Tabulka č. 4: Čas potřebný na nalezení klíčových bodů ve všech snímcích.

* U techniky není možné omezit počet nalezených klíčových bodů.

Obrázek 20 vyobrazuje nalezené klíčové body pro jednotlivé snímky metodou ORB. Při hledání klíčových bodů je nezbytné, aby byly nalezeny i jinde než na objektu v popředí, který je v pohybu a na dalším snímku se nebude shodovat jeho poloha s polohou na snímku předešlém, proto je nezbytné zvolit takovou metodu pro hledání klíčových bodů ve snímku, která nalezne dostatek bodů v pozadí obrázku a bude možné provést homografii.



Obrázek č. 20: Vykreslení nalezených klíčových bodů.

Na obrázku 21 jsou k vidění snímky, na které byla aplikována homografie, tyto snímky mají po svých okrajích černý pruh v důsledku aplikování projektivní transformace vůči snímku prvnímu 21a.



Obrázek č. 21: Vykreslení nalezených zarovnaných obrázků.

3.3 Odečítání pozadí

Odečítání pozadí (Background subtraction) je první představenou metodou k detekci pohybujících se objektů v sekvenci snímků ze statické kamery. Základem tohoto přístupu je detekce pohybujících se objektů z rozdílu mezi aktuálním snímkem a referenčním snímkem. Toto odečítání pozadí se obvykle provádí detekcí objektů v popředí ve videu a detekce popředí je hlavním úkolem celého tohoto přístupu. V této práci však bude algoritmus využit.

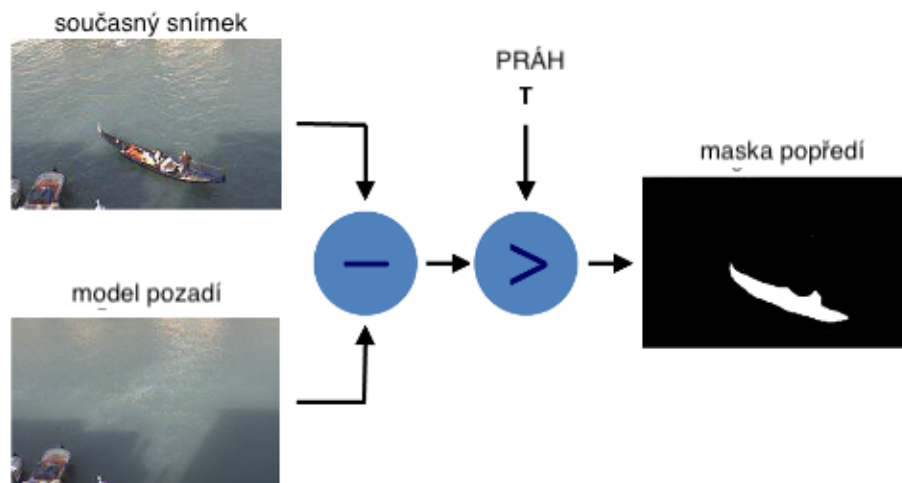
Všechny současné detekční techniky jsou založeny na modelování pozadí obrázku, tj. nastavení pozadí a detekci změn, ke kterým dochází. Definování správného pozadí může být velmi obtížné,

pokud obsahuje tvary, stíny a pohybující se objekty. Při definování pozadí se všemi technikami předpokládá, že se stacionární objekty mohou v průběhu času lišit barvou a intenzitou [38].

3.3.1 Časový průměr

Tento filtr odhaduje model pozadí z mediánu všech pixelů z předchozích sekvenčních obrázků. Využívá zásobník s hodnotami pixelů posledních dvou snímků k aktualizaci mediánu pro každý obrázek. Pro modelování pozadí systém zkoumá všechny sekvence v daném časovém úseku, během kterého se vypočítává medián pixelu po pixelu všech snímků.

Po natrénování je každý nový snímek a každá nová hodnota pixelu porovnána se vstupní hodnotou, která byla vypočtena, a pokud je vstupní pixel sledovaného snímku v mezích stanoveného prahu, pak je mapován jako pixel pozadí; jinak je mapován jako pixel v popředí.



Obrázek č. 22: Vizualizace metody odečtu pozadí [39].

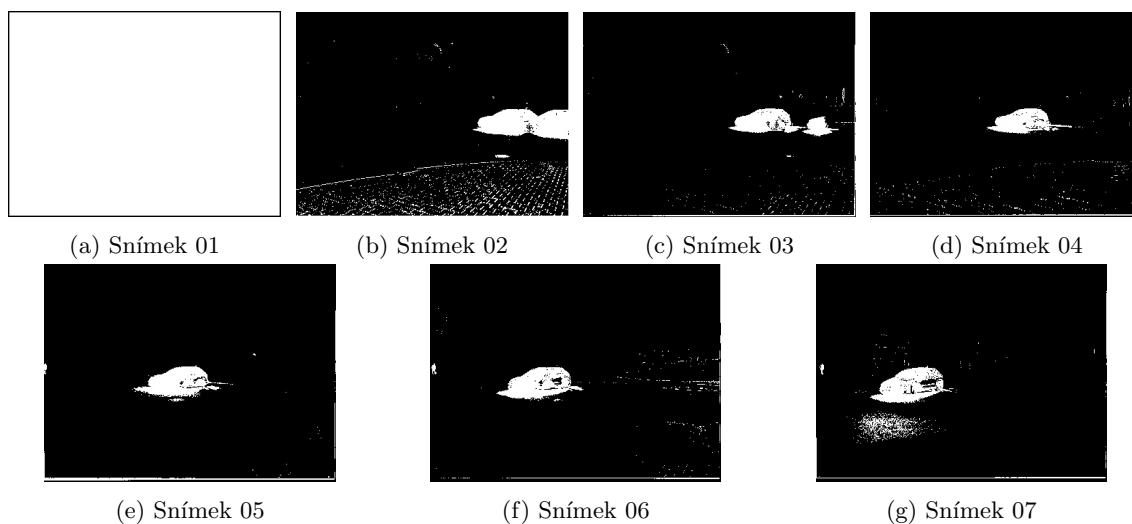
Tato metoda není tak účinná, jak by měla být, protože funguje na základě množství uložených dat, což vyžaduje vysoké výpočetní náklady a zároveň neobsahuje žádný statistický základ.

3.3.2 Model Gaussovských směsí

Model Gaussovských směsí (Mixture of Gaussians, MoG) je směsí k Gaussovským distribučním modelů pro každý pixel pozadí. předpokladem je, že každý pixel obrázku je reprezentován různou vahou. Váha každé z těchto distribucí použitých na modelu je úměrná době, po kterou každá barva zůstane na daném pixelu. Pokud je tedy váha rozložení pixelů nízká, pak je tento pixel klasifikován jako pixel v popředí.

3.3.3 Aplikace MoG

Aplikaci MoG pro představenou sérii snímků je představena na obrázku 23, jelikož algoritmus pracuje s pravděpodobností setrvání jasu konkrétního jasu na snímku, tak první snímek bude vždy automaticky celý jako popředí, protože všechny jasy jsou pro algoritmus nové. Na druhém snímku už bylo rozpoznáno pohybující se auto s tím, že zároveň pro MoG, bere za popředí i místo, kde stál objekt na snímku předchozím, jelikož jsou na tomto místě rovněž nové jasy. Na dalším snímku už je tento jev méně pozorován, avšak na snímcích zbylých už algoritmus rozeznává pohybující se objekt v popředí. Při aplikaci takovéto metody je zapotřebí uvažovat, že je zapotřebí několika snímků pro to, aby se algoritmus naučil rozpoznávat popředí od pozadí, proto bude v mobilní aplikaci jako minimální možný počet snímků potřebných k pořízení fotografie s odstraněným pohybem 5 snímků.



Obrázek č. 23: Vykreslení masky odečtení pozadí

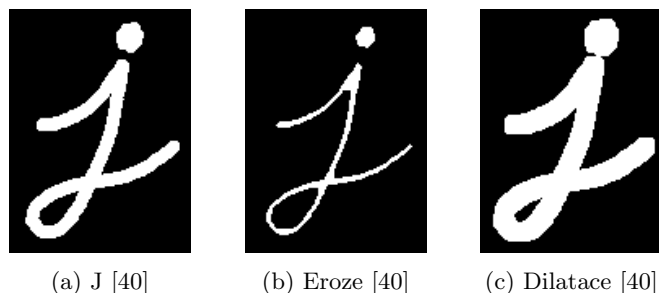
3.4 Rekonstrukce masky popředí

V předchozí kapitole byl představen způsob získání binární masky objektů, které jsou potřeba vyjmout ze snímku, avšak jak je vidět na obrázcích 19 maska objektu, v tomto případě auta, není kompletní, je proto potřeba přijít s opravnou technikou, aby bylo možné auto z obrázku kompletně odstranit.

3.4.1 Matematická morfologie

Morfologické transformace jsou jednoduché operace založené na tvaru obrazu. Obvykle se provádí na binárních obrazech a vyžaduje dva vstupy, jeden je původní obrázek, druhý se nazývá strukturální element nebo jádro, které rozhoduje o povaze operace. Dva základní morfologické operátory jsou eroze a dilatace. Pak přichází na řadu i jeho variantní formy jako je otevření nebo uzavření.

Operace jedna po druhé bude představena na následujícím obrázku :



Obrázek č. 24: Eroze vs. dilatace

3.4.1.1 Eroze

Základní myšlenka eroze je stejná jako u eroze půdy, eroduje hranice objektu v popředí, předpokládá se, že popředí má v masce hodnotu 1, tedy bílou barvu. Jádru klouže obrazem, podobně jako ve 2D konvoluci. Pixel v původním obrázku, buď 1 nebo 0, bude považován za 1, pouze pokud jsou všechny pixely pod jádrem 1, jinak je erodován.

Takže se stane, že všechny pixely blízko hranice budou vyřazeny v závislosti na velikosti jádra. Takže tloušťka nebo velikost objektu v popředí se zmenšuje nebo se jednoduše zmenšuje bílá oblast na obrázku. Je to užitečné pro odstranění malých bílých šumů, oddělení dvou spojených objektů atd.

Na obrázku 24b je jako příklad použito jádro 5x5 s plnými jedničkami.

3.4.1.2 Dilatace

Dilatace je pravý opak eroze. Zde je prvek pixelu '1', pokud je alespoň jeden pixel pod jádrem '1'. Zvětší tedy bílou oblast na obrázku nebo se zvětší velikost objektu v popředí. Normálně, v případech, jako je odstranění šumu, je eroze následována dilatací. Protože eroze odstraňuje bílé shluky, ale také zmenšuje objekt. Při odstranění šumu, proto dilatace navrátí velikost erodovanému objektu, je také užitečná při spojování rozbitých částí objektů.

3.4.1.3 Otevření

Otevření je jen jiný název eroze následovaná dilatací. Je užitečné při odstraňování šumu, jak bylo vysvětleno výše.



(a) Otevření [40]

(b) Uzavření [40]

Obrázek č. 25: Otevření vs. uzavření

3.4.1.4 Uzavření

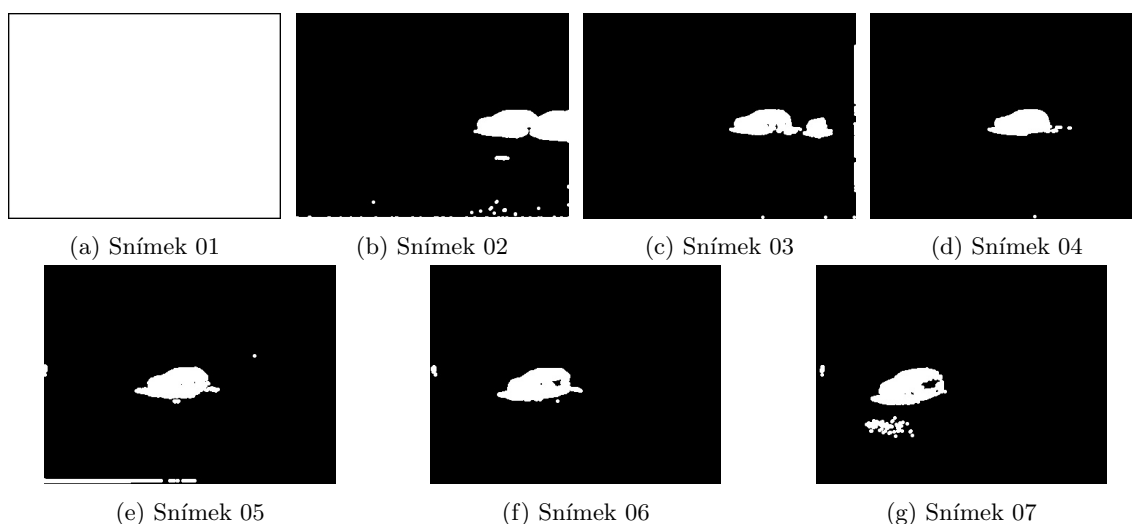
Uzavření je opak otevření, dilatace následovaná erozí. Je to užitečné při uzavírání malých otvorů uvnitř objektů v popředí.

3.4.2 Aplikace matematické morfologie

Na následujících řádcích bude popsán způsob jakým může být docíleno vylepšení masky za účelem vymaskování celého objektu v popředí, přičemž výsledek je možný vidět na 27.

V úvodu se aplikuje na masku otevření, které odstraní nechtěný šum z masky, ten je způsobený například drobnými rozdíly mezi jednotlivými snímky po aplikaci homografie, jelikož popředí se geometricky transformuje jinak než pozadí při aplikaci stejné matice H , protože v reálných situacích se nejedná o homografii, viz kapitola 2.8.

Následně se aplikuje uzavření, které spojí jednotlivé fragmenty masky objektu do jednoho kusu. Výsledek aplikace otevření s následným uzavřením může být viděn na obrázku 26.



(a) Snímek 01

(b) Snímek 02

(c) Snímek 03

(d) Snímek 04

(e) Snímek 05

(f) Snímek 06

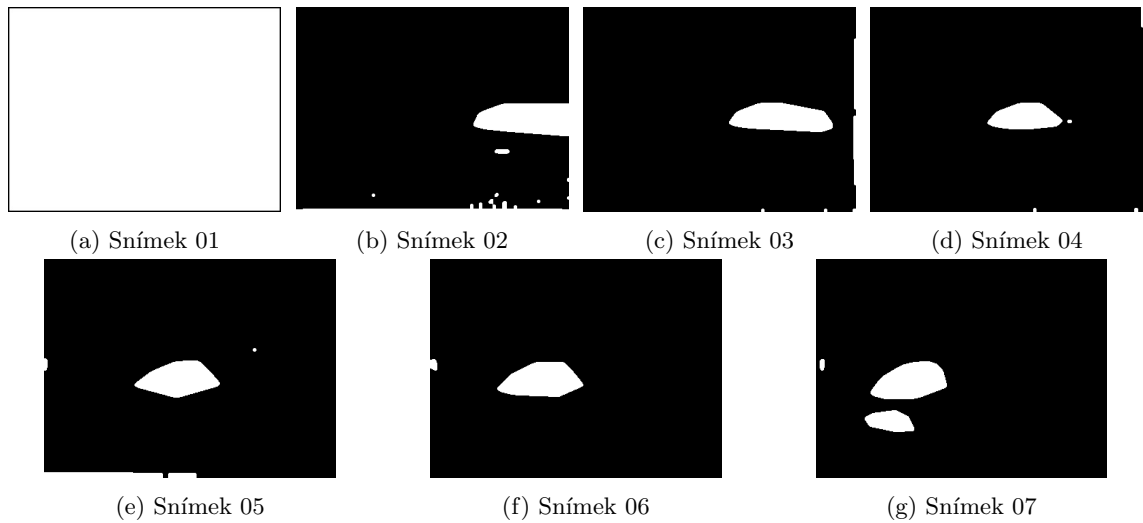
(g) Snímek 07

Obrázek č. 26: Vykreslení masky odečtení pozadí

Na snímcích 26f a 26g jsou stále vidět nedostatky ve vytvořených maskách. Tyto masky stále obsahují díry, zároveň ovšem může nastat případ, kdy díra v masce se nebude nacházet v jeho

středu, jak je v tomto případě, avšak může se vyskytovat i na jeho krajích, kdy by uzavření 25b nepomohlo. Lepším řešením než uzavření se proto skrývá ve využití v konvexních obalech kontur.

Kontury lze jednoduše vysvětlit jako křivku spojující všechny spojitě body podél hranice, které mají stejnou barvu nebo intenzitu. Obrysy jsou užitečným nástrojem pro analýzu tvaru a detekci a rozpoznávání objektů. Konvexní obal objektu je minimální hranicí, která může zcela uzavřít nebo obalit objekt a zároveň který nemá žádné vnitřní úhly větší než 180 stupňů. Takto získaná maska zcela vykryje objekt v popředí s minimální ztrátou obrazové informace v pozadí. Masky pro jednotlivé snímky jsou na obrázku 27.

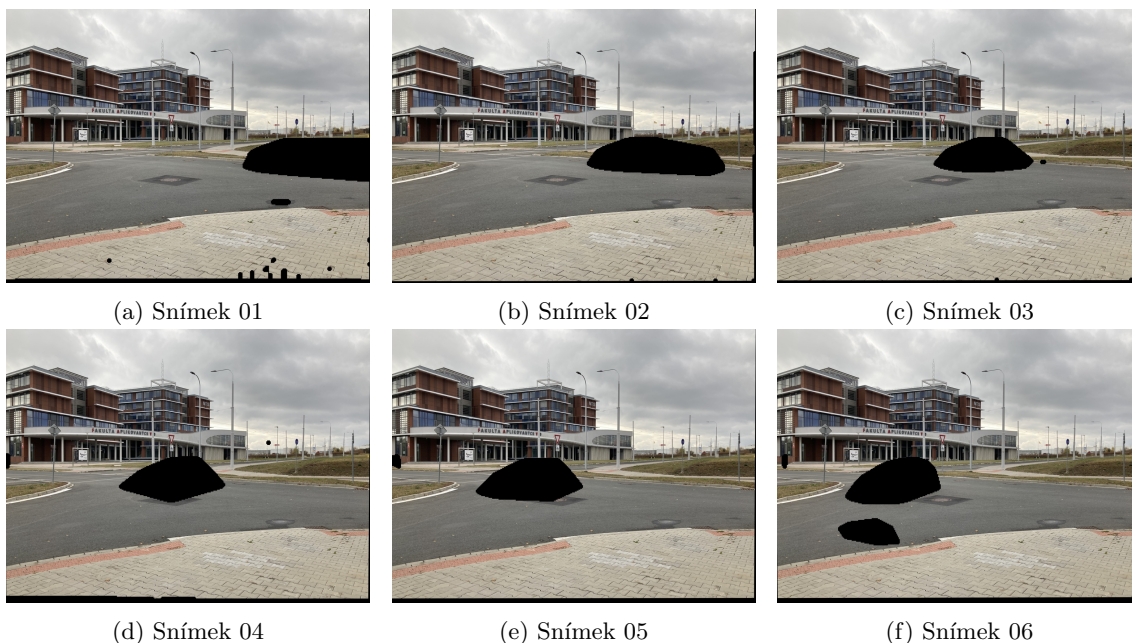


Obrázek č. 27: Vykreslení masky odečtení pozadí

3.5 Kompozice

V této kapitole dojde k samotné kompozici snímků do jednoho výsledného. V předchozí kapitole 2.8 byla získána maska popředí pro jednotlivé snímky, pomocí které je možno vyextrahovat pozadí jednotlivých snímků, které poté budou složeny do jednoho. Z důvodu použití algoritmu pro odečet pozadí 3.3 je informace z prvního snímku ztracena. Z původních 7 snímků, viz 23, po extrahování popředí se získá tedy snímků 6, jak je ukázáno na obrázku 28.

Na snímcích 28a a 28f jsou k vidění, že byly odstraněny i oblasti, které popředím nebyly, ať už se jedná o část chodníku na obrázku 28a nebo o část silnice na 28f. Pokud takto špatně vymazaná oblast je příliš velká, tak že by zakrývala většinou část snímku, dojde k znehodnocení samotného snímku, jinak se nic neděje a snímek bohatě poskytne zbylé obrazové informace.



Obrázek č. 28: Vykreslení masky odečtení pozadí

Z takto získaných snímků se získá obrazová záplata, která pak postupně zaplní díru v cílovém obrázku. Jako cílový obrázek, do kterého se budou vkládat záplaty je zvolený poslední obrázek. Důvodů je rovnou několik. Prvním důvodem je, že algoritmus pro odečet pozadí 3.3 je nejvíce natrénovaný, právě před posledním obrázkem, jelikož mu bylo poskytnuto více dat a tudíž šance na pohyblivý objekt v popředí, který by byl špatně odstraněn je minimální. Důvodem dalším je, že právě poloha objektů na prvním snímku a snímku posledním je nejvíce rozdílná, díky tomu je možné získat záplatu pro obrázek již z prvního použitelného snímku, jelikož právě, zde je největší pravděpodobnost, že pohybující se objekt v popředí nebude nalezen na daném místě v obrázku.

V tomto případě bude tedy referenčním snímkem snímek 28f a na záplatu budou použity snímky 28a až 28e. Díky aplikování záplat od prvního snímku do snímku posledního je vidět, že na vytvoření záplaty stačil první snímek 28a. Jak bude vypadat výsledná záplata je možné vidět na následujícím obrázku:



Obrázek č. 29: Záplata pro testovací obrázek

Zkombinují-li se tato získaná maska s posledním získaným snímkem 28f, dostane uživatel finální obrázek:



Obrázek č. 30: Finální snímek po odstranění pohybu

3.6 Optický tok

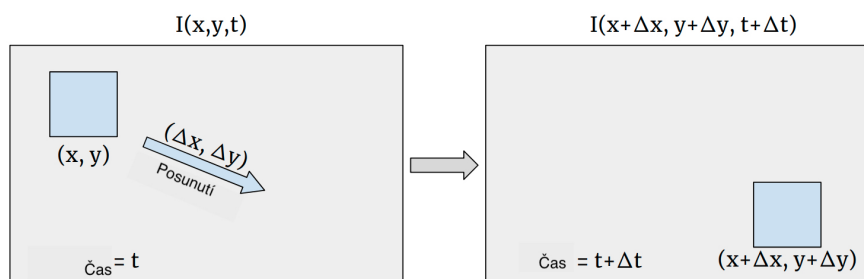
Optický tok je druhou představenou metodou pro odstranění pohybu. Optický tok metoda pro odhad pohybu pixelu mezi dvěma po sobě jdoucími snímky v jedné sekvenci snímků. Optický tok implikuje výpočet vektoru posunu pro pixel jako rozdíl posunutí objektu mezi dvěma sousedními obrazy. Hlavní myšlenkou optického toku je odhadnout vektor posunutí objektu způsobený jeho pohybem nebo pohyby kamery.

Uvede-li se v předpoklad, že je obrázek v šedotónové reprezentaci, takovýto obrázek tvoří matici s intenzitou pixelů. Zadefinuje-li se funkce $I(x, y, t)$, kde x, y jsou souřadnice pixelu a t je číslo snímku. Funkce $I(x, y, t)$ definuje přesnou intenzitu pixelů na snímku t .

Předpokládá-li se, že posunutí objektu nemění intenzitu pixelů, která patří přesnému objektu, to znamená, že

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \quad (36)$$

V takovém případě $\Delta t = 1$. Hlavním úkolem je tedy najít pohybový vektor $(\Delta x, \Delta y)$. Grafické znázornění je možné vidět na obrázku 31:



Obrázek č. 31: Posunutí objektu [36]

Pomocí rozvoje do Taylorovy řady je možné přepsat

$$I(x, y, t) - I(x + \Delta x, y + \Delta y, t + \Delta t) = 0 \quad (37)$$

jako

$$I'_x u + I'_y v = -I'_t, \text{ kde } u = \frac{dx}{dt}, v = \frac{dy}{dt} \quad (38)$$

a I'_x, I'_y jsou přechody obrázků. Je důležité, že je zde předpokládáno, že části Taylorovy řady vyššího řádu jsou zanedbatelné, takže se jedná o aproximaci funkcí pouze pomocí Taylorova rozvoje prvního řádu. Rozdíl pohybu pixelů mezi dvěma snímky $I_1 a I_2$ lze tedy zapsat jako

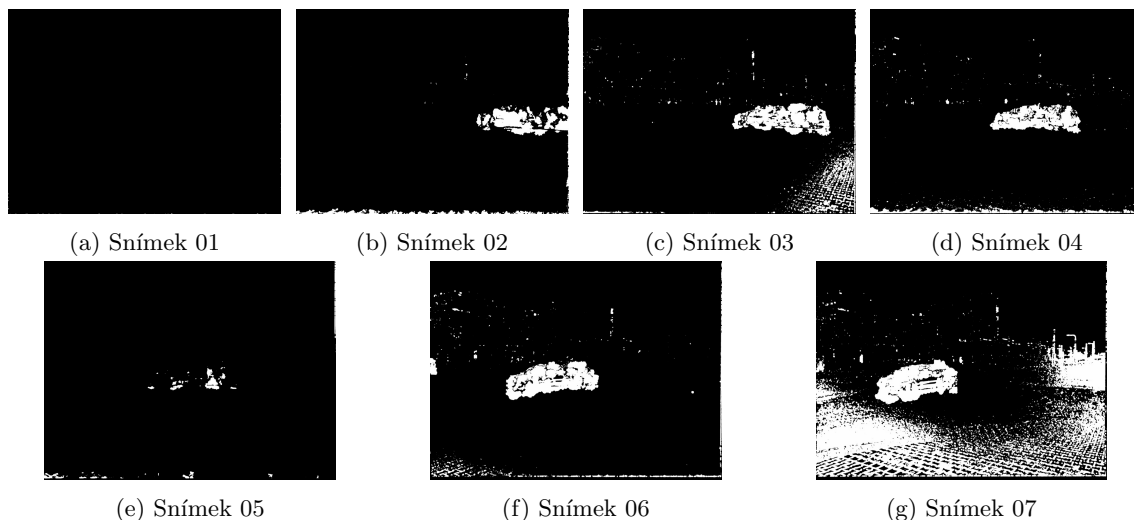
$$I_1 - I_2 \cong I'_x u + I'_y v + I'_t. \quad (39)$$

Metoda Lucas-Kanade počítá optický tok pro řídkou sadu prvků, která detekuje rohy objektu. Pro detekci objektů je zapotřebí využít algoritmus pro výpočet hustého optického toku, který

Odečítání pozadí	2.32 s
Optický tok	16.54 s

Tabulka č. 5: Porovnání rychlostí

je počítán pro všechny pixely v obrázku. Je založen na algoritmu Gunnara Farnebacka, viz [?]. Získá-li se 2-kanálové pole s vektory optického toku (u, v) . Zjistíme jejich velikost a směr. Výsledek barevně kódujeme pro lepší vizualizaci.



Obrázek č. 32: Detekce pohybu pomocí optického toku

Jedna z nevýhod této metody je, že podmínkou pro detekci objektů je pohyb objektů, jakmile tedy dojde k zastavení nebo příliš velkému zpomalení objektů v obraze, metoda je nerozpozná a považuje je jako součást pozadí. Dané snímky se pak stávají neplatnými. Největší nevýhodou tohoto přístupu je jeho výpočetní náročnost, doba výpočtu je tedy v porovnání s metodou odečítání pozadí mnohem delší. V tabulce je vidět, časový rozdíl obou metod:

Z důvodu že metoda optického toku je přibližně 8x pomalejší se metoda jeví jako nepoužitelná pro přístup odečítání pozadí, proto nebude následně dále rozebírána.

3.7 Vyhodnocení

V předchozích kapitolách pro odečet pozadí 3.3 a optický tok 3.6 bylo ukázáno, že jako nejvhodnějším řešením z hlediska výpočetních nároků se pro odstranění pohybu v představeném přístupu aplikace je odečet pozadí.

Následně je potřeba ověřit v jaké kvalitě bylo výsledku odečtu pozadí dosaženo.

Pro určení kvality odstranění snímku poslouží metrika Střední kvadratická chyba, snímek s odstraněnými objekty s odpovídajícím výstupním snímkem stejné scény bez objektů. Snímky musí být zarovnány, takže je potřeba aplikovat homografii, poté následně porovnat společnou část snímků.

3.7.1 Objektivní vyhodnocení

Jako metrika pro objektivní vyhodnocení byla zvolena střední kvadratická chyba (Mean Square Error, MSE) estimátoru měří průměr druhých mocnin chyb – to znamená průměrný čtvercový rozdíl mezi odhadovanými hodnotami a skutečnou hodnotou. Čím je její hodnota menší, tím je rozdíl ve snímcích menší.

$$MSE = \frac{1}{mn} \sum_{i=0}^m - \sum_{j=0}^n -1(I_{i,j} - K_{i,j})^2, \quad (40)$$

kde K je odpovídající výstupní obrázek a I je porovnávaný obrázek, (m, n) je velikost snímků.

3.7.2 Porovnání podle MSE

Odpovídající referenční obrázek (v ang. ground thruth), který je bez pohybu a slouží pouze k porovnání metody. Pro srovnání snímků je nezbytné zarovnat vytvořený snímek na snímek referenční. Na obra'ázku jsou vidět porovnávané snímky:



(a) Upravený získaný snímek



(b) Odpovídající výstupní snímek

Obrázek č. 33: Porovnání výstupního snímku se snímek upraveným

V tabulce 6 je výsledek MSE skóre po vyhodnocení podobnosti mezi výstupním obrázkem a výsledným snímekem po odstranění pohybujících se objektů.

Metoda	MSE
Vyhodnocení	130

Tabulka č. 6: MSE skóre

3.7.3 Subjektivní validace na testovacích sadách dat

V této kapitole budou probrány různé výsledné snímky po odstranění objektů, které je možné aplikací pořídit. Pro porovnání zde budou ještě uvedeny kombinace původního snímku a snímku finálního.

Prvním příkladem je porovnání výsledku z odvozovacího příkladu, ze snímku bylo zcela odstraněno auto i osoba nacházející se v levé krajní části snímku.



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 34: Porovnání snímků před a po úpravě

Experimentu byl rovněž podroben průchod osoby ve scéně, kde i algoritmus dobře zafungoval, jelikož, zde byl splněný předpoklad, že se objekt pohybuje podél uživatele a ne kolmo od něj.



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 35: Porovnání snímků před a po úpravě

Na následujícím obrázku 36a je vidět, že si algoritmus poradil i s velkým objektem, jako je kamion. Výsledný obrázek je ovšem zřetelně zúžený v důsledku aplikace transformační matice na snímky, kde se projevil třes v rukou uživatele.



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 36: Porovnání snímků před a po úpravě

Na následujících dvou příkladech 37a a 38a, že algoritmus dobře funguje i za zhoršených světelných podmínek, kde se podařilo odstranit více objektů.



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 37: Porovnání snímků před a po úpravě



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 38: Porovnání snímků před a po úpravě

Na posledním příkladu 39 je pohled na dálnici z dálničního mostu, kde se auta pohybují kolmo od uživatele, algoritmus dokázal detekovat i vozy daleko v pozadí a vše odstranit.



(a) Snímek po odstranění objektů v popředí



(b) Snímek před odstraněním objektů v popředí

Obrázek č. 39: Porovnání snímků před a po úpravě

4 Super kompozice

4.1 Motivace

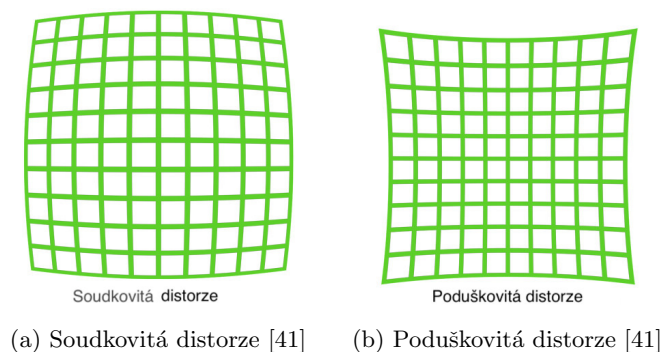
Dalším funkcí aplikace fotoaparátu do telefonu, bude možnost pořídít snímek objektů nebo scény, jež je příliš velká na to, aby ji čočka objektivu mohla zaznamenat vcelku.

4.1.1 Širokoúhlý objektiv

Pro takovéto případy jsou mobilní fotoaparáty často vybaveny dalším objektivem, který je širokoúhlý a objekty se tam tedy vejdu celé. Při fotografování subjektů se širokoúhlým objektivem je docela běžné, že dochází k soudkovité distorzi, takové problémy s perspektivou se projevují ve formě čar, které by měly být rovné, ale jsou zakřivené a zkreslené. Toto zkreslení je právě tím výraznější čím víc je objektiv širokoúhlý, protože zvětšení obrazu klesá, čím dál je objekt od optické osy objektivu. Na takovéto případy existují různé filtry, které čáry vyrovnají, avšak po takovéto úpravě budou hrany snímku zakulacené a uživatel přijde při zarovnávání o informace ve snímku. Další nevýhodou je, že uživatel často nemá možnosti si libovolně měnit poloměr stran a proto zachytí do širokoúhlého snímku i části pozadí uživatelem nechtěné.

Distorze

Nejčastějšími typy distorzí, které mohou nastat jsou



Obrázek č. 40: Typy distorzí.

4.1.1.1 Soudkovitá distorze

Soudková distorze je spojena se širokoúhlými objektivy a širokoúhlými objektivy se zoomem na kratších ohniskových vzdálenostech. Jde o vadu čočky, která způsobuje ohýbání přímých linií směrem ven a je nejvíce patrná podél okrajů. V extrémních případech vypadá střed obrazu, jako by se ohýbal směrem k nám.

Při kontrole rovnoběžných čar na fotografii, zda čáry vedou paralelně nebo rovnoměrně, což je dobrý způsob, jak určit soudkovité zkreslení. Zakřivený tvar čočky vytváří soudkovité zkreslení.

Vzhledem k tomu, že čočka je zakřivená, zvětšuje střed rámečku výrazně více než strany, díky čemuž se kolem okraje obrazu zdají rovné čáry.

Jak můžete vidět na tomto obrázku 40a, čáry se objevují přímo ve středu obrázku a teprve se začínají ohýbat směrem od středu. Je to proto, že objektivy fotoaparátů mají nejmenší zkreslení podél optické osy (střed objektivu) a zvětšení směrem k rohům se snižuje.

Soudkové zkreslení nastává, když čočka ohýbá světlo ze širokého zorného pole. To je obvykle případ širokoúhlých objektivů, které způsobují ohýbání okrajů, aby dosáhly k obrazovému snímáči.

Širokoúhlé objektivy mají obecně více sférického skla než teleobjektivy. Jak název napovídá, mají velký radius pro fotografování široké oblasti a ohýbání světla do obrazového snímáče. Typicky je to případ krajinářských fotografií pořízených širokoúhlými objektivy.

Soudkovité zkreslení je způsobeno zmenšením zvětšení obrazu od středu optické osy k okrajům.

Objektivy typu rybí oko často způsobují soudkovité zkreslení. Některé objektivy typu rybí oko jsou tak extrémní, že tvar obrazu je spíše kruhový než konvenční obdélníkový tvar. Přijdou vhod při fotografování stísněných prostor, jako je interiér domu pro fotografování nemovitostí.



Obrázek č. 41: Soudkovitá distorze před opravou [41].

4.1.1.2 Poduškovitá distorze

Poduškovitá distorze je opakem soudkové distorze. Způsobuje zakřivení přímých čar směrem dovnitř od okrajů ke středu. Teleobjektivy a teleobjektivy se zoomem jsou nejčastěji spojovány s poduškovitou distorzí.

Ke zkreslení obvykle dochází na teleobjektivu. Delší ohniskové vzdálenosti zvyšují zvětšení obrazu od optické středové osy k okrajům. Jak již bylo zmíněno dříve, soudkovité zkreslení způsobuje snížení zvětšení obrazu od středu.

Podobně jako u soudkovité distorze je poduškovitá distorze nejvíce patrná na fotografiích s rovnými liniemi. Přiblížené snímky teleobjektivem způsobí tuto optickou aberaci. Příkladem poduškovité distorze je:



Obrázek č. 42: Opravená poduškovitá distorze [41].

4.1.2 Panorama

Dalším řešením se naskýtá použití panorama snímku, který bývá jako softwarová součást nativních aplikací fotoaparátů chytrých mobilních telefonů. Panoramatická fotografie je technika, která kombinuje více snímků ze stejného otočného fotoaparátu do jediné, širokoúhlé fotografie. Zachycuje snímky s horizontálně nebo vertikálně protáhlými poli. Poté, co se otočí fotoaparát, aby byl vytvořen plný efekt 360 stupňů nebo méně, spojí se tyto snímky dohromady, aby byla získána panoramatická fotografie. Zde však jak již název napovídá není možné pořídit fotografii v jiném tvaru než v protáhlém a to buď ve směru vertikálním či horizontálním.

4.1.3 Problém sešívání obrazů

Alternativou se tedy stává nová funkce pro super kompozici. Uživatel bude moci postupně nafotit části objektu, po kterém chce, aby utvářel samotný snímek. Příkladem může být katedrála, mrakodrap, ale i přírodní scenérie. Podmínkou však je správná šířka/výška výsledného složeného snímku ku snímku dílčímu, jelikož při příliš velkém množství složených snímků vedle sebe či nad sebou rovněž dojde k deformaci výsledného snímku.

Takový způsob spojování se nazývá sešívání obrazů (z ang. image stitching). Na začátku procesu sešívání je jako vstup několik obrázků s překrývajícími se oblastmi. Výstupem je sjednocení těchto obrázků. Je důležité si uvědomit, že v procesu musí být zachována celá scéna ze vstupního obrazu.

Proces vytváření složeného obrazu se skládá z následujících kroků:

- Detekce klíčových bodů a deskriptorů
- Detekce sady shodných bodů, která je přítomna na obou snímcích (překrývající se oblast)

- Aplikování perspektivní transformace na jeden obraz s použitím druhého obrazu jako referenčního snímku
- Spojení obrázků dohromady

4.2 Kompozice

Pro kompozici několika snímků do jednoho se využije již představené homografie, viz kapitola 2.8, která vyžaduje nalezení klíčových bodů ve snímku. Kvalita složení snímků k sobě se bude odvíjet od kvality, počtu a rozložení klíčových snímků a zároveň bude záležet na tom, jak moc se budou jednotlivé snímky překrývat, aby mohla být na jednotlivé snímky aplikovaná homografie, je potřeba, aby primární snímek, který bude tvořit jádro snímku rozšířeného, obsahoval společné klíčové body se sekundárním snímkem, který bude přidružen k primárnímu [43].

Kompoziční efekt se dá vytvořit s následujícími obrázky:



Obrázek č. 43: Snímky pro super kompozici obrazu.

Prvním krokem v procesu vytváření kompozice snímků je zarovnání páru snímků. V zásadě se vezmou klíčové body na jednom obrázku a spojí se s prvky na druhém obrázku:



(a) Levý obrázek



(b) Pravý obrázek

Obrázek č. 44: Levý a pravý obrázek pro spojení.

Aby bylo možné dosáhnout cíle a vytvořit kompozici snímku, je potřeba se ujistit, že mezi těmito snímky existuje společná oblast. Důvodem je, že v této překrývající se oblasti by měla být detekována dostatečně silná sada charakteristických klíčových bodů, proto se vždy vyfiltruje maximálně 40% těch nejlepších klíčových bodů se kterými se bude následně uvažovat.

Na následujícím obrázku 45 jsou znázorněny klíčové body obsahující oba dva snímky:

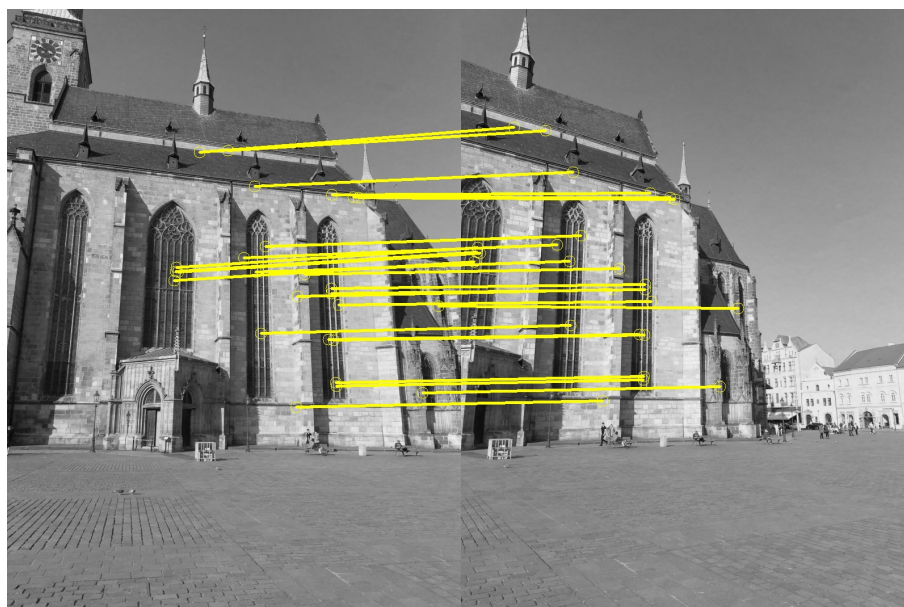


(a) Levý obrázek

(b) Pravý obrázek

Obrázek č. 45: Levý a pravý obrázek po spojení.

Na obrázku 46 je vykresleno 30 nejvýznamnějších klíčových bodů shodných pro oba dva snímky na obrázku.

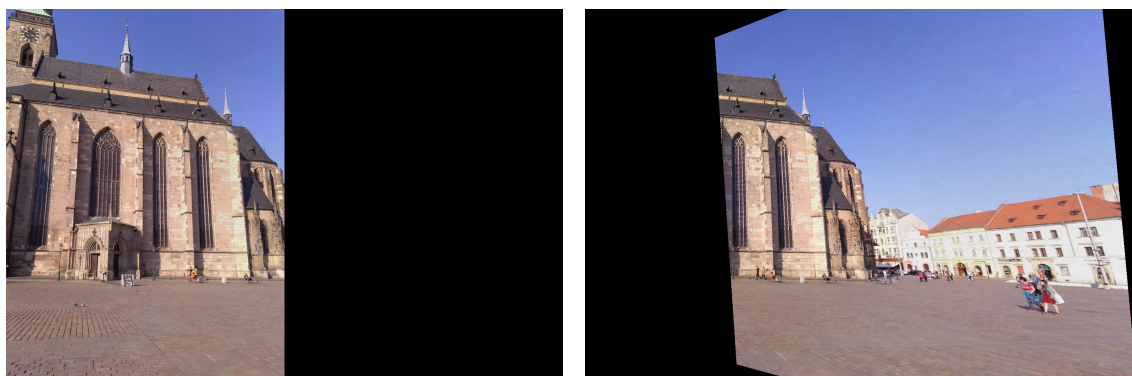


Obrázek č. 46: Shodné klíčové body.

Pak je potřeba použít perspektivní transformaci v případě, že dva obrázky nejsou umístěny ve stejné rovině. Taková to transformace je potřeba použít pokaždé, kdy jsou z jednoho bodu

vyfoceny dva různé snímky. První snímek se použije jako referenční snímek a druhý snímek se pokříví tak, aby byly klíčové body na obou snímcích dokonale zarovnány [43].

Podmínkou použití perspektivní transformace je definice velikosti cílového obrázku 47a, jelikož aplikací perspektivní transformací dojde, jak k transformaci obrázku 47b, tak i k jeho posunutí vůči referenčnímu obrázku [43].



(a) První referenční snímek

(b) Druhý transformovaný snímek

Obrázek č. 47: Perspektivní transformace.

Jakmile se rozložení klíčových snímků shoduje s identickými klíčovými body na obou obrázcích, můžeme je snadno překrýt, jak můžete vidět na následujícím obrázku.



Obrázek č. 48: Výsledný obrázek po sloučení dvou snímků.

4.3 Porovnání metod

V předchozí kapitole bylo ukázáno, jak docílit sešitím dvou obrazů dohromady, využitím homografie 2.8 a klíčových bodů 2.1, avšak je třeba ukázat jaká metoda je v tomto případě nejlepší z hlediska přesnosti, která je stěžejní pro tuto funkcionalitu.

Pro porovnání metod poslouží metrika Střední kvadratická chyba (MSE) 3.7.1, která porovná jednotlivé transformované snímky transformované perspektivní transformací automaticky získanými klíčovými body pro homografní matici. Tyto snímky budou porovnány se referenčním snímkem získaným ručně vybranými klíčovými body pro homografní matici.

Porovnání podle MSE

Odpovídající výstupní obrázek (v ang. ground thruth):



Obrázek č. 49: Odpovídající výstupní obrázek.

..

V tabulce ?? je k vidění porovnání jednotlivých technik pro nalezený klíčových bodů sloužící k perspektivní transformaci s odpovídajícími výstupními daty. Jak je v tabulce vidět, tak co se týče přesnosti je nejlepší technika AKAZE. Test byl proveden na 500 nejlepších společných klíčových bodech pro oba snímky (referenční, transformovaný).

Technika	MSE
SIFT	60.5
ORB	67.1
BRISK	66.0
AKAZE	59.8

Tabulka č. 7: Porovnání přesnosti technik

4.4 Výsledné snímky super kompozice

V této kapitole budou probrány výsledné super kompozice, které je možné aplikací pořídit. Jasové korekce aplikované na snímky jsou popsány v semestrální práci věnující se této problematice [42].



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04



(f) Snímek 05



(g) Snímek 06



(h) Snímek 07



(i) Snímek 08

Obrázek č. 50: Příklad super kompozice 1.



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04

Obrázek č. 51: Příklad super kompozice 2.

Na následujícím příkladu ?? je k vidění, že pokud dojde při snímání fotek ke změně osvětlení scény nebo pokud nebude zafixovaná expozice, jako je na tomto obrázku, můžou vzniknout takovéto stínové pruhy.



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04



(f) Snímek 05

Obrázek č. 52: Příklad super kompozice 3.

Stejný problém se změnou osvětlení scény nastavil i na obrázku 53a, kde je vidět, že super kompozice může sloužit i jako alternativa pro panoráma snímek.



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04



(f) Snímek 05



(g) Snímek 06



(h) Snímek 07

Obrázek č. 53: Příklad super kompozice 4.

Super kompozice může být použita i na oskenování billboardu nebo plakátu.



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04

Obrázek č. 54: Příklad super kompozice 5.

Na posledním příkladu ?? je patrné roztažení střechy u benzínové pumpy. Takováto chyba vznikne pokud se daný snímek má zarovnat do perspektivní roviny snímku se kterým svírá příliš velký úhel.



(a) Výsledná super kompozice



(b) Snímek 01



(c) Snímek 02



(d) Snímek 03



(e) Snímek 04

Obrázek č. 55: Příklad super kompozice 6.

4.4.1 Kompozice vs. panorama

Nakonec bude ukázáno, v čem se liší super kompozice oproti nativní funkci panorama. Na obrázku 56 je porovnání snímků pořízených super kompozicí a panoramem. Na obou snímcích je patrný přechod mezi změnou světla



(a) Super kompozice



(b) Panorama

Obrázek č. 56: Rozdíl super kompozice vs. panorama.

Při focení blízkých ploch vzhledem k uživateli, dochází přibližně v polovině snímku opět k jeho transformaci vzhledem k rovině billboardu a působí tak, že billboard je zatočený.



(a) Super kompozice



(b) Panorama

Obrázek č. 57: Rozdíl super kompozice cs. panorama.

U posledního příkladu ?? je možné vidět, že po kompoziční stránce se funkce super kompozice téměř vyrovná panoramatu. Avšak i zde je vidět, že panorama transformuje navazující snímky tak, aby se vešly do záběru a proto, strom napravo je menší v případě super kompozice, kde jsou snímky oříznuté.



(a) Super kompozice



(b) Panorama

Obrázek č. 58: Rozdíl super kompozice cs. panorama.

5 Filtr obrazové sekvence

5.1 Motivace

V posledním představeném modu nesoucí název Filtr obrazové sekvence, bude moci uživatel po stisknutí tlačítka začít zaznamenávat scénu, kdy telefon, bude ve smyčce pořizovat snímky, dokud se uživatel nerozhodne snímkování zastavit. Následně dostane k dispozici posledních 30 zaznamenaných snímků.

Taková to funkcionality je možná najít v obdobném řešení v běžných telefonech a v jejich nativních aplikacích pro fotoaparát, často pod položkou sportovní režim. Kdy uživatel po stisknutí tlačítka zaznamená sekvenci fotek, které se rovnou uloží do paměti.

Tato funkce se ovšem bude lišit tak, že uživatel bude mít možnost si vybrat, zde mu aplikace vyfiltruje rozmazané fotky, podle daného prahu nebo vyfiltruje fotky opakující se několikrát, např. kdy uživatel čeká než nastane změna ve scéně. Ze zbylých snímků poté dostane uživatel na výběr fotky, kde si může vybrat, která se mu zamlouvá.

5.2 Filtr rozostřených fotek

Na obrázky se aplikuje algoritmus pro nalezení klíčových bodů a cílem je podle počtu nalezených klíčových snímků rozeznat, zda se jedná o snímek rozmazaný, či nikoliv. Po spočítání klíčových bodů je potřeba stanovit prahovou hodnotu, která rozhodne o tom, zda je snímek rozostřený, či nikoliv.

Nejdříve se určí referenční hodnota pro určení prahu, ta se určí jako průměrný počet klíčových bodů ve snímku z prvních pěti snímků s nejpočetnější hodnotou klíčových bodů. Prahová hodnota počtu klíčových bodů pro nerozostřený snímek je stanovena na 70% z referenční hodnoty.

Příklad

Existuje-li seznam 10 snímků s následujícím počtem klíčových bodů:

snímek	01	02	03	04	05	06	07	08	09	10
klíčové body	50	79	222	39	209	132	79	32	87	95

Tabulka č. 8: Příklad na počet klíčových bodů ve snímku.

Referenční hodnotu bude tedy tvořit průměr čísel: 03,05,06,09,10 a výpočet je proto následující:

$$REF = \frac{232 + 219 + 187 + 87 + 95}{5} = 164.$$

Prahová hodnota bude 70% z referenční hodnoty REF .

$$TRES = REF \cdot 0,7 \doteq 115.$$

Algoritmus tedy následně detekuje snímky 9 jako rozostřené.

snímek	01	02	03	05	08	09	10
klíčové body	50	79	39	79	32	87	95

Tabulka č. 9: Příklad na výsledky po filtraci.

Jak bylo ukázáno na příkladě, stejným principem budou vybrány snímky ze sekvence, která je vidět na obrázku ?? ze kterých se vyfiltrují rozostřené snímky. Proto aby bylo možné definovat hranici pro rozpoznávání rozostřených snímků, byly v této sekvenci snímky náhodně uměle rozostřeny gaussovským filtrem o proměnlivé velikosti jádra, aby bylo rozostření různorodé. Rozmazány byly následující snímky: 01, 03, 04, 09, 12, 16, 17, 20.



Obrázek č. 59: Obrazová data před vyfiltrováním.

Následně budou prozkoumány představené techniky pro automatické hledání klíčových bodů v kapitole 2.1. Jednotlivé techniky budou porovnány přes F1 skóre.

F1 skóre

F1 skóre kombinuje přesnost a úplnost klasifikátoru do jediné metriky tím, že vezme jejich harmonický průměr. Primárně se používá k porovnání výkonu dvou klasifikátorů. Předpokládá-li se, že klasifikátor A má vyšší úplnost a klasifikátor B má vyšší přesnost. V tomto případě lze skóre

F1 pro oba klasifikátory použít k určení, který z nich poskytuje lepší výsledky.

Skóre F1 klasifikačního modelu se vypočítá následovně:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}, \quad (41)$$

kde P znamená přesnost a R úplnost.

Přesnost je zlomkem skutečných pozitivních obrazů mezi obrazy, které model klasifikoval jako pozitivní. Jinými slovy, počet skutečně pozitivních výsledků vydělený počtem falešně pozitivních plus skutečných pozitivních výsledků.

$$P = \frac{tp}{tp + fp}, \quad (42)$$

kde tp je počet skutečných pozitivních výsledků klasifikovaných modelem a fp je počet falešně pozitivních výsledků klasifikovaných modelem.

Úplnost je zlomek obrazů klasifikovaných jako pozitivní z celkového počtu pozitivních obrazů. Jinými slovy, počet skutečně pozitivních vydělený počtem skutečně pozitivních plus falešně negativních.

$$R = \frac{tp}{tp + fn}, \quad (43)$$

kde fn je počet falešně negativních výsledků klasifikovaných modelem.

V následující tabulce 12 je porovnána účinnost jednotlivých algoritmu pro automatické získání klíčových bodů pomocí F1 skóre pro různé nastavení prahu. Jako nejlepší řešení se jeví využití ORB a BRISK, které jako jediné dosáhli ve skóre hodnoty 1,0, přičemž se BRISK jeví jako nejcitlivější z hlediska rozmazanosti snímku, jelikož takové hodnoty dosáhl při různém nastavení prahu.

technika	SIFT	ORB	BRISK	AKAZE
F1 (90%)	0,07	0,92	0,76	0,32
F1 (80%)	0,07	1,0	1,0	0,76
F1 (70%)	0,07	0,96	1,0	0,84
F1 (60%)	0,56	0,96	1,0	0,89

Tabulka č. 10: F1 skóre pro různé nastavení prahu u jednotlivých technik.

Použije-li se ve filtru algoritmus BRISK a nastavení prahu bude referenční hodnota - 70% z ní. Zůstanou následující snímky na obrázku 60



Obrázek č. 60: Výsledek po aplikaci filtru na rozostření snímky.

5.3 Filtr opakujících se fotek

Po aplikaci algoritmu na nalezení klíčových bodů se využije právě získaných deskriptorů, které uchovávají hodnoty pro souřadnice klíčového bodu ve snímku. Pro společné klíčové body snímků i a $i+1$, kdy i je pořadí snímku, se nalezne vzdálenost mezi dvojicí deskriptorů odpovídající právě této dvojici snímků.

Stanoví-li se práh pro průměrnou vzdálenost dvojic deskriptorů, která rozhoduje o podobnosti dvou snímků. Je poté možné rozpoznávat podobné snímky.

V následující tabulce je uvedeno, jaké snímky by mohl uživatel subjektivně vyfiltroval a jaké nechat, podle takového rozhodnutí poté následně bude nastaveno kritérium pro filtraci. Vycházet se bude opět ze snímků na obrázku 59, jelikož budou oba filtry na sobě nezávislé.

snímek	1	2	3	4	5	6	7	8	9	10
rozhodnutí	1	1	0	1	1	0	1	0	0	1
	11	12	13	14	15	16	17	18	19	20
	1	1	0	0	0	0	1	1	1	1

Tabulka č. 11: Vybrané snímky, co mají být vyfiltrovány.

Následně podobně jako v předchozí kapitole 5.2 bude pomocí F1 skóre vybrán nejlepší možný práh tak, seznam vyfiltrovaný snímků se blíží k vlastnímu subjektivnímu vybrání.

technika	SIFT	ORB	BRISK	AKAZE
F1 (80)	0,17	0,50	0,73	0,31
F1 (70)	0,43	0,81	0,71	0,76
F1 (60)	0,76	0,71	0,71	0,79
F1 (50)	0,84	0,71	0,71	0,0,73

Tabulka č. 12: F1 skóre pro různé nastavení prahu u jednotlivých technik.

Podle F1 skóre vychází jako nejlepší volba pro filtrování snímků použití metody ORB s parametrem 70% jako maximální přípustná vzdálenost mezi klíčovými body v poměru k vzdálenostem maximálním, tedy obdobný způsob hledání řešení jako v předchozím filtru na rozostřené snímky 5.2.

Po aplikování filtru uživatel dostane výsledek v podobě vyfiltrovaného seznamu snímků, k vidění na obrázku 61z nichž si může libovolný snímek uložit.



Obrázek č. 61: Výsledek po aplikaci filtru na opakované snímky.

5.4 Implementace do aplikace

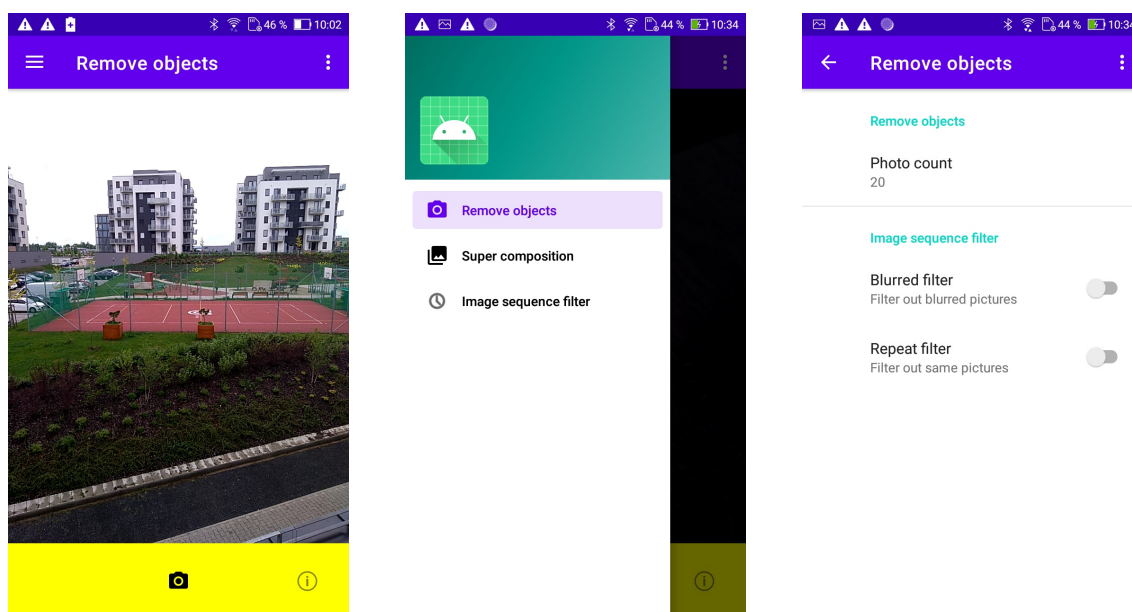
Jak bylo řečeno v úvodu, všechny tři představené módy jsou implementovány do jedné aplikace spustitelné na mobilních zařízeních s operačním systémem Android. Implementace algoritmů pro běh kamery jsou součástí samotné práce.

Samotná aplikace je napsána v jazyce Java, včetně uživatelského prostředí a rozhraní pro fotoaparát, zahrnuje do sebe skripty v Pythonu, které jsou pomocí pluginu Chaquopy [1] v Android studiu [2] spustitelné i v Android prostředí.

Aplikace kamery v Javě využívá balíček Camera2, umožňující přístup ke třídám, umožňující kontrolu kamery v telefonu. Aplikace je naprogramována na verzi Androidu 8.1 podporující minimální verzi Android API 27.

Jednotlivé módy mají vlastní záložky v uživatelském rozhraní a umožňují uživateli mezi nimi libovolně přepínat, viz obr. 62b.

Aplikace obsahuje i jednoduché nastavení parametrů pro jednotlivé módy, včetně zapnutí a vypnutí filtrů v rámci poslední funkcionality, takové nastavení je na obr. 62c.



(a) Hlavní obrazovka

(b) Menu aplikace

(c) Nastavení aplikace

Obrázek č. 62: Grafické rozhraní aplikace.

6 Závěr

V této práci byly ukázány různá praktická využití klíčových bodů ve snímku. Byly zde porovnány jednotlivé metody pro automatické nalezení těchto bodů. Jmenovitě se jednalo o SIFT,ORB,BRISK a AKAZE. Bylo ukázáno v jakých situacích jsou tyto metody vhodné k použití a jaké jsou jejich výhody a nevýhody v uvedených aplikacích.

Rovněž zde byly představeny principy homografie a využití transformační matice při transformování snímků z jedné perspektivní roviny do druhé. Na transformaci byly využity právě automaticky nalezené klíčové body.

V módu pro odstranění pohybujících se objektů, bylo potřeba nejdříve transformovat všechny snímky do roviny prvního snímku tak, aby se sjednotily souřadnice jednotlivých klíčových bodů napříč snímky. Při posuzování jednotlivých technik pro automatickou detekci klíčových bodů byl kladen důraz na rychlost jednotlivých technik, proto se jako nejlepší metoda podle měření časů jevila ORB. Poté se využilo algoritmu pro detekci pohybujících se snímků pomocí modelu gaussovských směrů, který vychází z pravděpodobnosti jasu pixelu pro konkrétní souřadnice napříč snímky. Pomocí získaných binárních masek pohybujících se objektů byly tyto objekty vyřiznuty a postupně nahrazeny záplatami sestávajících se z částí zbylých snímků. Pro kontrolu přesnosti toho filtru, byl vybrán pro stejnou scénu výstupní obrázek, který pohyb objektů neobsahoval a následně byl pomocí MSE porovnán se snímkem sním s odstraněnými daty.

V módu pro super kompozici se využívá homografie pro transformování snímků do roviny prvního snímku pomocí klíčových bodů detekovaných ve společné části snímku referenčního a transformovaného. V tomto případě se porovnávají techniky z hlediska přesnosti výsledné transformace vůči výstupní požadované transformaci. Jako nejlepší varianta podle porovnávací metriky MSE se jevila metoda AKAZE, která dosáhla nejlepší MSE skóre. Následně se snímky složí do jednoho obrázku. Takto se postupovalo, dokud uživatel nedosáhl požadované kompozice.

V posledním módu aplikace uživatel má možnost zaznamenat scénu v případě, že se schyluje k jednorázové události, ať už se jedná o sportovní okamžik nebo přírodní úkaz. Po stisknutí tlačítka pro začátek pořizování snímků, začne aplikace cyklicky pořizovat snímky, dokud neproběhne konkrétní událost a uživatel záznam ukončí. Mód následně uživateli odfiltruje rozmazané snímky a rovněž také snímky, které jsou podle nastaveného prahu velmi podobné či shodné. Uživatel si pak z vyfiltrovaného seznamu fotek vybere, která ho zajímá. Zde se jako metrika pro správné odstranění vymazaných snímků zvolila F1 skóre, jelikož skupina snímků byla předem vybrána a uměle rozmazána gaussovským filtrem s proměnou velikostí jádra. Podle F1 skóre byla následně vybrána za nejlepší techniku pro nalezení klíčových bodů BRISK. U filtru pro odstranění opakujících se fotek, byly jako odpovídající výstupní data vybrány snímky z pohledu uživatele. Následně stejně jako v prvním filtru jsou tyto data porovnána pomocí F1 skóre a metoda která, která získala nejlepší je ORB, která se tedy jeví jako nejlepší možný detektor pro podobné snímky.

Samostatně spustitelné skripty pro jednotlivé představené módy v aplikaci jsou dostupné ve veřejném repositáři na GitHubu na adrese https://github.com/vsafranek/Algoritmus_k_DP.

Zdroje

- [1] Chaquopy – Python SDK for Android. 302 Found [online][cit. 12.05.2022]. Dostupné z: <https://chaquo.com/chaquopy/>
- [2] Download Android Studio and SDK tools — Android Developers. Android Developers [online][cit. 12.05.2022]. Dostupné z: <https://developer.android.com/studio>
- [3] Retouch Remove Objects Editor - Apps on Google Play. [online][cit. 12.05.2022]. Dostupné z: <https://play.google.com/store/apps/details?id=com.magic.retouch&gl=NL>
- [4] Panorama 360 Camera: Virtual Tours: Instagram - Apps on Google Play. [online][cit. 15.05.2022]. Dostupné z: <https://play.google.com/store/apps/details?id=com.vtcreator.android360&gl=NL>
- [5] Google Street View - Apps on Google Play. [online][cit. 15.05.2022]. Dostupné z: <https://play.google.com/store/apps/details?id=com.google.android.street&gl=NL>
- [6] Photos To Video - Apps on Google Play. [online][cit. 15.05.2022]. Dostupné z: <https://play.google.com/store/apps/details?id=com.inglesdivino.imagestovideo&gl=NL>
- [7] Trilobit - Homografie a epipolární geometrie. Trilobit - Home Page [online][cit. 19.04.2022]. Dostupné z: <http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie>
- [8] OpenCV: Basic concepts of the homography explained with code. OpenCV documentation index [online][cit. 25.03.2022]. Dostupné z: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html
- [9] Introduction to ORB (Oriented FAST and Rotated BRIEF) — by Deepanshu Tyagi — Data Breach — Medium. Medium – Where good ideas find you. [online][cit. 04.02.2022]. Dostupné z: <https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf>
- [10] Introduction to ORB (Oriented FAST and Rotated BRIEF) — by Deepanshu Tyagi — Data Breach — Medium. Medium – Where good ideas find you. [online][cit. 04.02.2022]. Dostupné z: https://miro.medium.com/max/600/0*wGPpgnPImtwLb8NX.png
- [11] Introduction to ORB (Oriented FAST and Rotated BRIEF) — by Deepanshu Tyagi — Data Breach — Medium. Medium – Where good ideas find you. [online][cit. 04.02.2022]. Dostupné z: https://miro.medium.com/max/352/0*kpbHbTsvFR35oan
- [12] CHUAN, Luo. Overview of Image Matching Based on ORB Algorithm. Journal of Physics: Conference Series **1237**(032020), str. 1-4. [online][cit. 04.02.2022]. Dostupné z: <https://iopscience.iop.org/article/10.1088/1742-6596/1237/3/032020>

- [13] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In Computer Vision (ICCV), 2011 IEEE International Conference on, str. 2564–2571. IEEE, 2011.
- [14] OpenCV: ORB (Oriented FAST and Rotated BRIEF). OpenCV documentation index [online][cit. 04.02.2022]. Dostupné z: https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html
- [15] Medium – Where good ideas find you. [online][cit. 13.05.2022]. Dostupné z: <https://medium.com/@deepanshut041/introduction-to-brief-binary-robust-independent-elementary-features-436f4a31a0e6>
- [16] Home — Computer Science at UBC [online][cit. 13.05.2022]. Dostupné z: <https://www.cs.ubc.ca/lowe/papers/iccv99.pdf>
- [17] Scale-Invariant Feature Transform - 5KK73GPU2011. [online][cit. 25.03.2022]. Dostupné z: <https://sites.google.com/site/5kk73gpu2011/assignments/sift>
- [18] OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform). OpenCV documentation index [online][cit. 25.03.2022]. Dostupné z: https://docs.opencv.org/4.x/siftocal_extrema.jpg
- [19] OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform). OpenCV documentation index [online][cit. 25.03.2022]. Dostupné z: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html
- [20] HRŮZ, Marek. SIFT,SURF,MSER. [online]Plzeň: Katedra kybernetiky Fakulty aplikovaných věd Západočeské univerzity. 29. 10 2015 [cit. 25.03.2022].Dostupné z: <https://www.kky.zcu.cz/uploads/courses/mpv/04/materialy04.pdf>
- [21] Scale Invariant Feature Transform - Scholarpedia. [online][cit. 12.04.2022]. Dostupné z: http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform
- [22] OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform). OpenCV documentation index [online][cit. 25.03.2022]. Dostupné z: https://docs.opencv.org/4.x/sift_dog.jpg
- [23] Bag of Recurrence Patterns Representation for Time-Series Classification. ResearchGate — Find and share research [online][cit. 04.04.2022]. Dostupné z: <https://www.researchgate.net/profile/Nima-Hatami/publication/324104736/figure/fig3/AS:609726047977472@1522381426452/The-SIFT-descriptor-generation-First-the-gradient-magnitude-and-orientation-at-each.png>
- [24] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. Computer Vision–ECCV 2006, str. 404–417, 2006.

- [25] An Analysis of the SURF Method. ResearchGate — Find and share research [online][cit. 13.05.2022]. Dostupné z: https://www.researchgate.net/publication/282055016_An_Analysis_of_the_SURF_Method
- [26] Margarita Chli [online][cit. 13.05.2022]. Dostupné z: <https://margaritachli.com/papers/ICCV2011paper.pdf>
- [27] ResearchGate — Find and share research [online][cit. 13.05.2022]. Dostupné z: https://www.researchgate.net/publication/316712800/figure/fig1/AS962244854706200@1606428459690/The_scale_space_pyramid_in_BRISK_method-11.gif
- [28] The BRISK sampling pattern — Download Scientific Diagram. ResearchGate — Find and share research [online][cit. 07.04.2022]. Dostupné z: https://www.researchgate.net/profile/Roland-Siegwart/publication/221110715/figure/fig2/AS667632965525515@1536187511501/The_BRISK_sampling_pattern_with_N_60_points_the_small_blue_circles_denote_the_sampling.png
- [29] A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK — IEEE Conference Publication — IEEE Xplore. 301 Moved Permanently [online][cit. 10.04.2022]. Dostupné z: <https://ieeexplore.ieee.org/document/8346440>
- [30] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In Computer Vision—ECCV 2012, str. 214–227. 2012.
- [31] Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. Trans. Pattern Anal. Machine Intell, 2011.
- [32] Comparison between the Gaussian and nonlinear diffusion scale space for... — Download Scientific Diagram. ResearchGate — Find and share research [online][cit. 10.04.2022]. Dostupné z: https://www.researchgate.net/profile/Pablo-Fernandez-Alcantarilla/publication/236985005/figure/fig2/AS299529288077314@1448424755307/Comparison_between_the_Gaussian_and_nonlinear_diffusion_scale_space_for_several_evolution.png
- [33] X. Yang and K. T. Cheng. LDB: An ultra-fast feature for scalable augmented reality. In IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR), 2012.
- [34] Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. Trans. Pattern Anal. Machine Intell, 34(7) : 1281–1298, 2011.

- [35] Optical Flow in OpenCV — LearnOpenCV . AI Education For All, From Your First Steps to Mastery! [online][cit. 15.05.2022]. Dostupné z: <https://learnopencv.com/optical-flow-in-opencv/#optical-flow-theory>
- [36] Optical Flow in OpenCV (C++/Python) — LearnOpenCV . AI Education For All, From Your First Steps to Mastery! [online][cit. 15.05.2022]. Dostupné z: <https://learnopencv.com/wp-content/uploads/2020/08/object-displacement.png>
- [37] Two-Frame Motion Estimation Based on Polynomial Expansion. ResearchGate — Find and share research [online][cit. 15.05.2022]. Dostupné z: https://www.researchgate.net/publication/225138825_Two-Frame_Motion_Estimation_Based_on_Polynomial_Expansion
- [38] Using Background Subtraction Methods in Image Processing. Analytics India Magazine — Artificial Intelligence, Data Science, Machine Learning [online][cit. 16.04.2022]. Dostupné z: <https://analyticsindiamag.com/using-background-subtraction-methods-in-image-processing/>
- [39] OpenCV: How to Use Background Subtraction Methods. OpenCV documentation index [online][cit. 16.04.2022]. Dostupné z: <https://docs.opencv.org/4.x/BackgroundSubtractionTutorialScheme.png>
- [40] OpenCV: Morphological Transformations. OpenCV documentation index [online][cit. 18.04.2022]. Dostupné z: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
- [41] What is Lens Distortion? Barrel vs Pincushion Distortion. Aperlust - Travel Photography [online] [cit. 02.05.2022]. Dostupné z: <https://aperlust.com/barrel-vs-pincushion-lens-distortion/>
- [42] ŠAFRÁNEK, Vojtěch. Jasová korekce u skládání snímků. Plzeň: Západočeská univerzita v Plzni, 2022.
- [43] How to create a panorama image using OpenCV with Python. Master Data Science - Master Data Science [online][cit. 03.05.2022]. Dostupné z: <https://datahacker.rs/005-how-to-create-a-panorama-image-using-opencv-with-python/>
- [44] F-Score Definition — DeepAI. DeepAI: The front page of A.I. — DeepAI [online][cit. 08.05.2022]. Dostupné z: <https://deepai.org/machine-learning-glossary-and-terms/f-score>