



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Rozpoznávání a editace urbanistické scény
Student:	Jan Šefčík
Vedoucí:	Ing. Radek Richtr, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Úkolem je vytvořit nástroj umožňující na základě obrazových a lokálních geolokačních (akcelerometr, gyroskop atp.) dat rozpoznat urbanistickou scénu a určit přesnou pozici uživatele. Do rozpoznávaných scén pak bude umístěna historická podoba vybraných budov, primárně pak 3D modelů budov projektu VMČK.

- 1) Proveďte rešerši dosavadních možností k rozpoznávání obsahu urbanistických scén s ohledem na primární cíl projektu.
- 2) Vyhledané možnosti analyzujte s ohledem na omezení reprezentované projektem VMČK (např. již neexistující budovy).
- 3) Navrhněte a implementujte prototyp vhodný pro určování přesné polohy jak na základě jen historických dat, tak lokálních geolokačních dat.
- 4) Výsledné řešení důkladně otestujte.
- 5) Debatujte vhodnost a omezení prototypu pro projekt Věnných měst českých královen.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 11. února 2020



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Rozpoznávání a editace urbanistické scény

Jan Šefčík

Katedra softwarového inženýrství

Vedoucí práce: Ing. Radek Richtr, Ph.D.

2. června 2020

Poděkování

Chtěl bych poděkovat Ing. Radku Richtrovi, Ph.D., vedoucímu mé práce, za konzultace a odborné vedení v průběhu tvorby bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel dohodu, na základě níž se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 2. června 2020

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 Jan Šefčík. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Šefčík, Jan. *Rozpoznávání a editace urbanistické scény*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Rozpoznávání obrazu a jeho klasifikace jsou velmi komplexní problémy v rámci oboru strojového vidění. Tato práce se zabývá návrhem a implementací skriptu na rozpoznávání budov v urbanistických scénách za použití geolokačních a obrazových dat.

Pozice uživatele je získána z metadat pořízeného snímku a pro následnou klasifikaci obrazu jsou použity lokální příznaky (SIFT). Součástí práce je i analýza lokálních příznaků a jejich vhodnosti pro dané použití. V závěru je přesnost metody otestována vzhledem k porovnání různých postupů předzpracování obrazu. Přínosem této práce je řešení na míru a rozsáhlá analýza pro projekt Věnná města českých královen. Tato práce bude tvořit základ pro navazující bakalářské práce.

Klíčová slova prototyp nástroje pro rozpoznávání budov, SIFT, SURF, ORB, porovnávání příznaků, urbanistické scény, lokální příznaky, rozpoznávání v obraze, klasifikace obrazu

Abstract

Image recognition and image classification are very complex problems in computer vision. This thesis deals with the design and the implementation of an urban scene recognition script using geological and image data.

GPS coordinates of the image are used to reduce the number of buildings that are later recognized. GPS coordinates are obtained from the input image. Later, SIFT local features are used for image classification. Chapter of the thesis is the analysis of local features and their suitability for this task. The conclusion focuses on the testing the accuracy of each implementation and comparison of different methods of image preprocessing. Relevance of this thesis is a specific solution and extensive analysis for the project called Věnná města českých královen. This thesis is a baseline for a future development that goes beyond a bachelor thesis.

Keywords prototype of the tool for urban scene recognition, SIFT, SURF, ORB, feature matching, urban scenes, local features, image recognition, image classification

Obsah

Úvod	1
Cíl práce	3
1 Rešerše	5
1.1 Předzpracování obrazu	5
1.1.1 Přehled vybraných metod	5
1.2 Lokální příznaky	8
1.2.1 Detekce	8
1.2.2 Párování a spojování příznaků	10
1.3 Algoritmy pro výpočet lokálních příznaků	12
1.3.1 LBP	12
1.3.2 HOG	13
1.3.3 Haralickovy příznaky	16
1.3.4 ORB	19
1.3.5 SIFT	22
1.3.6 SURF	25
1.4 Metody párování příznaků	29
1.4.1 Brute force matcher	29
1.4.2 FLANN based matcher	29
1.5 Metody filtrování příznaků	30
1.5.1 Loweho poměrový test	30
1.5.2 Cross-check test	30
1.5.3 RANSAC	31
2 Analýza	33
2.1 Výběr vhodného typu příznaků	33
2.1.1 Podrobné porovnání vlastností	33
2.1.2 Párování příznaků	35

2.2	Porovnání postupů pro předzpracování obrazu	35
2.3	Výběr nejlepších bodů	38
2.4	Zrychlení vyhledávání	39
3	Návrh	41
3.1	Funkční a nefunkční požadavky	41
3.2	Případy užití	42
3.3	Diagram komponent	43
3.4	Databáze budov	44
3.5	Předzpracování obrazu	44
3.6	Rozpoznání budovy	44
3.7	Vložení snímku/modelu do reality	45
3.8	Výsledná aplikace	45
4	Realizace	47
4.1	Technologie	47
4.2	Konfigurační soubor	48
4.3	Tvorba testovacího datasetu	48
4.4	Databáze budov	49
4.5	Úpravu vstupního snímku	49
4.6	Lokalizační data	50
4.7	Extrakce a porovnání příznaků	50
4.8	Výběr nejlepší shody	50
4.9	Transformace budovy	51
4.10	Skládání obrazů	52
5	Testování	53
5.1	Technické limitace prototypu	53
5.2	Testy parametrů	54
5.2.1	Předzpracování obrazu	55
5.2.2	Výběr dobrých spojení	56
5.2.3	Selekce klíčových bodů	57
	Závěr	59
	Bibliografie	61
	A Seznam použitých zkratk	65
	B Obsah příloženého CD	67
	C Seznam budov	69
	D Výsledky testování	73

Seznam obrázků

1.1	Histogram obrazu před a po jeho ekvalizaci	7
1.2	Demonstrace detekce a párování příznaků	9
1.3	Za předpokladu malého okolí jsou rohy v obraze dostatečně unikátní a dobře lokalizovatelné. Modrý čtverec popisuje okolí bodu.	9
1.4	Ilustrace opakovatelnosti. Horní řádek: fotografie panoramatu s vyznačenými detekovanými okny. Spodní řádek: detaily příslušných oblastí z fotografií	10
1.5	Příklady různých invariantů	11
1.6	Ukázka párování příznaků	11
1.7	Konstrukce invariance LBP vůči rotaci	13
1.8	Histogramy orientací gradientů pro různé hodnoty	14
1.9	Ukázka oříznutí histogramu	15
1.10	Princip konstrukce HOG deskriptoru	15
1.11	Tvorba jedné GLCM matice. Vpravo vstupní obraz, vlevo GLCM matice	16
1.12	Každá úroveň pyramidy obsahuje zmenšenou verzi předchozího	20
1.13	Hledání klíčového bodu pomocí metody FAST	20
1.14	Ukázka výběru dvou pixelů v okolí	21
1.15	Ukázka jednotlivých oktáv	23
1.16	Tvorba DOG	24
1.17	Vrcholy nad 80% jsou započítány do směru bodu	25
1.18	Tvorba vektoru příznaku	25
1.19	Gaussovy částečné derivace v x y a y	27
1.20	Pyramidové schéma pro filtry	28
1.21	Ukázka hledání orientace deskriptoru příznaku	28
1.22	Tvorba deskriptoru. Každý region je rozdělen na 5×5 vzorků	29

1.23	Pravděpodobnost, že dojde ke správnému spojení příznaků za použití sousední vzdálenosti. Na horizontální ose je popsán poměr vzdáleností mezi nejbližší a druhou nejbližší vzdáleností. Na vertikální ose je popsána funkce hustoty pravděpodobnosti (ang. Probability Density Function). Plná čára značí správné spojení a čerchovaná ty nesprávné	31
2.1	Ukázka chybného párování	35
2.2	Ukázka ekvalizace a adaptivní ekvalizace histogramu na dvou snímcích. Zleva: originální snímek, ekvalizace histogramu a adaptivní ekvalizace histogramu	36
2.3	Ukázka sto nejlepších spojení na snímku. Horní: bez úpravy, Spodní: s oříznutou budovou s černým pozadím a neupraveným obrázkem	38
2.4	Mnoho příznaků je také nalezeno na plakátech a značkách	38
2.5	Výběr bodů, které jsou rovnoměrněji rozloženy dosahuje vyšší přesnosti. Horní: nejlepší čtyři body, Spodní: rovnoměrněji rozložené body	39
3.1	Use case diagram základních cílů	43
3.2	Diagram komponent	43
3.3	Konceptuální model části databáze	44
3.4	Wireframe výsledné aplikace	46
4.1	Diagram aktivit	47
4.2	Složková struktura prototypu	49
4.3	Výsledné umístění rozpoznané budovy do scény	51
4.4	Ukázka postupu skládání obrazu	52
5.1	Ukázky hodnocení přesnosti. Zleva 1, 2, 3, 4	54
5.2	Výsledky nejúspěšnějšího nastavení parametrů: CLAHE, poměrový test na 0,6 a RANSAC. Budovy na obrázku ze shora (pravý, levý): (1, 8), (4, 9), (5, 10), (7,11), 12	55
D.1	Výsledné umístění budov bez úprav histogramu (pravý, levý): (1, 8), (7, 10), 12	74
D.2	Výsledné umístění budov s ekvalizací histogramu: (pravý, levý): (1, 8), 7	74
D.3	Výsledné umístění budov s adaptivní ekvalizací histogramu (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8	75
D.4	Výsledné umístění budov pro poměrový test s hodnotou prahu 0,4. Budovy na obrázku ze shora: 1, 7, 8, 10	76
D.5	Výsledné umístění budov pro poměrový test s hodnotou prahu 0,5. Budovy na obrázku ze shora (pravý, levý): (1, 8), (4, 10), (5, 12), 7	77

D.6	Výsledné umístění budov pro poměrový test s hodnotou prahu 0,6. Budovy na obrázku ze shora (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8	78
D.7	Výsledné umístění budov pro poměrový test s hodnotou prahu 0,7. Budovy na obrázku ze shora (pravý, levý): (1, 10), (8, 12)	78
D.8	Výsledné umístění budov pro poměrový test s hodnotou prahu 0,8. Budovy na obrázku ze shora (pravý, levý): (1, 10), (7, 12), 8	79
D.9	Výsledné umístění budov pro čtyři nejlepší klíčové body. Budovy na obrázku ze shora (pravý, levý): (1, 8), (5, 10), (7, 11), 12	79
D.10	Výsledné umístění budov s minimální vzdáleností 80 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (1, 8), (5, 10), (7, 11)	80
D.11	Výsledné umístění budov s minimální vzdáleností 100 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8	81
D.12	Výsledné umístění budov s minimální vzdáleností 120 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (7, 10), (8, 12)	81

Seznam tabulek

2.1	Výsledky porovnání se měnící se intenzitou	34
2.2	Výsledky porovnání v otočeném obraze	34
2.3	Úspěšnost párování vůči rotaci obrazu o daný úhel	34
2.4	Výsledky porovnání se zvětšeným obrazem	35
2.5	Výsledky porovnání se měnící se intenzitou	40
5.1	Výsledky nejúspěšnějšího nastavení parametrů: CLAHE, poměrový test na 0,6 a RANSAC	54
D.1	Výsledky testu bez úpravy histogramu	73
D.2	Výsledky testu s úpravou histogramu	73
D.3	Výsledky testu s adaptivní úpravou histogramu	73
D.4	Výsledky testu s nastaveným poměrovým testem na 0,4	74
D.5	Výsledky testu s nastaveným poměrovým testem na 0,5	75
D.6	Výsledky testu s nastaveným poměrovým testem na 0,6	75
D.7	Výsledky testu s nastaveným poměrovým testem na 0,7	76
D.8	Výsledky testu s nastaveným poměrovým testem na 0,8	76
D.9	Výsledky testu s výběrem nejlepších bodů čistě podle vzdálenosti .	76
D.10	Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a mi- nimální vzdáleností, kde práh je nastaven na 80 pixelů	76
D.11	Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a mi- nimální vzdáleností, kde práh je nastaven na 100 pixelů	77
D.12	Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a mi- nimální vzdáleností, kde práh je nastaven na 120 pixelů	77

Seznam výpisů kódu

4.1	Meta data budovy ve formátu JSON	49
-----	--	----

Úvod

Strojové vidění je jedním z novějších, pro mne velmi zajímavých, oborů informačních technologií. Tento obor se zabývá nahrazením komplexního lidského zrakového aparátu tak, aby i stroje/počítače mohly vnímat obraz a video stejně jako člověk. Do nedávné doby byla tato schopnost strojů značně omezená. Především díky umělé inteligenci, hlubokému učení a neuronovým sítím zažil tento obor prudký rozvoj. V některých případech rozpoznávání a kategorizace objektů dokonce stroje dosahují lepších výsledků než lidé. Cílem strojového vidění je vytvoření umělého mozku, který bude zpracovávat obrazová data stejně jako ten lidský.

Tato práce se zaměřuje zejména na problém klasifikace, neboli nalezení specifických znaků pro jednotlivé druhy objektů, které mají být rozpoznány. Lidé na první pohled poznají, že na obrázku je například zachycené malé koťátko, ale pro počítač to není tak jednoduché a samozřejmé. Obraz je vnímán jen jako velká matice s čísly, které představují jednotlivé hodnoty pixelů / barevných kanálů.

Pro popis charakteristických rysů objektu se používají tzv. příznaky, které popisují nějakou vlastnost objektu na snímku, např. barvu, tvar nebo pozici. Příznaky mají různé úrovně komplexnosti. Výše uvedené příznaky mají velmi nízkou úroveň komplexnosti, ale může se jednat i o jejich kombinace. Například šedý trojúhelník s bílým vrcholem by mohl popisovat vrcholky hor - takový příznak má pak vyšší úroveň komplexnosti.

V rámci analytické části práce jsou testovány nejpoužívanější algoritmy pro extrahování příznaků z obrazu a vhodnost jejich použití pro rozpoznávání budov. Před samotným získáváním příznaků probíhají úpravy (transformace) vstupního snímku v rámci jeho předzpracování. Mezi tyto úpravy patří metody od změn jasu a kontrastu až po perspektivní zkreslení. Vybrané metody jsou otestovány a na základě výsledku je navrženo nejlepší možné řešení. Pro implementaci tohoto prototypu nástroje je použitý programovací jazyk Python a vybrané knihovny pro práci s obrazem, například OpenCV.

Práce je součástí dlouhodobého projektu Věnná města českých královen ve spolupráci z historiky z Hradce Králové se studenty/pedagogy z Fakulty informačních technologií v Praze a Historického ústavu Akademie věd České republiky. Výslednou částí projektu bude mobilní aplikace, která umožní uživateli pomocí rozšířené reality procházet věnná města ve středověké/historické době. Místo novodobých nebo renovovaných staveb budou do prostoru umístěny historicky přesné 3D modely budov. Práce se zabývá problémem, který je potřeba prozkoumat a vyřešit na cestě k takové aplikaci.

Práci jsem si vybral na základě její zajímavosti a přesahu do různých odvětví. Dalším faktorem byla možnost dále práci rozvíjet až do finální verze. Znalosti načerpané v rámci této práce mají využití v budoucnosti.

Cíl práce

Cílem této bakalářské práce je vytvoření prototypu nástroje, který umožní v rámci projektu Věnných měst českých královen rozpoznávat budovy na snímku a lokalizovat pozici uživatele na základě obrazových a geolokačních dat. Do rozpoznávaných scén bude poté umístěna historická podoba vybraných budov.

Prvním cílem je seznámení se s možnostmi rozpoznávání obrazu urbanistických (venkovních) scén. Druhým, následujícím cílem, je analýza vyhledaných možností a omezení v rámci projektu Věnná města českých královen. Třetím cílem je návrh a implementace prototypu vhodného pro určování polohy na základě lokálních geolokačních a obrazových dat. Čtvrtým cílem je otestování výsledného řešení a porovnání výsledků vůči omezení daného projektu.

Výsledky této práce budou přínosné pro projekt Věnných měst českých královen, na kterém spolupracují studenti i pedagogové z Fakulty informačních technologií na ČVUT V Praze s historiky z Hradce Králové a Historického ústavu Akademie věd České republiky, v. v. i.

Rešerše

V této kapitole jsou popsány a představeny vybrané lokální příznaky, např. LBP, ORB a SIFT. Prostor je také věnován předzpracování obrazu a porovnávání/párování jednotlivých příznaků.

1.1 Předzpracování obrazu

Předzpracování [1] upraví obraz do stavu, který je nejvhodnější pro konkrétní použití. Jeho úkolem je vyrovnat podmínky a zbavit se nechtěných vlivů během pořizování dat. Každý snímek je pořízen za jiných podmínek od slunečného dne až po zataženou oblohu či déšť. Nezáleží pouze na počasí, ale i na technických parametrech kamery.

Strojové vidění se z velké části věnuje návrhu nejlepšího prostředí během získávání obrazu. Ne vždy je však možné používat nebo vytvořit takové prostředí, kde by podmínky byly konstantní.

Obecně není známé žádné pořadí operací, které zaručí nejlepší výsledky. Všechny volby metod a parametrů se opírají o jejich otestování a porovnání výsledků. Více o konkrétním návrhu postupu je napsáno v kapitole Analýza.

1.1.1 Přehled vybraných metod

Notace

Celý obraz I je matice o rozměrech $x \times y \times s$, kde x a y značí souřadnice pixelu a s barevný kanál. Výstupní transformovaný obraz je značen I' . Obraz I_r , kde $r = [x, y, s]$ značí konkrétní barevnou složku s pixelu na pozici x, y jako i_r , kde $r = [x, y, \bullet]$ značí všechny spektrální hodnoty na pozici x, y , tedy celý RGB pixel. Obraz I_R , kde $R = [\bullet, \bullet, 0]$ značí celý červený kanál. Kanály G a B mají pouze jinou hodnotu s .

Převod snímku do odstínů šedi

Jedná se o základní krok téměř u všech algoritmů na detekci hran, které velmi úzce souvisí s výpočtem příznaků. Existují dva způsoby převodu obrazu: průměrovací metoda a metoda váhování. V průměrovací metodě každá složka RGB přispívá k výsledné hodnotě právě jednou třetinou:

$$I' = \frac{I_R + I_G + I_B}{3} . \quad (1.1)$$

Ve skutečnosti však barvy nepřispívají stejným dílem jasové informace k výsledku [2]. Proto existuje metoda váhování, která přidělí každé složce svoji váhu:

$$I' = 0,3I_R + 0,59I_G + 0,11I_B . \quad (1.2)$$

Ta vychází z biologické stavby lidského oka. Přesněji, podle poměru čípků, které jsou citlivé na jiné barvy (světelné délky).

Úpravy jasu a kontrastu

Mezi metody pro úpravu kontrastu patří i níže popsaná gama korekce a ekvalizace histogramu. Tyto metody patří k nelineárním úpravám. Při úpravách jasu a kontrastu je důležité myslet na to, že ne všechny metody jsou invertibilní a je možné ztratit informaci v obraze [3]. Tato ztráta se nazývá clipping.

$$I' = \alpha \cdot I + \beta , \quad (1.3)$$

kde parametry $\alpha > 0$ a β představují kontrast a jas. Postupným přičítáním konstanty mohou světlá místa (s vysokými hodnotami) překročit hranici 255, kdy dochází ke ztrátě informací o kresbě snímku.

Gama korekce

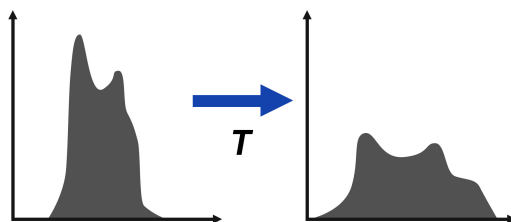
Jedná se o nelineární operaci, která se používá k zakódování a dekodování jasových hodnot [4]. Gama korekce pracuje s jasovými hodnotami tak, jak jsou světelné podmínky vnímány lidským okem. Gama korekce se snaží dát více prostoru stínům a světlům.

$$I' = AI^\gamma \quad (1.4)$$

Nejčastěji se používá $A = 1$. Vstupní a výstupní hodnoty jsou v rozsahu 0-1. Hodnota $\gamma < 1$ se nazývá kódovací gama a v opačném případě ($\gamma > 1$) se nazývá dekodovací gama.

Ekvalizace histogramu

Tato metoda upravuje kontrast obrazu. Ekvalizace se používá pro vytažení kresby ve snímku, např. z přeexponovaného snímku vytvoří tmavší kontrastnější obrázek, kde zvýrazní části kresby, které předtím splývaly. Pokud je obrázek podexponovaný, tak histogram tohoto snímku nemusí využívat jeho plochu, zejména šířku naplno. Ekvalizace se snaží tzv. narovnat histogram tak, aby byl vyvážený a počty jasových složek rovnoměrně rozmístěné, viz Obrázek 1.1.



Obrázek 1.1: Histogram obrazu před a po jeho ekvalizaci

V prostřední části by měl být histogram lehce „vyvýšený“. Velkou výhodou této funkce je, že pokud je známá funkce ekvalizace histogramu [5], tak je možné obnovit původní obrázek. Tato metoda má ale i jednu nevýhodu, může zvýrazňovat šum místo užitečných signálů.

Normalizace

Jedná se o proces, který mění rozsah intenzity pixelů. Používá se pro obrazy s nízkým kontrastem. Histogram takového snímku je velmi „úzký“ a díky přemapování jeho intervalu je možné histogram tzv. rozšířit. Níže uvedený vzorec popisuje normalizaci pouze jednoho kanálu, v případě barevného snímku je nutné normalizovat všechny barevné kanály [6].

$$I' = 255 \times (I - I_{\min}) / (I_{\max} - I_{\min}) , \quad (1.5)$$

kde I_{\max} a I_{\min} jsou nejvyšší a nejnižší hodnotu pro daný kanál snímku.

Perspektivní transformace

Provádí transformaci z 3D světa do 2D obrázku [7]. Používá se pro narovnání, neboli převedení dané části obrazu na obraz, který není zkreslený perspektivou. Příkladem může být skenování dokumentů, kdy je pod nějakým úhlem zachycen snímek s dokumentem a poté je sestavena transformační matice,

kteřá převede snímek a zbaví ho zkreslení. Ukázka transformace jednoho pixelu, viz 1.6.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1.6)$$

Ořez

Ořez je klasický postup, který se dá jednoduše zbavit rušivých a nedůležitých elementů v obraze. Po oříznutí bude objekt zájmu zabírat větší plochu obrazu. Velmi známým problémem je práce s okraji obrázku. Existují různé metody na protažení nebo zrcadlení snímku. Žádná z nich však není ideální. Pokud se předpokládá, že žádný z pixelů na okraji obrazu neobsahuje důležitou informaci, je možné obrázek oříznout a problém rohů a krajů zcela ignorovat.

1.2 Lokální příznaky

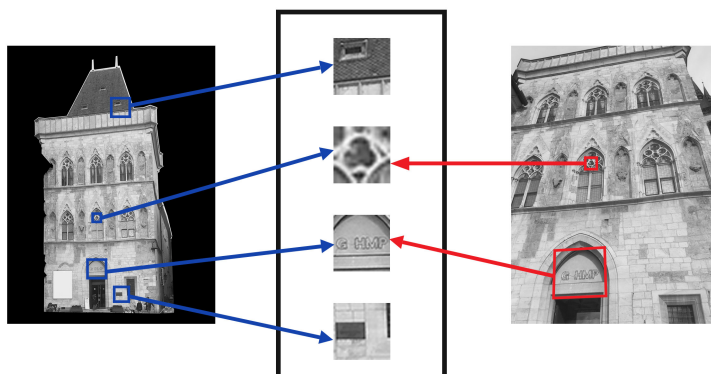
Rozpoznávání objektů netvoří člověku téměř žádný problém. Stačí krátký pohled a mozek automaticky přiřadí název/kategorii ke sledovanému objektu. Pro počítač však tento úkol není tak jednoduchý, ten vnímá obraz jen jako velký počet pixelů. Postrádá (naši) lidskou znalost, že tyto pixely mají spolu nějaký vztah, např. černé pixely, které jsou uspořádané do kruhu, tvoří kolo od auta. Právě tyto vazby mezi body v obraze jsou popisovány (kontextovými) příznaky [8].

Příznaky mohou charakterizovat vlastnost obrazu. Nejdříve je potřeba detekovat body zájmu, které budou poté popsány pomocí příznaků. Ze získaných příznaků jsou vybrány vhodné příznaky [9]. Nakonec je možné tyto příznaky, z různých obrazů, mezi sebou porovnávat a na základě podobnosti je párovat.

Více příznaků tvoří tzv. vektor příznaků, který je určen k reprezentaci zvolených vlastností obrazových dat. Pomocí něj je možné párovat jednotlivé příznaky (z různých obrazů) mezi sebou. Dělení příznaků na užitečné a zbytečné vždy záleží až na konkrétních problémech.

1.2.1 Detekce

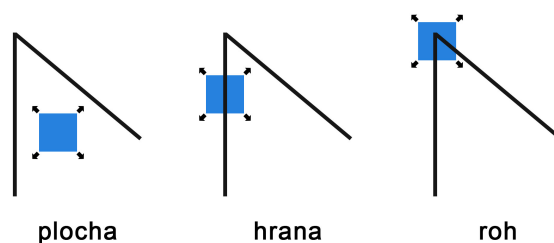
Proces detekce začíná hledáním potencionálních bodů zájmu, oblastí nebo rovnou celých dat [10]. Aby se například z bodu a jeho okolí mohl stát příznak, tak musí splňovat níže vypsane vlastnosti. Tento postup je aplikovaný na oba snímky, které mají být porovnané. Poté probíhá párování a hledají se shody mezi jednotlivými příznaky. Tento proces párování je popsán v následujících kapitolách. Obrázek 1.2 ilustruje ukázkou celého procesu rozpoznání.



Obrázek 1.2: Demonstrace detekce a párování příznaků

Unikátnost

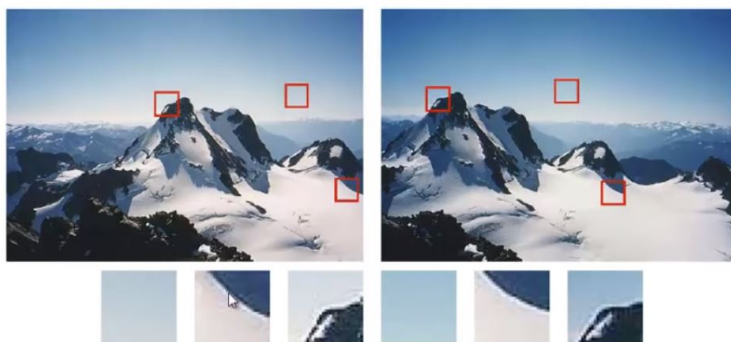
Okolí kolem bodu v obraze musí být dostatečně unikátní, aby ho bylo možné spolehlivě porovnávat s ostatními příznaky. Při použití malého okolí jsou za unikátní považovány například rohy, protože na přímce, či ploše není možné určit přesnou pozici. V případě přímky je možné s bodem posouvat po přímce a v rámci plochy může být bod posouván všemi směry, viz Obrázek 1.3.



Obrázek 1.3: Za předpokladu malého okolí jsou rohy v obraze dostatečně unikátní a dobře lokalizovatelné. Modrý čtverec popisuje okolí bodu.

Opakovaná detekovatelnost

Daný příznak bude počítán pro více obrázků, proto by měl být příznak dostatečně výrazný tak, aby mohl být detekován na každém snímku, kde se nachází. V případě deskriptoru níže, viz Obrázek 1.4, je použit stín, který nemůže být opakovaně detekovatelný, pokud by byl snímek pořízen v jiný čas. V případě pouhého spojení těchto fotek do jedné (panorama) je takový příznak dostatečný.



Obrázek 1.4: Ilustrace opakovatelnosti. Horní řádek: fotografie panoramatu s vyznačenými detekovanými okny. Spodní řádek: detaily příslušných oblastí z fotografií

Lokalizovatelnost

Příznak má jedinečnou pozici v obraze. Jakékoliv změny v podmínkách zobrazení, např. šum, pozice kamery, nemají žádný vliv na jeho umístění. Neboli není možné ho zaměnit za jiný příznak na stejném snímku. Podle Obrázku 1.3, kde v případě hrany může být získán ten stejný příznak na pozicích, které leží na stejné svislé hraně.

Deskriptor příznaku

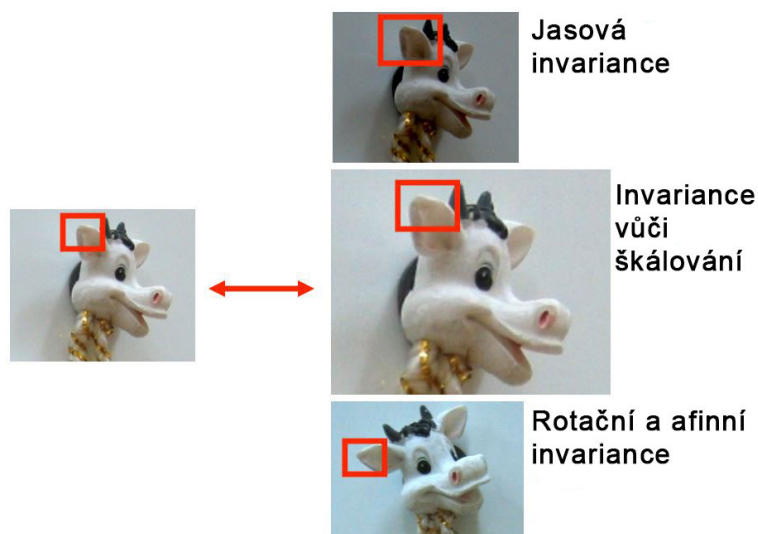
Deskriptor příznaků je algoritmus, který kóduje vlastnosti příznaku pomocí čísel. Algoritmus převádí vlastnosti příznaku do číselné řady, která by měla být co nejvíce originální, např. jako otisk prstu. Výstupem deskriptoru je vektor [8, 11] zakódovaných příznaků (ang. feature vector). Deskriptor by měl poskytovat určitou invarianci ke geometrickým a fotometrickým rozdílům mezi snímky. Často se pro lokalitu příznaku nepoužívají jen jeho souřadnice (x, y) , ale důležitá je také jeho velikost $(x, y, scale)$

Invariance

Pokud je příznak invariantní, tak to znamená, že je příznak neměnný vůči různým operacím/transformacím [12, 13], např. rotaci, zkosení, škálování, změny jasů. Pokud je příznak invariantní, ztrácí tím svoji mohutnost, neboli transformované příznaky se mohou slít v jeden.

1.2.2 Párování a spojování příznaků

Obecné párování/spojování příznaků (ang. feature matching) je založeno na porovnání příznaku se všemi ostatními. Porovnání funguje na základě



Obrázek 1.5: Příklady různých invariantů

vzdálenostní funkce, která vrací vzdálenosti, neboli odlišnosti mezi příznaky. Spojen je ten příznak, který má nejmenší vzdálenost. Typ vzdálenosti se liší podle volby deskriptoru.



Obrázek 1.6: Ukázka párování příznaků

Velké množství příznaků může být získáno ze snímků efektivními algoritmy. Příznaky jsou velmi odlišné, což dovoluje právě jednomu příznaku být s vysokou pravděpodobností správně přiřazen, a to i přes velkou skupinu příznaků. Důležité je rozdělení příznaků na dobré a špatné.

Klasickým a nejjednodušším způsobem je použití prahu [14] definujícího hranici maximální vzdálenosti, která rozlišuje příznaky na dobré a špatné. Dalším důležitým úkolem je nalezení dané hranice, která propustí pouze ty dobré, pro dané použití užitečné příznaky a ty budou dále spojeny.

1.3 Algoritmy pro výpočet lokálních příznaků

Jedním z nejjednodušších příznaků je LBP, který se používá zejména při práci s texturami. Haralickovy příznaky obsahují informaci o charakteristikách vzoru, např. homogenitě, jasů, lineární závislosti, kontrastu. Další popsanou metodou je SIFT, která se inspirovala histogramem orientovaných gradientů. Níže popsané metody mají hodně podobného a navzájem ze sebe často vycházejí.

1.3.1 LBP

Metoda LBP [15] (z ang. Local Binary Patterns) byla publikována v roce 1994. Její velkou výhodou je její jednoduchost výpočtu. Dosahuje dobrých výsledků i při velikosti vzorového obrazu 16×16 pixelů. Tato vlastnost je velmi užitečná zejména pro klasifikaci textur nebo segmentaci.

Výpočet LBP příznaků funguje na principu porovnávání hodnot okolních pixelů kolem středového pixelu. Porovnávají se hodnoty obrazu po převedení do jednoho spektra, např. ve stupních šedi. Pokud je hodnota středního pixelu větší než porovnávaného pixelu, tak se na stejné místo (do nové matice 3×3) uloží 0.

Dobrý příznak/deskriptor by měl být odolný vůči různým změnám v obraze. LBP je invariantní vůči jasovým transformacím a je možné odvodit i rotační invariant. Operátor $LBP_{P,R}$, který charakterizuje prostorové uspořádání textury:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad (1.7)$$

kde R znamená poloměr kružnice se středem ve zkoumaném bodě, na kterém je rozmístěno P okolních pixelů, g_c je hodnota středového pixelu a g_p značí hodnotu sousedního pixelu.

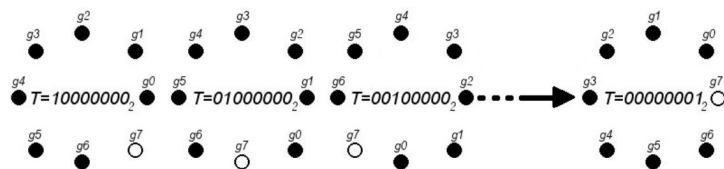
Pro zvýšení robustnosti LBP metody se její deskriptor skládá jen z uniformních vzorů v obraze, které během testování prokázaly největší stabilitu. Vzory tvoří šablonu pro detekci bodů, hran a konstantních ploch v obraze. Uniformita vzorů udává počet přechodu mezi 1 a 0 v daném řetězci.

$$LBP_{P,R}^i = \min [rot(LBP_{P,R,i}) | i = 0, 1, \dots, P-1], \quad (1.8)$$

kde $rot(x, i)$ provádí kruhový bitový posun (doprava) na P -bitovém čísle x , i -krát. Operátor $LBP_{P,R}^i$ kvantifikuje výskyt jednotlivých rotačně invariantních vzorů. Tyto vzory lze považovat za detektor příznaků.

Například, vzor 1111111_2 má $U=0$ a vzor 11100111_2 má $U=2$. Vzory s mírou uniformnosti o hodnotě maximálně dva jsou považovány za uniformní. Vzor 11111100 je možné orotovat do podoby 00111111 , neboli v desítkové sou-

stavě číslo 252 rotuje na číslo 63, viz Obrázek 1.7. Obvykle se pro reprezentaci používají vzory, které mají nejnižší hodnotu.



Obrázek 1.7: Konstrukce invariance LBP vůči rotaci

Deskriptor

Deskriptor příznaku LBP je sestaven[16] tak, že obraz na vstupu je rovnoměrně rozdělen na čtvercové podoblasti. Velikost oblasti záleží na velikosti vstupního obrazu i zvolených parametřů lokálního okolí P a R . Pro každou oblast jsou získány binární vzory textur pro všechny její pixely, $T_{P,R}$ a rotačně nezávislý tvar těchto vzorů $T_{P,R}^{riu0}$, kde horní index $riu0$ značí invarianci vůči rotaci a míru uniformnosti $u = 0$. Z množiny získaných vzorů jsou poté vybrány jen ty, které splňují podmínku uniformnosti ($u = 2$). Poté jsou spočteny jednotlivé operátory $LBP_{P,R}^{riu2}$. Pro každou podoblast je tak získán určitý počet operátorů, které jsou zaneseny do histogramů dané podoblasti. Histogramy tvoří jednotlivé prvky výsledného LBP příznaku.

1.3.2 HOG

Metoda HOG [17] (z ang. Histogram of Oriented Gradients) byla vyvinuta v roce 1986 zejména pro detekování postav v obraze. Původní myšlenka je použití distribuce gradientů pro popis objektů. Vstupní obrázky nemusí být jen v odstínech šedi, HOG funguje i pro barevné snímky, ty však musí být nejdříve normalizovány pomocí gama korekce, aby byla metoda robustnější.

Výpočet gradientu

Proces výpočtu gradientu má velký vliv na výsledný deskriptor. V rámci prvního kroku je použita konvoluce Gaussovým jádrem s různými hodnotami parametru σ (včetně $\sigma = 0$) a následně jsou použity tyto 2D masky pro výpočet diskrétního gradientu:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \quad (1.9)$$

HOG získává gradienty pomocí konvoluce Gaussovým filtrem obrazu I při $\sigma = 0$ s maskou $[-1, 0, 1]$:

$$I_x = I * [-1, 0, 1] , \quad (1.10)$$

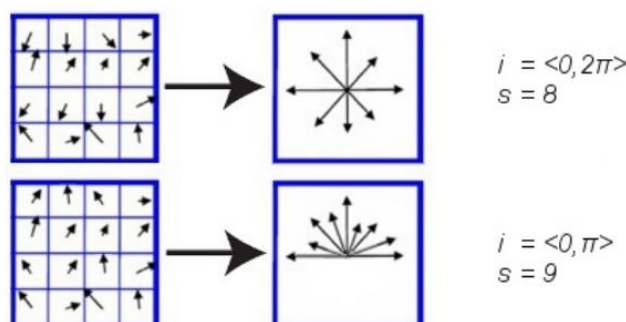
$$I_y = I * [-1, 0, 1]^T . \quad (1.11)$$

Po získání obrazů I_x a I_y , je pro každý pixel vypočten směr $m(x, y)$ a velikost gradientu (x, y) :

$$m(x, y) = \sqrt{I_x^2 + I_y^2} , \quad (1.12)$$

$$\Theta(x, y) = \tan^{-1} \left(\frac{I_y}{I_x} \right) . \quad (1.13)$$

Následně je sestaven histogram orientací, jehož počet složek je určen sektorem s a intervalem i . Interval i udává rozsah, ve kterém je zjišťován směr gradientu (x, y) . Sektor s definuje počet výsečí, na které je interval i rozdělen, viz Obrázek 1.8. Přínosy jednotlivých složek jsou váženy jejich velikostmi $m(x, y)$, a navíc bilineárně rozděleny i mezi sousedy, díky čemu je deskriptor více robustní.



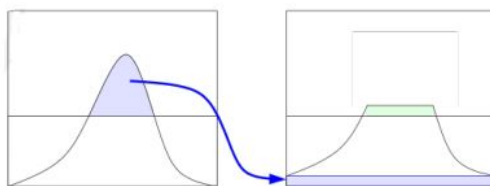
Obrázek 1.8: Histogramy orientací gradientů pro různé hodnoty

Z důvodu zvýšení robustnosti příznaku vůči jasovým transformacím snímku jsou získané histogramy normalizovány. Mezi nejpoužívanější normalizační schémata metody HOG patří:

$$L^1 - \text{norm} = \frac{v}{(\|v\|_1 + e)} , \quad (1.14)$$

$$L^2 - \text{norm} = \frac{v}{(\|v\|_2^2 + e^2)} . \quad (1.15)$$

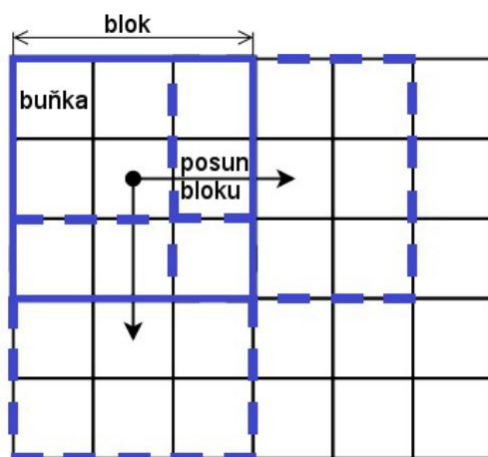
Ne-normalizovaný vektor $\|v\|_k$ představuje k -tou normu a e je malá konstanta. Speciálním případem je L_2 -Hys, kde je vektor v nejprve normalizován a poté dojde k tzv. oříznutí (ang. clipping), viz Obrázek 1.9.



Obrázek 1.9: Ukázka oříznutí histogramu

Deskriptor

Deskriptor metody HOG, je vytvořen tak, že se nejdříve rovnoměrně rozdělí vstupní obraz do čtvercových bloků, které jsou později rozděleny na menší části tzv. buňky. Velikost bloků je definovaná jako délka hrany k pixelů a buňka obdobně pomocí l pixelů ($l < k$). Pro jednotlivé buňky je vypočten normovaný histogram orientací gradientů. Výsledné gradienty z každé buňky jsou následně zprůměrovány a tvoří popis celého bloku. Blok se poté posune v obraze o daný počet pixelů a celý proces se opakuje. Deskriptor HOG je tvořen řetězcem všech popisů jednotlivých bloků vstupního obrazu, viz Obrázek 1.10, kde se blok posouvá o dvě buňky.



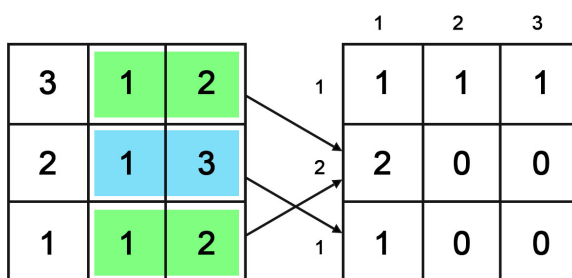
Obrázek 1.10: Princip konstrukce HOG deskriptoru

V případě použití čtvercových bloků je výsledný deskriptor popisován jako R-HOG. Pokud je tvar bloků kruhový, je místo klasického R-HOG použito značení C-HOG.

1.3.3 Haralickovy příznaky

Haralickovy texturní příznaky [18] byly sestaveny Robertem Haralickem v roce 1973. Statisticky popisují čtrnáct různých vlastností obrazu, jako je jeho hrubost, či entropie. Textura definuje opakování/chování nějakého vzoru, barev pro daný objekt, např. cihly, písek, školní uniformy. Aby mohl být objekt klasifikován na základě textury musí mít na svém povrchu pravidelné vzory a barvy. Typickými vlastnostmi jsou hrubost a tvrdost atd.

Pro extrakci texturálních informací z matice GLCM (z ang. Gray Level Co-occurrence Matrix) je použito čtrnáct metod. GLCM používá ve snímkách koncept sousednosti, neboli jsou hledány páry sousedních hodnot pixelů. Tyto dvojice jsou poté ukládány do menší matice 255×255 , kde souřadnice buňky odpovídají dané dvojici hodnot, v buňce je čítač, který počítá počet výskytů, viz Obrázek 1.11. Existují čtyři způsoby pro výběr dvojic: zleva-doprava, shora-dolů, levý horní – pravý dolní a pravý horní – levý dolní.



Obrázek 1.11: Tvorba jedné GLCM matice. Vpravo vstupní obraz, vlevo GLCM matice

Vektor Haralick příznaků je vytvořen výpočtem čtrnácti strukturálních příznaků ze všech čtyř GLCM matic. Obvykle vektory příznaků mívají pouze 13 dimenzí, protože výpočet posledního příznaku je velmi časově náročný.

Notace:

Pro tuto kapitolu značí $P(i, j)$ počet nalezených dvojic v obraze, který je uložený v GLCM matici, na souřadnicích i a j , $p(i, j)$ normalizovaný záznam $P(i, j)$. $p_x(i)$ i -tý záznam mezní pravděpodobnostní matice získaném sečtením řádků, N_g je počet rozdílných úrovní šedé v kvantovaném obraze a R je normalizační konstanta, která udává celkový počet pixelů v dané oblasti.

$$\begin{aligned}
 p(i, j) &= \frac{1}{R} \sum_{j=1}^{N_g} P(i, j) , \\
 p_x(i) &= \sum_{j=1}^{N_g} p(i, j) , \\
 p_y(j) &= \sum_{i=1}^{N_g} p(i, j) , \\
 p_{x+y}(k) &= \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j), \quad i + j = k, \quad k = 2, 3, \dots, 2N_g , \\
 p_{x-y}(k) &= \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j), \quad |i - j| = k, \quad k = 0, 1, \dots, N_g - 1 .
 \end{aligned}$$

Texturální příznaky:

1. Druhý úhlový moment

$$f_1 = \sum_i \sum_j [p(i, j)^2] \quad (1.16)$$

2. Kontrast

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left[\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right] \quad (1.17)$$

3. Korelace

$$f_3 = \frac{\sum_i \sum_j (i, j) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (1.18)$$

kde $\mu_x, \mu_y, \sigma_x, \sigma_y$ jsou průměry a standardní odchylky p_x a p_y

4. Součet čtverců/variance

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j) \quad (1.19)$$

5. Inverzní rozdílný moment

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \quad (1.20)$$

6. Součet průměrů

$$f_6 = \sum_{i=2}^{2N_g} ip_{x+y}(i) \quad (1.21)$$

7. Součet odchylek

$$f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i) \quad (1.22)$$

8. Součet entropií

$$f_8 = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log[p_{x+y}(i)] \quad (1.23)$$

9. Entropie

$$f_9 = - \sum_i \sum_j p(i, j) \log[p(i, j)] \quad (1.24)$$

10. Rozdíl variancí

$$f_{10} = \text{variance of } p_{x-y} \quad (1.25)$$

11. Rozdíl entropií

$$f_{11} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log[p_{x-y}(i)] \quad (1.26)$$

12. Informační míra korelace

$$f_{12} = \frac{HXY - HXY1}{\max(HX, HY)} \quad (1.27)$$

13. Informační míra korelace 2

$$f_{13} = (1 - \exp[-2(HXY2 - HXY)])^{\frac{1}{2}} , \quad (1.28)$$

$$HXY = - \sum_i \sum_j p(i, j) \log[p(i, j)] , \quad (1.29)$$

kde HX , HY jsou entropie p_x , p_y a

$$HXY1 = - \sum_i \sum_j p(i, j) \log[p_x(i)p_y(i)] ,$$

$$HXY2 = - \sum_i \sum_j p_x(i)p_y(j) \log[p_x(i)p_y(i)] .$$

14. Maximální korelační koeficient

$$f_{14} = (\text{second largest eigenvalue of } Q)^{\frac{1}{2}} , \quad (1.30)$$

kde

$$Q(i, j) = \sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)} .$$

Deskriptor

Pro danou vzdálenost d jsou vypočítané čtyři úhlové černo-bílé prostorově závislé matice. Výsledkem je množina čtrnácti měření pro každou ze čtyř maticí. Hodnoty průměru a rozsahu jednotlivých měření jsou zprůměrovány čtyřmi a jsou obsaženy ve výsledné množině 28 příznaků, která tvoří deskriptor Haralick příznaku.

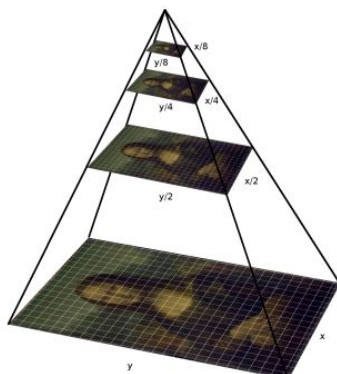
1.3.4 ORB

ORB (z ang. Oriented FAST and Rotated BRIEF) byl vyvinutý v roce 2011 v laboratořích OpenCV jako efektivní životaschopná alternativa k metodám SIFT a SURF. Hlavním důvodem pro vytvoření metody ORB byl fakt, že SIFT a SURF jsou patentované algoritmy, zato ORB je zadarmo k použití.

Detekce klíčových bodů

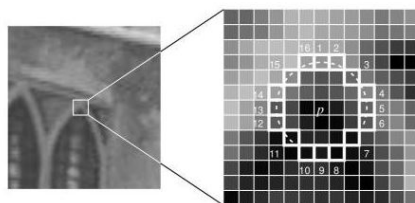
FAST (z ang. Features from accelerated and Segments Test) a jeho invarianty jsou metody pro hledání klíčových bodů. Používá se pro systémy, které pracují ve skutečném čase, např. paralelní trackování a mapování. Je velmi efektivní a nachází rozumné rohové klíčové body, přestože je invariantní vůči škálování, kterého dosahuje díky pyramidovému schématu, viz Obrázek 1.12

Mnoho detektorů klíčových bodů obsahuje orientační operátor, např. SIFT a SURF, ale FAST ho neobsahuje. Existuje mnoho způsobů, jak popsat orientaci klíčového bodu, často je používán histogram gradientů, který je ale velmi výpočetně náročný. Například SURF používá pouze aproximace, aby snížil výpočetní náročnost, ale výsledné odhady nejsou moc přesné [19].



Obrázek 1.12: Každá úroveň pyramidy obsahuje zmenšenou verzi předchozího

FAST porovnává jasovou informaci daného pixelu p s ostatními v kruhovém okolí šestnácti pixelů, viz Obrázek 1.13. Pixely jsou poté seřazeny do tří skupin: světlejší než p , tmavší než p a podobné p . Pokud více jak 8 pixelů je tmavších nebo stejných, tak je pixel p vybrán jako klíčový bod. Tato operace je provedena na všech úrovních pyramidy.



Obrázek 1.13: Hledání klíčového bodu pomocí metody FAST

Dále je bodům přiřazena orientace pomocí změn intenzity v jejich okolí. Pro detekci změny intenzity ORB používá centroid, který předpokládá, že roh je odsazen od středu. Tento vektor je použitý pro výpočet orientace pomocí momentů:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \quad , \quad (1.31)$$

kde je možné najít těžiště (ang. center of mass) pomocí:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad , \quad (1.32)$$

dále je vytvořen vektor ze středu O do centroidu OC . Orientace je dána:

$$\Theta = \arctan^2(m_{01}, m_{10}) . \quad (1.33)$$

Nakonec je spočítána orientace pro okolí bodu, které bude poté orotováno do kónické rotace a následně bude vypočítán deskriptor. Díky těmto krokům je příznak „trochu“ invariantní vůči rotaci.

Deskriptor

BRIEF (z ang. Binary robust independent elementary feature) je deskriptor příznaků, který používá jednoduché binární testy mezi pixely ve vyhlazeném obraze. Jeho výkon je v mnoha ohledech podobný metodě SIFT včetně odolnosti vůči jasů, rozmazání a perspektivy. BRIEF je však velmi citlivý na rotaci v rovině.

Deskriptor BRIEF všechny klíčové body převede na vektory binárních příznaků. Každý klíčový bod je popsán vektorem příznaku, který je tvořen bitovým řetězcem o délce 128-512 bitů.

Nejdříve je obraz vyhlazen Gaussovým jádrem, aby se zvýšila robustnost vůči vysokofrekvenčnímu šumu. Poté BRIEF náhodně zvolí dva pixely a definuje tak okolí klíčového bodu. Tím definuje okolí kolem pixelu, které je nazýváno patch. Pixely byly náhodně zvolené podle normálního rozdělení a mají rozdílné odchylky. Pokud je první pixel světlejší než druhý, tak je výsledek 1. V opačném případě je výsledkem 0.



Obrázek 1.14: Ukázka výběru dvou pixelů v okolí

Dále jsou náhodně zvoleny další pixely a jsou jim přiřazeny hodnoty dle metody popsané výše. Pro 128-bitový vektor se tento proces opakuje 128 krát. BRIEF vytvoří tolik takových vektorů kolik je klíčových bodů v obraze. Přestože BRIEF není invariantní vůči rotaci, tak ORB používá rBRIEF (z ang. rotation-aware BRIEF). Robustnost ORB je díky tomu zvětšena bez ztráty na rychlosti. Vyhlazená část obrazu a binární test τ jsou definované:

$$f(n) = \sum_{1 < i < n} 2^{i-1} \tau(p; x_i, y_i) , \quad (1.34)$$

kde $\tau(p; x, y)$ je definované jako:

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases}, \quad (1.35)$$

kde $p(x)$ je hodnota intenzity pixelu x .

Schopnost napárování příznaků deskriptoru BRIEF rychle upadá po rotaci obrazu i o pár stupňů, např. při rotaci o více jak 45 stupňů je poté schopnost správného napárování téměř rovna nule. ORB proto nabízí metodu *steer BRIEF*, která ukládá orientaci klíčových bodů. Pro množinu příznaků z n binárních testů (x_i, y_i) je potřeba $2 \times n$ matice:

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix}. \quad (1.36)$$

Steer BRIEF využívá orientaci θ a ta odpovídá rotaci matice $R\theta$ a vytvoří řízenou (*steered*) verzi S_θ z S :

$$S_\theta = R_\theta S, \quad (1.37)$$

Steered BRIEF operátor:

$$g_n(p, \theta) = f_n(p) | (x_i, y_i) \in S_\theta. \quad (1.38)$$

Dále diskretizuje okolí na $2\pi/30$ (12 stupňů) a vytvoří tabulku pro předpočítané BRIEF vzory. Pokud je orientace klíčového bodu θ skrz všechny pohledy konzistentní, tak bude na výpočet deskriptoru použita správná sada bodů S_θ . *rBRIEF* je velké vylepšení ve varianci a korelaci oproti metodě *steered BRIEF*. [19]

1.3.5 SIFT

Metoda SIFT (z ang. Scale Invariant Feature Transform) byla publikována v roce 1999 Davidem Lowem [20] a v roce 2004 byla patentována¹ [21]. Část vlastností metody SIFT vyplývá rovnou z jejího názvu. SIFT příznaky jsou invariantní na škálování, rotaci a částečně invariantní vůči změnám jasu a pohledu 3D kamery. Jsou také dobře lokalizovatelné v prostorové a frekvenční oblasti, což snižuje šanci na narušení šumem.

Velmi důležitá část tohoto přístupu je možnost generovat velké množství příznaků, které hustě pokrývají obraz skrze všechny možné velikosti a umístění. Dle autora metody SIFT: „Typická velikost obrázku 500×500 pixelů dává zhruba 2000 stabilních příznaků (Toto číslo závisí jak na obsahu obrazu, tak i výběru parametrů)“ [22]. Počet příznaků je zvláště důležitý pro rozpoznávání objektů, kde schopnost rozpoznat malé objekty v přeplněném pozadí vyžaduje

¹Platnost patentu vypršela během psaní této práce.

alespoň 3 příznaky, které jsou správně přiřazené. Jinak je není možné spolehlivě rozpoznat.

Scale-space peak Selection

Scale-space je funkce $L(x, y, \sigma)$, která se skládá z konvoluce mezi Gaussovým jádrem a různými velikostmi vstupního obrazu

Scale-space je funkce $L(x, y, \sigma)$, která se skládá ze vstupních obrazů o různých velikostech a s postupně větším vyhlazením Gaussovým jádrem. Scale-space je rozdělena na tzv. oktávy a parametry závisí na velikosti vstupního obrazu. Pro detekci bodů/okolí, které jsou invariantní vůči škálování, jsou hledány příznaky, které jsou stabilní skrze více obrazů s různými velikostmi a vyhlazením:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) . \quad (1.39)$$

Každá další oktáva je o polovinu menší než ta předchozí. Takto je vygenerováno několik oktáv. V rámci jedné oktávy se vytvoří několik dalších postupně k-krát více rozmazaných/vyhlazených obrázků, viz Obrázek 1.15.



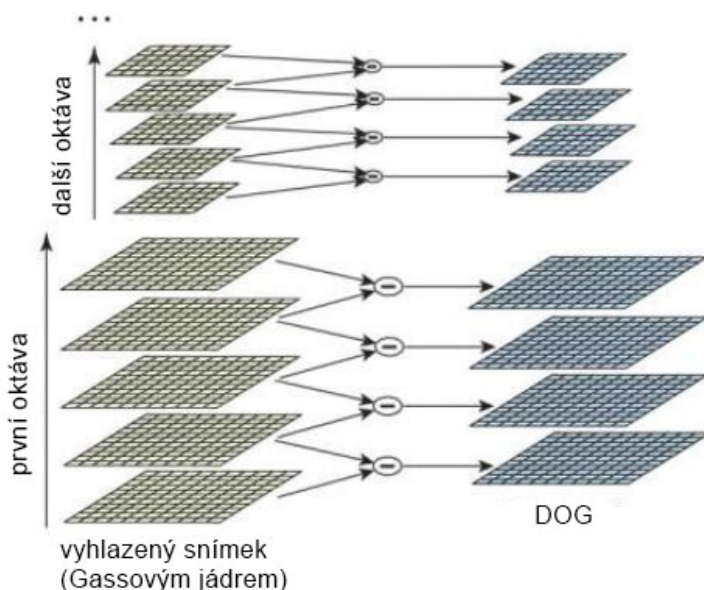
Obrázek 1.15: Ukázka jednotlivých oktáv

DOG

Algoritmus DOG (z ang. Difference of Gaussian kernel) je rozdíl mezi snímky, které jsou rozmazány/vyhlazeny pomocí Gaussova jádra s rozdílným parametrem σ a značí se $D(x, y, \sigma)$:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma). \quad (1.40)$$

Obraz tohoto rozdílu je velmi vhodný pro efektivní hledání kvalitních klíčových bodů. Tento proces se opakuje pro různé oktávy obrazu v Gaussově pyramidě, viz Obrázek 1.16.



Obrázek 1.16: Tvorba DOG

Klíčové body

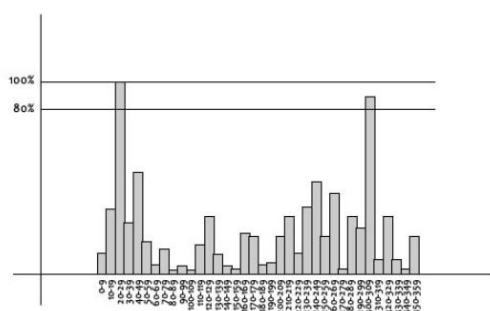
Z výsledku DOG se vypočítá Laplacian of Gaussian tzv. LOG, $\sigma_2 \nabla_2 G$ a ten je invariální vůči škálování. Středový pixel je porovnán se svým okolím osmy pixelů. Toto porovnání je aplikováno na všechny velikosti obrazu a je tak provedeno 26 kontrol. Pokud je daný pixel lokální extrém v rámci dané velikosti, je považován za potenciální klíčový bod.

Takto je získáno mnoho klíčových bodů, ale stále je mezi nimi velké množství nekvalitních. DOG vrací rohy i hrany, kterých se je potřeba zbavit. Pro odstranění hran a dalších slabých bodů se používá 2×2 Hessova matice H , je spočítána v místě a měřítku daného klíčového bodu. Dále jsou odhadovány její derivace, aby byl získán práh pro odstranění slabých klíčových bodů.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (1.41)$$

Klíčové body, které prošly testem, jsou stabilní a bude jim následně přiřazena orientace. Nechtě pro okolí, jehož velikost se odvíjí od velikosti obrazu, je spočítán histogram orientovaných gradientů. Histogram ukazuje četnosti pro okolí, kde je 360 stupňů rozděleno po deseti stupních na 36 kategorií/sloupců.

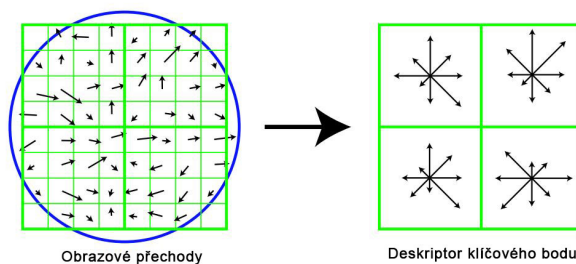
Je vybrán nejvyšší vrchol histogramu, který vytvoří klíčový bod na stejném místě a se stejným měřítkem, ale s upraveným směrem/orientací. Toto velmi zvýší stabilitu při párování.



Obrázek 1.17: Vrcholy nad 80% jsou započítány do směru bodu

Deskriptor

V tomto bodě má každý klíčový bod: umístění, velikost a směr. Další na řadě je výpočet deskriptoru pro každé okolí klíčového bodu, který bude dostatečně rozdílný a invariantní na co nejvíce změn. Pro každý klíčový bod je jeho okolí o velikosti 16×16 rozděleno na 16 pod-bloků o velikosti 4×4 , viz Obrázek 1.18. Pro každý pod-blok je vypočítán orientovaný histogram pro 8 směrů.



Obrázek 1.18: Tvorba vektoru příznaku

Deskriptor o velikosti 4×4 nad vzorkem 16×16 použitým v příkladu se jedná o $4 \times 4 \times 8 = 128$ binárních hodnot, které tvoří vektor příznaku. Dále je vektor normalizován, aby byl zbaven rotační a jasové závislosti.

1.3.6 SURF

Metoda SURF [23] (z ang. Speeded Up Robust Features) je novější alternativa oproti metodě SIFT, kterou je také inspirována. Při vývoji metody SURF se autoři zaměřili zejména na rychlost tak, aby mohl být SURF bez problému použitelný v real-time aplikacích. Příznaky metody SURF jsou invariantní vůči rotaci a vzdálenosti kamery od zachyceného objektu. Existuje také varianta zvaná U-SURF, která není invariantní vůči rotaci a je určena pro použití tam, kde bude kamera nejčastěji v horizontální pozici a nebude docházet k vět-

ším výkyvům/rotaci. Tato metoda byla představena a patentována Herbertem Bayem [24] v roce 2006.

Extrakce příznaků

SURF používá integrální obraz, který je rychlým a efektivním způsobem jak spočítat součet hodnot, v tomto případě pixelů, v daném obraze. Integrální obraz umožňuje rychlý výpočet konvolučních filtrů typu box. Vstupem je integrální obraz $I_{\Sigma}(x)$ na souřadnicích $x = (x, y)^T$, kde je reprezentovaný součet všech pixelů z obdélníkového okolí kolem klíčového bodu x . S vypočítaným $I_{\Sigma}(x)$ stačí provést pouze čtyři sčítání pro výpočet součtu intenzit v obdélníkové oblasti nezávisle na její velikosti. Díky integrálnímu obrazu je možné získat údaje o intenzitě části obrazu v konstantním čase. Stačí pouze znát krajní body daných oblastí.

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1.42)$$

Často používaný Harrisův detektor hran [25] není totiž invariální vůči škálování. Metoda SIFT používá Hessův-Laplaceův detektor, který je více stabilní a opakovatelný oproti Harrisova-Laplaceova detektoru [26]. SURF používá Hessovu matici, protože dosahuje přesných a rychlých výsledků. Pro měření umístění a měřítka SURF používá determinant Hessova matice. Daný pixel a determinant Hessovy matice vypadá takto:

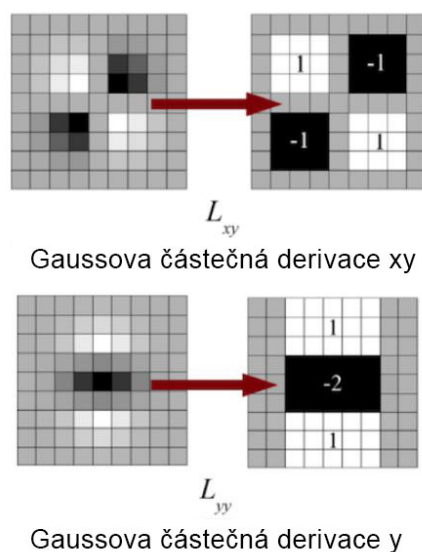
$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}. \quad (1.43)$$

Aby byla metoda invariální vůči škálování, tak je obraz filtrovaný Gaussovým filtrem, takže vzhledem k bodu $X = (x, y)$ je Hessova matice $H(x, \sigma)$ v x stupnici σ definovaná jako:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}, \quad (1.44)$$

kde $L_{xx}(x, \sigma)$ je konvoluce Gaussovy druhé derivace se vstupním obrazem I v bodě x . Kvůli potřebné diskretizaci a oříznutí je metoda poté slabá na rotaci v násobcích $\pi/4$. Tato slabost obecně platí pro všechny detektory založené na Hessově matici.

Pro výpočet determinantu Hessovy matice se nejdříve musí provést konvoluce Gaussovým jádrem a následně druhá derivace. Poté, co měl Lowe úspěch s LOG aproximací u metody SIFT, tak SURF posouvá tuto aproximaci ještě dál s box filtry. Tyto aproximace (obě konvoluce a druhá derivace) mohou být velmi jednoduše vypočítané s použitím integrálního obrazu nezávisle na velikosti. Tento aproximační krok je ten důvod, proč je SURF tak rychlý [27].



Obrázek 1.19: Gaussovy částečné derivace v xy a y

Box filtr 9×9 , viz Obrázek 1.19, je aproximace Gaussovských derivátů druhých řádů s $\sigma = 1, 2$. Tyto aproximace jsou označovány jako D_{xx}, D_{yy} a D_{xy} . Takto je reprezentovaný determinant Hessianovy matice:

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2, \quad (1.45)$$

kde $w = 0,9$.

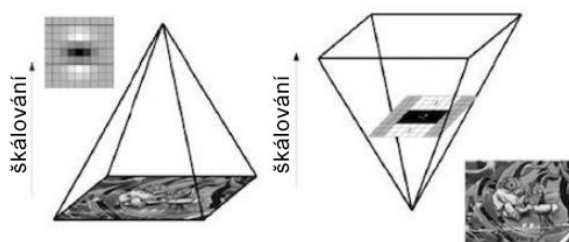
Scale-space

Měřítko prostoru bývá často implementováno jako pyramidové schéma. Díky použití box filtrů a integrálního obrazu SURF nemusí iterativně používat stejný filtr na předchozí vrstvu pyramidy. Místo toho používá stejný filtr libovolné velikosti přímo na vstupní obraz. Tento proces je možné počítat i paralelně. Měřítko prostoru je analyzováno zvětšováním filtru ($9 \times 9 \rightarrow 15 \times 15 \rightarrow 21 \times 21$, atd.) místo zmenšování vstupního obrazu. Pro každou novou oktávu se zvětšuje velikost filtru a zdvojnásobují vzorkovací intervaly pro extrakci klíčových bodů σ . Pro lokalizaci bodů zájmu v obraze v daném měřítku je použita Non-maximum suppression (Nemaximální potlačení) v okolí 3×3 .

Deskriptor

Vytvoření SURF deskriptoru je dvoukrokový proces. Prvním krokem je utvrzení reprodukovatelné orientace, která je založená na datech z kruhového okolí

²Doporučení od autora metody SURF

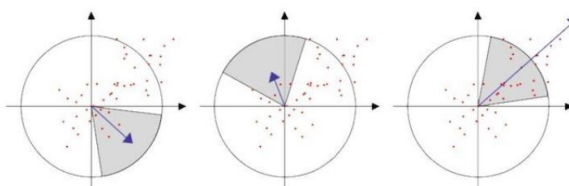


Obrázek 1.20: Pyramidové schéma pro filtry

kolem klíčového bodu. Dále je vytvořena čtvercová oblast se zvolenou orientací a z té je následně extrahovaný příznak SURF.

Deskriptor příznaku je založený na součtu výsledků po použití metody Haar wavelet. Wavelet analýza je velmi podobná Fourierově analýze v tom, že dovoluje, aby cílová funkce byla reprezentována v ortogonální bázi. Tento součet je možné také spočítat pomocí integrálního obrazu.

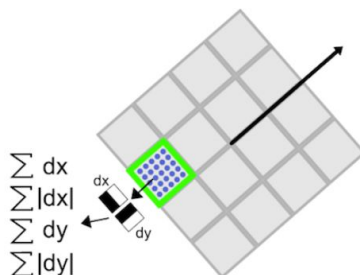
SURF nejdříve spočítá vertikálně a horizontálně Harr-wavelet transformaci. Ta je spočítána pro každé měřítko. Dále je spočítán součet výstupu z Harr-wavelett v daném okolí. Poté je upravena skenovaná orientace na $(\pi/3)$ a poté je přepočítávána, dokud není nalezena orientace s největším součtem, která je poté vybrána jako hlavní orientace deskriptoru příznaku.



Obrázek 1.21: Ukázka hledání orientace deskriptoru příznaku

Extrakce začíná vytvořením čtvercového okolí kolem klíčového bodu a jeho orientace je nastavena na již dříve vypočítanou orientaci. Poté je okolí rozděleno na menší (4×4) regiony. Pro každý region je vypočítáno pár jednoduchých příznaků. Pro zjednodušení značíme výsledky Haar-wavelet transformace jako dx, dy . Robustnost vůči geometrickým deformacím a nepřesnému umístění jsou dx, dy váhovány pomocí Gaussovy funkce od středu bodu zájmu.

Nad každým regionem jsou sečteny výsledky Haar-wavelet transformace (dx, dy) , ty tvoří první vstupy pro vektor příznaku. Informace ohledně polarity a intenzity jsou tvořeny součtem absolutních hodnot $|dx|, |dy|$. Takto má každý region čtyř-dimenzionální deskriptor, vektor $V = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$. Tento vektor má pro všechny 4×4 regiony délku 64 bitů. V případě metody SIFT je délka deskriptoru 128 bitů. [23]



Obrázek 1.22: Tvorba deskriptoru. Každý region je rozdělen na 5×5 vzorků

1.4 Metody párování příznaků

Pro popsání podobnosti dvou příznaků se obecně používá vzdálenost. Konkrétní typy vzdáleností už záleží na použitém algoritmu. Například SIFT používá euklidovskou vzdálenost [14], ORB používá Hammingovu vzdálenost [19]. Čím je délka mezi příznaky menší, tím si jsou podobnější.

1.4.1 Brute force matcher

Brute-force matcher (dále jen BF) je velmi jednoduchý a používá se zejména pro malé datasety, kde se neprojeví jeho neefektivita [28]. BF vezme příznak z prvního obrazu a spojí je se všemi ostatními příznaky z druhého obrazu. Poté na základně vzdálenosti mezi nimi vybere spojení, které má nejkratší délku a pokračuje dále. BF se často používá v kombinaci s Cross-check testem, aby dosahoval přesnějších výsledků.

1.4.2 FLANN based matcher

FLANN [29] (z ang. Fast Library for Approximate Nearest Neighbors) obsahuje několik optimalizovaných algoritmů pro hledání nejbližšího souseda ve velkých datasetech pro více dimenzionální příznaky. Je mnohem rychlejší než Brute-force matcher a to zejména na větších datasetech.

FLANN based matcher dostává na vstupu dva slovníky (index a vyhledávací parametry), které vybírají strukturu, která bude používána, počet např. náhodných kd-stromů nebo k-průměrových stromů a počet kolikrát se může algoritmus vrátit zpět v rámci rekurzivního průchodu. Ve výsledném testování v práci, kde byl FLANN představen, nejlépe vycházely dvě struktury k-průměrové stromy s prioritním vyhledáváním a náhodné kd-stromy.

Volba optimálních parametrů je závislá na několika faktorech, např. struktuře datasetu. Základními třemi parametry jsou vyhledávací čas, stavba struktury a paměťová náročnost [30]. Jedná se o optimalizační problém v parametrickém prostoru. V některých případech strom bude postaven jen jednou nebo

je potřeba se paměťově vejít do nějakého limitu. Důležitost jednotlivých parametrů je možné ovládat pomocí vah, kde jednotlivé váhy jsou značeny jako build-time w_b a paměť w_m . Pro výpočet celkové ceny je použitý vzorec:

$$cost = \frac{s + w_{bb}}{(s + w_{bb})_{opt}} + w_{mm} , \quad (1.46)$$

kde b značí dobu stavby stromu a $m = m_t/m_d$ značí poměr mezi pamětí použitou na strom a data.

1.5 Metody filtrování příznaků

V této podkapitole jsou popsány metody pro eliminaci špatných napárování/spojení mezi příznaky.

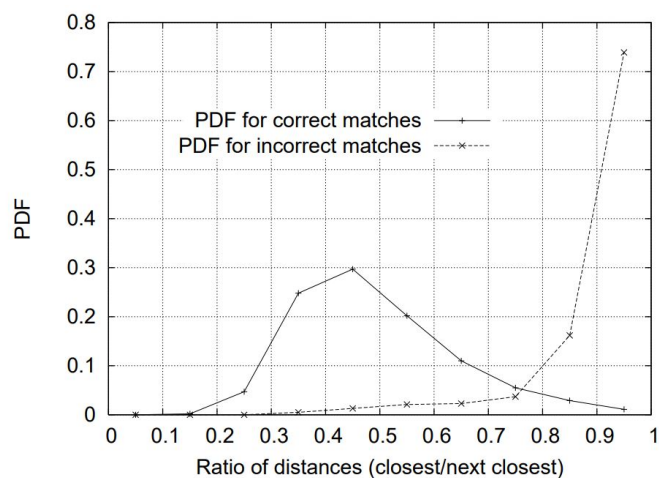
1.5.1 Loweho poměrový test

Loweho poměrový test [22] (ang. Lowe's ratio test) je také nazýván jako poměr vzdálenosti nejbližšího souseda (ang. Nearest neighbour distance ratio) a byl představen v roce 2004. Pro mnoho příznaků neexistuje dokonalá a správná shoda s těmi v databázi. Loweho poměrový test umí rozhodnout, zda má smysl daný příznak vyřadit, či nikoli. Globální práh vzdálenosti podle nejbližšího příznaku nevrací moc dobré výsledky. Někaké příznaky jsou více rozdílné než jiné. Více efektivní měřítko nám dává porovnání nejbližší a druhé nejbližší vzdálenosti. Tato metoda funguje dobře, protože správná spojení budou mít mnohem blíže nejbližšímu sousedovi oproti špatné shodě.

Druhá nejbližší shoda může být chápána jako odhad hustoty falešných shod v této části prostoru a současně identifikovat konkrétní případy nejasnosti příznaku. Na obrázku 1.23 lze vidět, že mezi 0,7 a 0,8 nastává zlom, kdy většina spojení bývá chybná. Tento graf je publikován v práci D. Loweho [22], kde testoval svoji metodu na databázi o 40 000 klíčových bodech.

1.5.2 Cross-check test

Nejedná se přímo o způsob párování příznaků, ale o jeho vylepšení, zpřesnění. Jak již vyplývá z názvu, tak místo jednostranného porovnání příznaků ze snímku1 se snímek2 bude provedené i porovnání snímku2 se snímek1. Například s použitím Brute-force párování výsledné páry budou jen ty, které utvořili výsledné páry v obou směrech [28]. Takový přístup vrací nejlepší výsledky s minimálním počtem odlehlých hodnot (pokud existuje dostatek shod). Cross-check test je alternativa k poměrovému testu, který používá D.Lowe ve své práci o metodě SIFT.



Obrázek 1.23: Pravděpodobnost, že dojde ke správnému spojení příznaků za použití sousední vzdálenosti. Na horizontální ose je popsán poměr vzdáleností mezi nejbližší a druhou nejbližší vzdáleností. Na vertikální ose je popsána funkce hustoty pravděpodobnosti (ang. Probability Density Function). Plná čára značí správné spojení a čerchovaná ty nesprávné

1.5.3 RANSAC

Metoda RANSAC [31] (ang. Random Sample Consensus) slouží pro přizpůsobení modelu experimentálním datům. Metoda přizpůsobuje model tím, že eliminuje odlehlé hodnoty. Článek, ve kterém byl RANSAC představen, se věnuje popisu aplikace metody RANSAC pro určení polohy tzv. LDP (ang. Location Determination Problem).

Analýza

V této kapitole je popsána analýza informací z rešeršní části, vlastních pozorování, porovnání jednotlivých algoritmů, metod předzpracování a výběr těch nejvhodnějších pro dané zadání. Dále kapitola obsahuje řešení otázky, zrychlení vyhledávání v databázi budov.

2.1 Výběr vhodného typu příznaků

Metoda ORB je první, která byla zvažována. Zejména z důvodu, že není patentovaná na rozdíl od metod SIFT, SURF a je jejich plnohodnotná alternativa. Na základě výsledků v obecných testech ve vědeckých pracích jsem zvolil metodu SIFT, ta dosahovala ve většině případů lepších výsledků než ORB a její použití pro akademické účely je povoleno. Během psaní této práce skončila platnost patentu metody SIFT.

2.1.1 Podrobné porovnání vlastností

Rublee et al. uvádějí [19], že metoda ORB má lepší výsledky než SIFT při rozpoznávání venkovních scén. Další pozorovanou vlastností metody ORB je pozice nalezených klíčových bodů, které se oproti metodě SIFT, vyskytují zejména kolem středu snímku. SIFT je pomalejší oproti metodám ORB a SURF (přibližně o řád pomalejší).

Je vhodné zmínit, že rovnoměrnější rozdělení klíčových bodů, tak jak je detekuje metoda SIFT, lze považovat spíše za výhodu, neboť ne vždy se musí budova nacházet uprostřed snímku.

Článek [32], který porovnává všechny tři metody SIFT, SURF, ORB a měří úspěšnost dle počtu spojených příznaků z obou snímků vůči nalezeným příznakům a rychlosti výpočtu. Článek je zaměřen na měření citlivosti a rychlosti výpočtu metod SIFT, SURF a ORB proti změnám jasu, rotaci, škálování, zkosení, zkreslení rybím okem a šumu.

2. ANALÝZA

Cílem této práce je rozpoznávat budovy, proto nejzajímavějšími vlastnostmi jsou škálování, rotace a změny jasu. Při výběru vhodné metody/-deskriptoru pro tuto práci byla největší váha přiřazena právě jim.

Lze předpokládat, že zejména počasí může značně ovlivňovat jasovou složku. Snímky budou přeexponované, či podexponované. V případě přeexponování se nejvíce vytratí obloha, což detekci nijak neomezuje. Obdobně v případě případě podexponování, kde se ale mohou nějaká stinná místa úplně ztratit.

Tabulka 2.1: Výsledky porovnání se měnící se intenzitou

	Čas (sek)	Klíč. body1	Klíč. body2	Spojení	Úspěch spojení
SIFT	0,13	248	229	183	76,6
SURF	0,04	162	166	119	72,6
ORB	0,03	261	267	168	63,6

Při pořizování snímku je počítáno s rotací menší jak 45 stupňů. Nedává moc smysl počítat s možností, kde uživatel bude pořizovat fotografie, které jsou otočené o 90 nebo 180 stupňů. Na tabulce 2.3 je vidět, že úspěšnost párování strmě klesá, ale v případě metody SIFT se stále pohybuje okolo 65%. Toto číslo potvrzuje také tabulka 2.2, kde druhý obrázek byl otočený zhruba o 45 stupňů.

Tabulka 2.2: Výsledky porovnání v otočeném obraze

	Čas (sek)	Klíč. body1	Klíč. body2	Spojení	Úspěch spojení
SIFT	0,16	248	260	166	65,4
SURF	0,03	162	271	110	50,8
ORB	0,03	216	423	158	45,2

Tabulka 2.3: Úspěšnost párování vůči rotaci obrazu o daný úhel

Úhel rotace	0°	45°	90°	135°	180°	225°	270°
SIFT	100	65	93	67	92	65	93
SURF	99	51	99	52	96	51	95
ORB	100	46	97	46	100	46	97

Obrázek číslo 1 byl přibližně o polovinu menší než obrázek 2. Velký náskok v přesnosti metody ORB si vysvětlují ve výrazně nižším počtu klíčových bodů u obrázku 2. Větší smysl má se řídit počtem bodů na obrázku 1 a celkovým počtem párů. V tomto případě nejlepších výsledků dosáhla metoda SIFT.

Metody byly testovány na různé transformace a zkreslení. Jediný případ, kdy SIFT ztrácel byly případy s rotací o 90 stupňů a v silně zašuměných

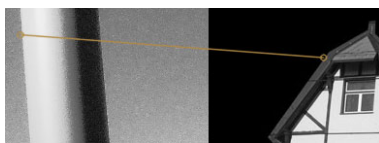
snímcích. Rotaci o více jak 45 stupňů byla zavržena a velmi zašuměné snímky jsou nejčastěji pořizovány ve špatných světelných podmínkách, např. v noci³.

2.1.2 Párování příznaků

Předpokládá se, že se bude databáze budov postupně zvětšovat, kvůli tomu byl zvolen FLANN based matcher jako způsob párování příznaků. Pro odfiltrování nevhodných/špatných příznaků je vybrán Loweho poměrový test. Podle výsledků jeho studie [22] je zvolen práh na hodnotu 0,6, viz Obrázek 1.23. Tato volba je trochu striktní, ale je potřeba získat minimálně čtyři správně napárované příznaky na dané budově.

2.2 Porovnání postupů pro předzpracování obrazu

Metoda lineární změny jasu a kontrastu byla rovnou zavržena, protože by v obraze mohlo často docházet ke clippingu oběma směry. Tím by mohlo vznikat mnoho špatných příznaků, které by mohly velmi ovlivnit výsledky párování. Velkým problémem by také mohla být samotná volba konstant pro výpočet. Počasí je nevyzpytatelné a silně přexponované, či podexponované snímky by mohli tvořit problémy, viz Obrázek 2.1.



Obrázek 2.1: Ukázka chybného párování

Nejlepším způsobem jak předejít této situaci, je zaměřit se při získávání fotek pro dataset na to, aby fotky obsahovaly, co nejvíce kresby. Případné malé nuance je už možné doladit v postprodukci, kde je již možné použít i lineární změny jasu a kontrastu a manuálně nastavit proměnné. Toto řešení není možné použít pro vstupní snímek. Případné vzniklé špatné páry je možné

³Noční módy na telefonech nejsou brány v potaz. Pravděpodobně by bylo nutné přizpůsobit celý dataset fotografií.

Tabulka 2.4: Výsledky porovnání se zvětšeným obrazem

	Čas (sek)	Klíč. body1	Klíč. body2	Spojení	Úspěch spojení
SIFT	0,25	248	1210	232	31,8
SURF	0,08	162	581	136	36,6
ORB	0,02	216	471	181	49,5

eliminovat pomocí metod RANSAC (ang. Random Sample Consensus) nebo mediánovou regresí (ang. Least Median of Squares Regression).

Normalizace a ekvalizace histogramu

Normalizace představuje klasické přemapování jednoho intervalu na druhý. Použití normalizace má smysl pouze, pokud histogram upravovaného snímku není na celém intervalu 0-255. Stačí aby celý obrázek byl tmavý a obsahoval jeden bílý pixel, viz Obrázek 2.2, kde by normalizace neměla žádný efekt.



Obrázek 2.2: Ukázka ekvalizace a adaptivní ekvalizace histogramu na dvou snímcích. Zleva: originální snímek, ekvalizace histogramu a adaptivní ekvalizace histogramu

V první verzi prototypu byla použita klasická ekvalizace histogramu, ale po rozšíření datasetu o nové budovy, které byly vyfoceny za jiných světelných podmínek než ty první, tak se ukázalo, že klasická ekvalizace histogramu není dostatečná a i ona dosahuje přetékání jasových hodnot.

Snímky pořízené během ranních nebo odpoledních hodin, kdy jsou tzv. ostré a dlouhé stíny. Ostré stíny poté tvoří výrazné jasové přechody. Na těchto přechodech jsou poté vypočítány příznaky, které se často párují s příznaky na krajích segmentovaných budov v databázi. S takovými stíny si obyčejná ekvalizace histogramu neporadí, ale adaptivní ekvalizace histogramu ano.

Adaptivní ekvalizace histogramu je velmi užitečná u snímků, kde se nacházejí temné stíny a silně osvětlená místa, viz Obrázek 2.2. V případě dobře osvětleného obrazu bez stínů adaptivní ekvalizace zvýší kontrast. Moc vysoký kontrast není také úplně ideální, ale je lepší než aby byl snímek přexponovaný/podexponovaný. Pokud je ale použita i adaptivní ekvalizace histogramu na obrazy z datasetu, tak je úspěšnost párování výrazně vyšší.

Gama korekce

Také úprava obrazu pomocí gama korekce není úplně vhodná pro vstupní obraz, protože je velmi komplikované automaticky měnit hodnotu konstanty γ tak, aby výsledný obraz neztrácel kvalitu a informace. Tato metoda může být, ale užitečná při tvorbě datasetu. Pokud se nepodaří zachytit vhodný snímek, je možné jej vylepšit právě díky gama korekci.

Škálování

Cílem je snížit velikost dané fotografie ze zbytečně velkého rozlišení, zejména kvůli zlepšení výkonu a paměťové náročnosti. Například ve vědeckých článcích, ze kterých jsem čerpal, byly používány snímky o velikostech 640×480 pixelů. Díky tomu, že metody jsou invariantní vůči škálovatelnosti snímků, není tedy jejich zmenšování žádný problém. Velikost obrazu je také úměrná vůči počtu nalezených klíčových bodů.

Během testování bylo zjištěno, že pokud jsou oba snímky podobně velké, tak výsledky párování mají vyšší úspěšnost, než při velmi rozdílných velikostech.

Ořez (ruční)

Byl zvolen ruční ořez budovy jako jeden z přípravných kroků pro snímek budovy, aby mohl být přidán do datasetu. Důvodem je eliminace okolních staveb a jiných rušivých elementů, např. osob, aut. Pokud se bude jednat o složitou stavbu s výrazným okolím, tak není žádané, aby přidání této budovy do datasetu zabralo hodiny času, proto ořez nemusí být úplně dokonalý. Tento krok má předejít chybnému párování i samotné detekci klíčových bodů jiných budov, viz Obrázek 2.3.

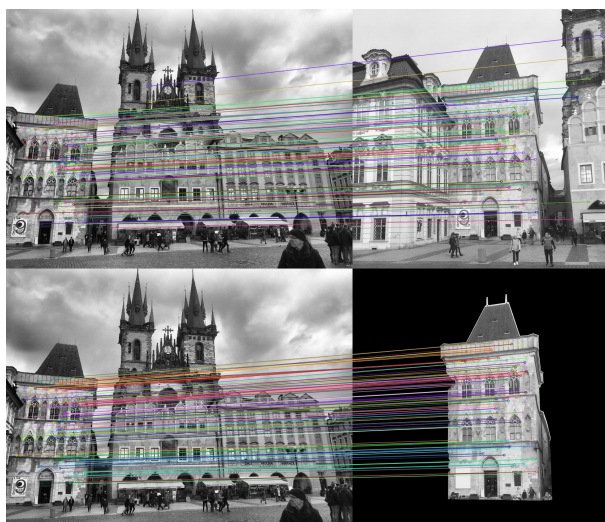
Volba vhodného pozadí

V rámci testování na datasetu lépe vyšlo černé pozadí. Při použití bílého pozadí bylo nalezeno více chybných spojení mezi příznaky. Například byly spojeny příznaky, které se nacházely na okraji střechy budovy s přechodem do oblohy. Tento příznak byl spojen s bílým pozadím a šedými zdmi. Jelikož obloha tvoří velkou část pohledu na budovu, tak by pravděpodobnost špatného spojení byla mnohem větší s bílým pozadím. Černé pozadí je párováno s ostrými stíny, ty ale nejsou tak časté na rozdíl od oblohy.

Mazání rušivých prvků

O mazání velké části rušivých elementů se postará ořez. Mezi rušivé prvky v obraze jsou považovány ty, které se na obraze vyskytují v momentu pořizování snímku pro dataset, ale z dlouhodobého hlediska nedávají smysl, aby byly

2. ANALÝZA



Obrázek 2.3: Ukázka sto nejlepších spojení na snímku. Horní: bez úpravy, Spodní: s oříznutou budovou s černým pozadím a neupraveným obrázkem

součástí obrazu v datasetu, např. plakáty a cedule. Na Obrázku 2.4 lze vidět, že na cedulích bylo nalezeno mnoho příznaků a všechny z nich mohou ovlivnit následné párování. Proto v rámci přípravy snímku před přidáním do datasetu musí být tyto značky, plakáty smazány.

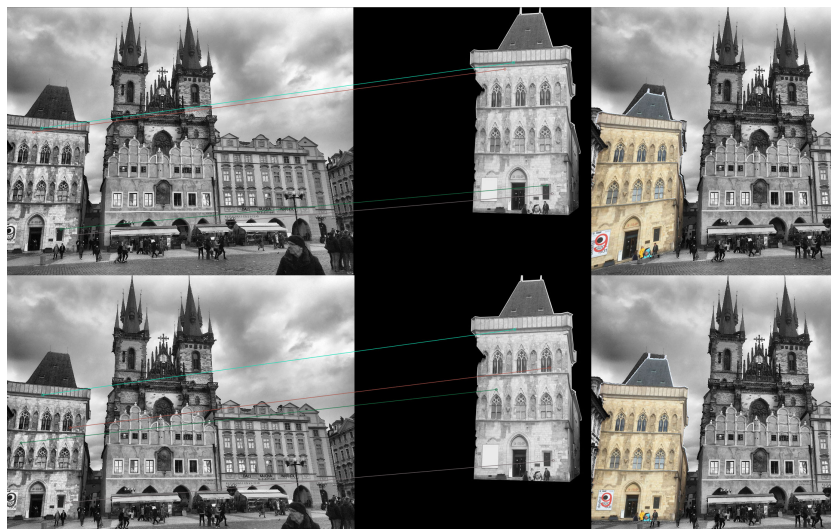


Obrázek 2.4: Mnoho příznaků je také nalezeno na plakátech a značkách

2.3 Výběr nejlepších bodů

Aby mohla být budova umístěna do reálné scény, tak musí být vytvořena transformační matice. Pro její vytvoření je potřeba čtveřice bodů na vstupním snímku a druhá čtveřice na snímku z datasetu. Bylo zjištěno, že nepřesnosti umístění po výsledné transformaci jsou způsobeny zaokrouhlovací chybou na celé pixely. Díky pozorování bylo zjištěno, že je dosahováno lepších výsledků

pokud jsou vybrány 4 body, co možná nejvzdálenější. Další vypočítaná vlastnost je taková, že by vybrané klíčové body neměly ležet na přímce. Viz Obrázek 2.5, kde je viditelný výše představený problém s pozicí bodů.



Obrázek 2.5: Výběr bodů, které jsou rovnoměrněji rozloženy dosahuje vyšší přesnosti. Horní: nejlepší čtyři body, Spodní: rovnoměrněji rozložené body

Předpokladem je, že aplikace bude používat i úpravu renderu po transformaci pro ještě přesnější umístění do scény.

2.4 Zrychlení vyhledávání

Filtrování v databázi pomocí polohy získané z fotografie dokáže podstatným způsobem urychlit proces eliminace zjevně nevhodných kandidátů. Informace o GPS lokalitě jsou např. v metadatech snímku, která jsou známá jako EXIF Data [33] (ang. Exchangeable image file format).

Z metadat je možné získat např. informaci o nadmořské výšce. Formát dat GPS udává zeměpisnou šířku a délku ve formátu: stupně, minuty a sekundy. Častěji je, ale používán formát pouze stupních.

Přesnost samotných dat je reprezentována pomocí čísel za desetinou čárkou, která jsou značena od 1 do 9, dle desetinných míst (minuty a vteřiny jsou převedeny na stupně). Tabulka 2.5 ukazuje úroveň přesnosti pro jednotlivá čísla. Dle [34] je přesnost průměrného přijímače GPS signálu okolo 5 metrů. Tato přesnost je ale snížena blízkostí budov, stromů a vysokou oblačností.

Při používání aplikace se uživatel nedostane dovnitř budovy. Vždy bude v jejím okolí např. 150 metrů. Rozpoznávat budovu na více jak 300 metrů nemá smysl. Budova bude už hodně malá a nemusí být ani dobře viditelná.

2. ANALÝZA

Tabulka 2.5: Výsledky porovnání se měnicí se intenzitou

Desetinné místo	Přesnost	Překlad	Originální názvosloví
1	10 km	Město	Town or City
2	1 km	Sousedství	Neighborhood
3	100m	Městský blok	City Block
4	10 m	Adresa	Street Address
5	1 m	Vstup	Store Entrance
6	10 cm	Ovládání dronů	Drone Control
7	1 cm	Armádní operace	Military Operations
8	1 mm	Stavební inženýrství	Civil Engineering
9	0,1 mm	Vymyšlené	Likely Fabricated

Pokud bude budova v okolí o poloměru 200 metrů, tak projde první úrovní filtrování. V poloměru je započítána nepřesnost zařízení, vlivy okolí a také rozumná vzdálenost na rozpoznávání budov. Tak velký rádius je zvolen z důvodu nepřesností samotného GPS signálu a např. rozsáhle budovy, které jsou velké desítky až stovky metrů, mají lokalizaci jako by byly bod. Proto je volba velikosti okolí kolem uživatele tak vysoké číslo.

Návrh

V této kapitole je popsána softwarová analýza aplikace a její funkčnost. Výsledná mobilní aplikace dovolí uživateli procházet věnná města v době minulé. Pomocí rozšířené reality budou ve městě nahrazeny novodobé budovy za dobové. Uživatel si může projít celé město tak, jak vypadalo ve středověku.

Jádrem této aplikace je algoritmus na rozpoznávání a umístění příslušného renderu do scény. Detailní návrh aplikace bude vytvořen v navazující práci. Tato práce se věnuje pouze tvorbě prototypu jádra na rozpoznávání budov a jejich následné umístění do scény.

3.1 Funkční a nefunkční požadavky

Funkční požadavky

- **V reálném čase** - Výsledná aplikace bude umět rozpoznat budovu a umístit render modelu do scény a poté ho trackovat v reálném čase. V oblasti s největší hustotou budov (v rámci databáze) nebude rozpoznání a umístění trvat déle jak 4 vteřiny.
- **Přizpůsobení podle počasí** - Modely umísťované do reálné scény budou renderované pro danou denní dobu, počasí a roční období. Podle roční a denní doby bude simulována pozice slunce. Počasí bude určovat, zda bude budova zapadaná sněhem nebo se lesknout od deště.
- **Přesné umístění do scény** - Render historické budovy bude umístěn do scény s maximální možnou přesností. Raději budovu neumístit do scény, pokud by byla umístěna velmi špatně. Minimální přesnost, která by byla považována za dobrou, je 0,6 při použití metody IoU (ang. Intersection over Union)
- **Jednoduchá správa budov** - Databáze budov a celého obsahu bude spravovatelná přes webové rozhraní. Bude možné přidávání a spravování

3. NÁVRH

jednotlivých částí budov od snímků, textů až po rendery. Administrace bude obsahovat správu uživatelů a kontaktní formulář pro hlášení chyb, či zpráv ohledně budov, např. budova je nově opravená. To znamená, že je potřeba aktualizovat fotografie budovy. Bude možné přiřazovat práva na jednotlivé části administrace nebo akce, např. správce bude mít možnost upravovat texty budov, ale nemůže celou budovu odstranit z databáze.

- **Ukládání dat** - Uživatel si bude moci stáhnout vybrané budovy lokálně, aby nemusel stahovat všechna data za běhu aplikace.

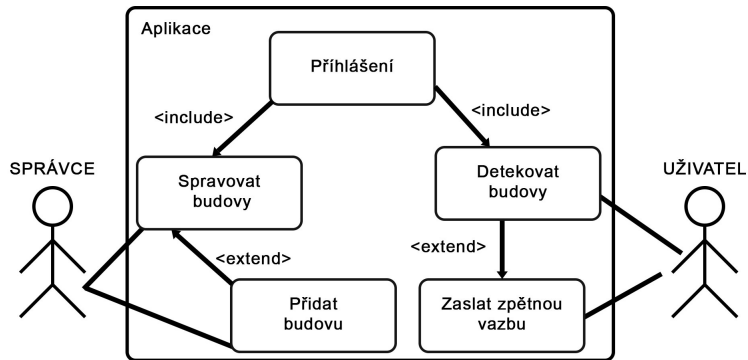
Nefunkční požadavky

- **Platforma** - Aplikace poběží na systému Android verze 9.0 Pie a vyšší. Po uživateli bude aplikace vyžadovat přístup ke kameře, aktuální pozici a přístup na internet.
- **Modularita** - Aplikace by měla být rozdělena na jednotlivé části, které jsou slabě provázané a nezávislé. Díky tomu bude aplikace dobře udržitelná a jednoduše rozšiřitelná.
- **Kvalitní dokumentace** - Na této aplikaci budou pracovat další studenti, proto je potřeba popsat všechny metody i výsledky vlastního testování. Bude vytvořena technická dokumentace, programátorská dokumentace, uživatelský manuál a video tutoriály pro správce, jak spravovat všechny parametry budov a uživatele.

3.2 Případy užití

V této podkapitole jsou ukázky základních případů užití pro uživatele a správce/administrátory, viz Obrázek 3.1. Podrobnější scénář je popsán pro každou činnost níže v textu.

- **Přihlášení/autentizace** - Pro úpravu dat v aplikaci se musí správce nejdříve přihlásit. Uživatel musí být registrovaný a přihlášený, pokud chce aplikaci používat.
- **Přidání budovy** - Správce může přidat novou budovu do aplikace. Pro přidání nové budovy je nutné připravit její podklady. Jedná se o upravené fotografie, k nim přiřazené rendery a informace o budově. Po přidání nové budovy budou automaticky předpočítány příznaky.
- **Správa budovy** - Správce může upravovat informace od popisného textu budovy až po jednotlivé rendery. Správce bude limitovaný přiřazenými právy na úpravu daných částí v administraci.

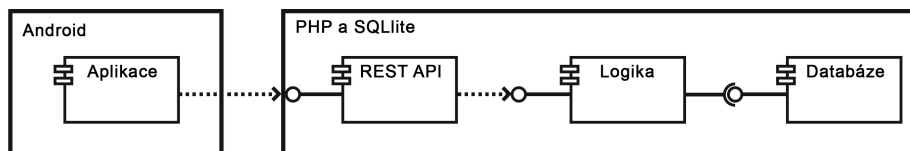


Obrázek 3.1: Use case diagram základních cílů

- **Rozpoznání budovy** - Uživatel pomocí aplikace rozpozná budovu ve scéně a uvidí její středověkou podobu. Po spuštění rozpoznávání uživatel musí mířit fotoaparátem telefonu na scénu, která bude zpracována. V případě nalezení budovy ve scéně je do ní umístěn daný render. V opačném případě uživatel obdrží zprávu, že nebyla žádná budova nalezena a případné tipy ke zlepšení podmínek pro detekci.
- **Zaslání zpětné vazby** - Uživatel může kontaktovat vývojáře, např. o změnách budov, historických nepřesnostech nebo při chybě při rozpoznávání. Může i zaslat snímek obrazovky telefonu spolu s vybranými daty vývojářům.

3.3 Diagram komponent

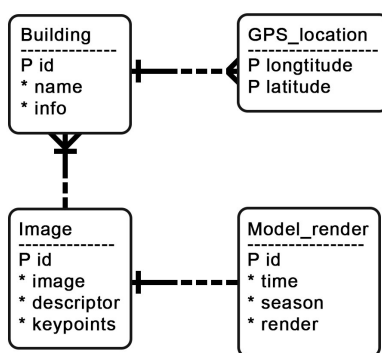
Aplikace bude používat architekturu klient-server, viz Obrázek 3.2. Na straně klienta bude systém Android a na straně serveru může být použité PHP a SQLite pro databázi. Klient bude komunikovat se serverem pomocí REST API. Na straně serveru bude také oddělená business logika.



Obrázek 3.2: Diagram komponent

3.4 Databáze budov

Aplikace bude komunikovat s databází, kde budou uloženy fotografie a rendery budov, jejich předpočítané příznaky a metadata, viz Obrázek 3.3. Budova bude mít uložené parametry: jméno, lokalitu, set obrázků a renderů. Každá budova bude mít několik fotografií z různých úhlů a k nim přiřazené rendery. Tyto fotografie jsou upraveny tak, aby mohly sloužit pro rozpoznávání. Dalšími obrázky budou rendery historických modelů, které budou vyrenderované z podobné pozice jako příslušný snímek budovy.



Obrázek 3.3: Konceptuální model části databáze

3.5 Předzpracování obrazu

Analýza odhalila možný problém s nestálostí počasí. V případě jasného slunce jsou tvořeny velmi ostré stíny a ztlačně se mění odstíny barev budovy. Tyto změny by mohly značně ztížit přesnost párování.

Je navrhováno řešení, kde na základě barevného histogramu a dat o počasí by docházelo k vyvažování barev, např. automatické vyvažování bílé a ekvalizace histogramu by mohly výrazně zlepšit výsledek párování.

3.6 Rozpoznání budovy

Pro efektivnější rozpoznání budou budovy v databázi filtrovány na základě jejich GPS lokalizace, která bude získávána z telefonu. Další zajímavá data pro upřesnění polohy/orientace telefonu mohou být získána např. z gyroskopu. Párování příznaků by se mělo provádět s minimálním počtem budov, které budou v okolí uživatele. Pro výpočet příznaků bude použita metoda SIFT.

Pro menší dataset by mohl být použit i obyčejný brute-force matcher. Vzhledem k tomu, že v budoucnosti se může databáze budov velmi zvětšit,

byl zvolen FLANN matcher s kd-stromy, který je mnohem rychlejší na větších datasetech, viz. podkapitola 1.4.2.

Výsledky napárování příznaků určují, zda byla nějaká budova v obraze nalezena. Pokud byl nalezen dostatek dobrých spojení příznaků, tak je render budovy umístěn do obrazu. Minimální počet pro vytvoření transformační matice jsou čtyři dobrá spojení. Na základě analýzy bylo zjištěno, že pro lepší a přesnější výsledky je dobré relativní vyjádření např. vůči vzdálenosti uživatele od budovy.

3.7 Vložení snímku/modelu do reality

Fotografie budovy, která byla označena jako nejpodobnější má k sobě přiřazené rendery modelu. Každý snímek v databázi bude mít přiřazený render, který bude následně umístěn na správné místo do scény pomocí transformační matice. Podle ročního období, počasí a času bude vybrán render se správným osvětlením.

Přesnost umístění budovy do scény může být zvýšena pomocí následného zpracování. Z analýzy vyplývá, že při transformaci dochází k zaokrouhlovacím chybám. Ty mohou být zmenšeny nebo odstraněny pomocí úpravy transformační matice a následnému otestování, zda je budova umístěna lépe, či nikoliv.

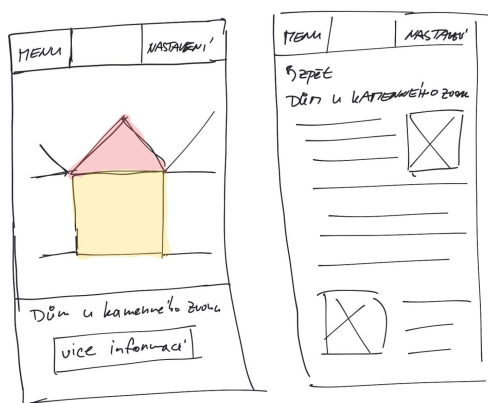
3.8 Výsledná aplikace

Aplikace bude určena zejména pro mobilní telefony s operačním systémem android. Jádro aplikace by mohlo být použito tak, aby fungovalo např. ve spolupráci s Google Street View⁴. Po uživateli bude vyžadována jeho lokace a internetové připojení. Pokud bude budova ve scéně detekována, tak do ní bude umístěna a zobrazí se její název a tlačítko pro zobrazení více informací o budově, viz Obrázek 3.4.

Na tvorbě aplikace se budou podílet i historici, kteří ji budou plnit novým obsahem a budovami. Ve výsledné aplikaci bude možné nastavovat parametry, např. CLAHE (z ang. Contrast Limited Adaptive Histogram Equalization) dle počasí, aby mohlo být dosaženo nejlepších možných výsledků. Počasí a čas bude řídit vyrenderované modely, které budou osvětlené podle denní doby a ročního období. Pokud bude město zasněžené, tak bude i budova zasněžená.

⁴google.com/streetview/

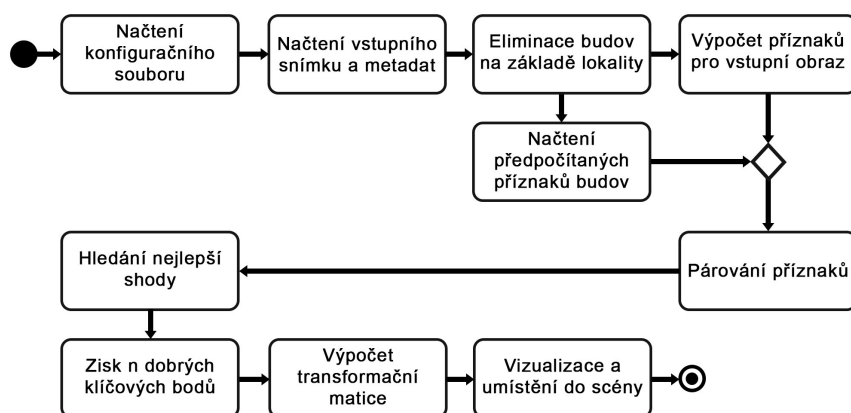
3. NÁVRH



Obrázek 3.4: Wireframe výsledné aplikace

Realizace

V této kapitole jsou popsány zajímavé problémy a jejich řešení. V rámci implementační části práce byla realizována pouze implementace prototypu, který se zaměřuje na rozpoznání budovy a její případné umístění do scény, viz Obrázek 4.1. Celá aplikace přesahuje rozsah bakalářské práce, proto se realizace věnuje jen její části v podobě prototypu, který testuje možnosti/postupy použitelné následně při vývoji aplikace.



Obrázek 4.1: Diagram aktivit

4.1 Technologie

Pro tvorbu prototypu je použit programovací jazyk Python s jeho standardní knihovnou a níže vypsány moduly. Základní parametry lze upravit v konfiguračním souboru. Většina použitých knihoven je učena pro práci s obrazem.

Během vývoje byl používán Github, kde je také repositář⁵ s celým prototypem, dokumentací a návodem na instalaci psaným v Markdown.

- **OpenCV**[35] – (ang. Open Source Computer Vision Library) je hojně používaná knihovna určená pro strojové vidění a učení.
- **NumPy**[36] – je základní balíček pro vědecké výpočty v Pythonu. V této práci jsou použity zejména n-dimenzionální pole.
- **Pillow**[37] – (ang. Python Image Library) je knihovna pro práci a manipulaci s obrazem. V práci se využívá kvůli jednoduché práci s metadaty snímků.

4.2 Konfigurační soubor

Nastavitelné parametry pro prototyp mohou být upraveny v konfiguračním souboru. Konkrétně se jedná o následující seznam. Konfigurační soubor je generován skriptem *config.py*.

- **metadata** – cesta k json souboru s metadaty budov
- **buildings** – cesta k adresáři s budovami
- **input_img** – cesta ke vstupnímu snímku
- **gps** – velikost okolí kolem uživatele
- **matching** – konstanty pro nastavení FLANN a další konstanty, které se týkají párování příznaků

4.3 Tvorba testovacího datasetu

Prvním krokem bylo vytvoření testovacího datasetu. Pro tvorbu nejvhodnějších testů byl vytvořen vlastní dataset míst, které by měly prověřit výsledné řešení. Fotografie jednotlivých budov byly získávány ve stejném okolí tak, aby nemohla být většina z nich eliminována pouze na základě lokality.

Jasové podmínky se mohou často měnit, a proto i některé pořízené fotografie jsou podexponované, či přeexponované, aby otestovaly robustnost řešení. Dataset obsahuje i špatné příklady pro demonstraci chyb při pořizování snímků na rozpoznání budov.

⁵github.com/sefcija4/FeatureDetection

4.4 Databáze budov

Oproti návrhu je v rámci bakalářské práce použita lokální složková struktura, viz Obrázek 4.2. Fotografie objektů jsou rozmístěné do vlastních složek podle jednotlivých budov. V hlavní složce je soubor s metadaty všech budov, viz Výpis kódu 4.1.

```

├── skripty..... všechny skripty .py
├── data..... adresář pro budovy
│   ├── metadata..... metadata budov ve formátu JSON
│   ├── b1..... adresář jedné konkrétní budovy
│   │   ├── obrázky..... upravené snímky budovy
│   │   ├── deskriptory..... vypočítané příznaky
│   │   ├── klíčové body..... vypočítané klíčové body
│   └── b1_original..... adresář budovy s daty pro vizualizaci
│       └── obrázky..... originální snímky budov

```

Obrázek 4.2: Složková struktura prototypu

Soubor s metadaty budov je generován skriptem *json_data.py*. Skript *extract_features.py* vypočítá příznaky pro všechny snímky budov v datasetu. Ve složce každé budovy jsou fotky dané budovy a dva soubory pro každý snímek s předpočítanými klíčovými body a vektory příznaků. Vektory příznaků jsou používány při párování příznaků se vstupním obrazem. Klíčové body jsou potřeba pro následnou vizualizaci párování a výpočtu transformační matice.

```

{
  'id': '0',
  'name': 'Dům U Kamenného zvonu',
  'path': str(Path(f'{config.get_folder_name()}/b1')),
  'latitude': 50.0877694,
  'longtitude': 14.4196813
}

```

Výpis kódu 4.1: Meta data budovy ve formátu JSON

4.5 Úpravu vstupního snímku

Prototyp získává GPS souřadnice z metadat obrázku na rozdíl od návrhu. Volba maximálního rozměru snímku je nastavená na 960 pixelů tak, aby se dvě fotografie při vizualizaci vešly vedle sebe na full-HD monitor.

Vstupní snímek je nejprve zmenšený tak, aby největší rozměr snímku byl 960 px a dále je na něj aplikována adaptivní ekvalizace histogramu (CLAHE).

Z metadat vstupního snímku jsou získána GPS data. V tomto stavu je snímek připraven pro výpočet příznaků.

4.6 Lokalizační data

Oproti návrhu prototyp získává lokalizační data ze vstupní fotografie. Pro získání lokalizačních dat ze snímku ve formátu EXIF je použit upravený objekt vytvořený Justinem Michelem [38], který načte informaci a zeměpisné délce a šířce. Ty jsou původně uloženy ve formátu stupně, minuty a vteřiny. Data jsou převedena pouze na stupně a uloženy do objektu GPS souřadnic.

Poloměr okolí kolem uživatele je zvolen na 250 metrů. V této konstantě je započítána nepřesnost dat o poloze uživatele i budov. Rozpoznávat budovu na více jak 250 m nemá smysl, případně je možné velikost poloměru upravit v konfiguračním souboru.

Například u budovy č.8 (Liberecká radnice) jsou souřadnice budovy nastavené na střed budovy, která má rozlohu přibližně $2500 m^2$ (čtvercový půdorys) a nejvyšší věž má 65 m. Uživatel poté musí stát dalších 60 m od budovy, aby se mu vešla celá do snímku. Ve výsledku uživatel stojí více jak 100 m od středu budovy, která zabírá celou plochu snímku.

4.7 Extrakce a porovnání příznaků

Na základě analýzy byl zvolen SIFT pro výpočet příznaků. Dle návrhu je pro jejich následné párování použitý FLANN based matcher. Příznaky jsou již předpočítané pro snímky z datasetu a výpočet se bude provádět pouze na vstupním snímku. Získaný vektor příznaků je poté porovnán se všemi budovami v okolí a podle výsledných vzdáleností a provedených spojení je budova na snímku rozpoznána, či nikoli.

4.8 Výběr nejlepší shody

Nejlepší shoda je závislá na počtu spojení a vzdáleností. Špatné klíčové body jsou odstraněny před párováním pomocí poměrového testu D. Loweho, kde hodnota prahu je nastavena na 0,6, viz Obrázek 1.23.

Snímek budovy je vybrán na základě nejvyššího počtu dobrých spojení. Dále jsou na tomto snímku vybrány 4 nejlepší body. Při výběru je kontrolována vzdálenost mezi nimi tak, aby byly od sebe nejméně vzdálené 100 pixelů. Tato konstanta byla získána jako výsledek testování na datasetu.

Před odevzdáním práce byly nalezeny metody RANSAC a mediánová regrese, které jsou obsaženy v knihovně OpenCV. Tyto metody je možné použít v prototypu v rámci metody nalezení transformační matice.

4.9 Transformace budovy

Pro vytvoření transformační matice je potřeba čtveřice bodů z obou obrázků. Nalezení této čtveřice je výpočetně náročné. Pomocí čtyř napárovaných bodů je možné vytvořit transformační matici.

Poté se pomocí knihovní funkce `cv2.warpPerspective()` vytvoří nový transformovaný obrázek. Pomocí segmentace jsou vytvořeny masky budovy a reálné scény, které poté umožní složit snímky dohromady, viz Obrázek 4.3.



Obrázek 4.3: Výsledné umístění rozpoznané budovy do scény

4.10 Skládání obrazů

Pro výslednou vizualizaci, kde je vložený snímek budovy barevný a scéna černobílá jsou použity masky. Transformovaná budova má kolem sebe černé pozadí, pomocí metody prahování je vytvořena maska budovy. Pro lepší výsledky je ještě použito morfologické uzavření. U snímků, které obsahovali černé pixely by jinak vznikaly díry v maskách. Celý proces složení je ukázán na Obrázku 4.4. Tento způsob vizualizace byl zvolen pouze pro prototyp v rámci této práce.



Obrázek 4.4: Ukázka postupu skládání obrazu

Testování

V této kapitole je prototyp otestován na rozpoznání budov a jejich umístění do scény. Konkrétně je otestován na předzpracování obrazu, výběr dobrých klíčových bodů a jejich rozmístění ve snímku. Hodnoty, ze kterých se vycházelo, byly získány v rámci analýzy. Prototyp byl vyvíjen na operačním systému Windows 10.

5.1 Technické limitace prototypu

Při použití větší minimální vzdálenosti mezi klíčovými body, jsou často vzdálenější/menší budovy pouze rozpoznány. Budovy by měly tvořit alespoň jednu čtvrtinu plochy vstupního snímku, aby mohl být použitý práh pro minimální vzdálenost mezi klíčovými body. Pro některé budovy čtvrtina snímku představovala přibližně 30 – 60 metrů od budovy. Jinak není možné získat čtveřici klíčových bodů pro tvorbu transformační matice. Předpokládá se, že tato minimální vzdálenost mezi pixely bude relativně vypočítávána dle vzdálenosti od budovy.

Pokud je vstupní snímek pořízován proti silnému slunečnímu svitu, je velmi obtížně budovu rozpoznat pouze s předzpracováním, které tento prototyp využívá. Téměř to samé platí v případě velmi tmavých ostrých stínů, se těmi si však adaptivní ekvalizace často poradí. Další pozorovanou nevýhodou byl styl domu, např. několik stejných oken vedle sebe často způsobuje chybné napárování. Tento problém není tak velký při použití metody RANSAC.

Testovací dataset obsahuje 11 budov⁶, viz příloha C, kde má pouze 9 budov vstupní snímek. Tyto budovy slouží jen jako další možnosti pro párování. Jiné budovy mají více vstupních snímků, ale pro testování je po celou dobu používán pouze jeden. Některé budovy mají schválně nevhodně vytvořený soubor fotek, aby mohl sloužit pro demonstraci slabin.

⁶Budova č.6 není ve výsledném datasetu.

5.2 Testy parametrů

Pro ohodnocení výsledků testování jsou použity následující značení:

- **A** – Správné
- **N** – Nesprávné, v tomto případě není ohodnocená přesnost umístění
- **X** – Nebylo rozpoznáno/umístěno

Ohodnocení přesnosti je rozděleno do čtyř kategorií, viz Obrázek 5.1.

- **1** – Výborné, přesné umístění
- **2** – Dobré umístění, místy lehce neseďí (stěny jsou rovnoběžné)
- **3** – Budova znatelně neseďí a působí křivě
- **4** – Umístění je velmi nepřesné a zkreslené, ale budovu jde stále poznat



Obrázek 5.1: Ukázky hodnocení přesnosti. Zleva 1, 2, 3, 4

Výsledky pro nejlepší nastavení hodnot pro testovací dataset, viz Příloha C ukazují Tabulka 5.1. Hodnoty parametrů byly získány z analýzy a následně potvrzeny testováním. Pro testování byly ostatní hodnoty nastavené na tyto výchozí hodnoty z analýzy a měnily se jen hodnoty jednoho zvoleného parametru, např. způsobu předzpracování, proto jsou výsledky s takovými parametry shodné⁷.

Tabulka 5.1: Výsledky nejúspěšnějšího nastavení parametrů: CLAHE, poměrový test na 0,6 a RANSAC

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	A	A	A	A	A	A	A	A
Přesnost	2	1	1	1	1	1	2	1	3

⁷Opakují se, aby tabulky výsledky daných testů mohli být pohromadě, dle jednotlivých testovaných kategorií



Obrázek 5.2: Výsledky nejuspěšnějšího nastavení parametrů: CLAHE, poměrový test na 0,6 a RANSAC. Budovy na obrázku ze shora (pravý, levý): (1, 8), (4, 9), (5, 10), (7,11), 12

5.2.1 Předzpracování obrazu

Snímky budov jsou předzpracovány před výpočtem jejich příznaků. Převod obrazu do odstínu šedi je nezbytná část procesu pro výpočet příznaků, úpravy jasu a kontrastu atd. mohou být různé. Otestovány jsou následující varianty:

1. Bez ekvalizace histogramu. Viz Tabulka D.1 a výsledné umístění budovy viz Obrázek D.1
2. Ekvalizace histogramu. Viz Tabulka D.2 a výsledné umístění budovy viz Obrázek D.2
3. Adaptivní ekvalizace histogramu. Viz Tabulka D.3 a výsledné umístění budovy viz Obrázek D.3

Adaptivní ekvalizace histogramu byla používání až později po rozšíření datasetu. Výsledky testování potvrzují, že je dosahováno mnohem lepších vý-

sledků než s obyčejnou ekvalizací, která měla nakonec horší výsledky než neupravený obraz. Je to zejména z důvodu clippingu světlých míst. CLAHE si dobře poradí, jak s přexponovanými, tak podexponovanými fotkami.

Po dokončení celého testování bylo uvažováno, zda by vyvážení bíle nepomohlo s posunutím barev do teplých žlutých odstínů. Tato myšlenka už nebyla řádně otestována v rámci tohoto testování a je možné v tomto směru na práci navázat.

5.2.2 Výběr dobrých spojení

Pro výběr dobrých spojení je použit Loweho poměrový test. Ostatní parametry byly nastavené na: adaptivní ekvalizaci histogramu a minimální vzdálenost mezi klíčovými body byla 100 pixelů. Hodnocení rozpoznávání a umístění je popsáno podle úspěchu. Test dobrých spojení proběhl pro následující hodnoty:

1. Práh je nastaven na 0,4. Výsledky viz Tabulka D.4 a výsledné umístění budov viz Obrázek D.4
2. Práh je nastaven na 0,5 viz Tabulka D.5 a výsledné umístění budov viz Obrázek D.5
3. Práh je nastaven na 0,6 viz Tabulka D.6 a výsledné umístění budov viz Obrázek D.6
4. Práh je nastaven na 0,7 viz Tabulka D.7 a výsledné umístění budov viz Obrázek D.7
5. Práh je nastaven na 0,8 viz Tabulka D.8 a výsledné umístění budov viz Obrázek D.8

Testování potvrdilo graf úspěšnosti napárování, viz Obrázek 1.23, kde je viditelná změna poměru dobrých a špatných příznaků. Nejlepší výsledky byly dosaženy pro hodnotu 0,6, viz Tabulka D.6. Představení budov a informace o jejich datasetu naleznete v příloze C, kde je například popsáno, že budovy 4 a 9 mají schválně vytvořený špatný dataset. U těchto budov se většinou podařilo najít pouze 2–3 správné klíčové body.

5.2.3 Selekce klíčových bodů

Výběr klíčových bodů může silně ovlivnit přesnost výsledku. Pro tento test je nastavený práh na deset dobrých spojení. Pokud jich nemá více jak deset (včetně), tak je snímek označen, že budova nebyla nalezena. Otestovány jsou následující varianty:

1. Výběr nejlepších (dle podobnosti/vzdálenosti párování). Výsledky testování viz Tabulka D.9 a výsledné umístění budovy viz Obrázek D.9
2. Výběr nejlepších (dle podobnosti/vzdálenosti párování), minimální vzdálenosti mezi sebou, kde je práh nastaven na 80 pixelů. Výsledky testování viz Tabulka D.10 a výsledné umístění budovy viz Obrázek D.10
3. Výběr nejlepších (dle podobnosti/vzdálenosti párování), minimální vzdálenosti mezi sebou, kde je práh nastaven na 100 pixelů. Výsledky testování viz Tabulka D.11 a výsledné umístění budovy viz Obrázek D.11
4. Výběr nejlepších (dle podobnosti/vzdálenosti párování), minimální vzdálenosti mezi sebou, kde je práh nastaven na 120 pixelů. Výsledky testování viz Tabulka D.12 a výsledné umístění budovy viz Obrázek D.12
5. Výběr nejlepších (dle podobnosti/vzdálenosti párování) a RANSAC⁸. Výsledky testování viz Tabulka 5.1 a výsledné umístění budovy viz Obrázek 5.2

Minimální vzdálenost bodů se lišila pro každý snímek budovy. Nejvíce záleželo, aby budovy tvořily přibližně stejnou plochu vstupního snímku, aby mohla být nastavena globální konstanta. Předpokládá se, že právě tato konstanta se bude vypočítávat relativně, např. na základě vzdálenosti od budovy.

U výsledků tohoto testu je možné vidět, že už pro některé budovy nebylo možné najít čtyři dobré klíčové body v dostatečné vzdálenosti, pokud byly nalezeny, tak často dosahovali špatných/nepoužitelných výsledků. Nejlepších výsledků bylo dosaženo, pokud vybrané klíčové body tvořily obdélník. Pokud by např. 3–4 klíčové body ležely přímce, tak výsledné umístění stále obsahovalo velké nepřesnosti. Tyto nepřesnosti je možné eliminovat pomocí metody RANSAC, která dosáhla dobrých výsledků i u snímků, které obsahovali mnoho špatných párů.

⁸Metoda RANSAC byla přidána těsně před odevzdáním

Závěr

Cílem této bakalářské práce bylo vytvoření prototypu na základě analýzy s ohledem na omezení projektem Věnná města českých královen. Analýza obsahuje možnosti předzpracování obrazu, výpočtu příznaků a jejich následného párování. Prototyp simuluje hlavní funkcionalitu mobilní aplikace pro Android, na rozšířenou realitu, kde ve snímku scény rozpozná a umístí obraz rozpoznané budovy zpět do scény.

Pro implementaci prototypu byl použitý jazyk Python společně s knihovnamy pro práci s obrazem, např. OpenCV. Prototyp byl následně otestován na vytvořeném testovacím datasetu. Testování potvrdilo teorii i ukázalo možné problémy, které byly popsány již v analytické části. Některé budovy ukázaly velkou stabilitu i s různými parametry. Nejlepších výsledků bylo dosaženo s použitím metody RANSAC pro eliminaci chybných spojení.

Cíle této práce byly splněny. Práce tvoří první část většího projektu a její výsledky budou použity v minimálně dalších dvou bakalářských pracích. Do budoucna lze uvažovat i o používání rozpoznávacího jádra pro jiné aplikace než je projekt Věnná města českých královen.

Bibliografie

1. BOW, Sing T. *Pattern recognition and image preprocessing* [online]. CRC press, 2002 [cit. 2020-04-10]. ISBN 0-8247-0659-5.
2. COOK, John D. *Changing the contrast and brightness of an image* [online]. 2009 [cit. 2020-02-26]. Dostupné z: <https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/>.
3. *OpenCV*. Changing the contrast and brightness of an image [online]. 2020 [cit. 2020-02-20]. Dostupné z: https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html.
4. *Cambridge in colour*. Understanding gamma correction [online]. Cambridge, 2005 [cit. 2020-02-21]. Dostupné z: <https://www.cambridgeincolour.com/tutorials/gamma-correction.html>.
5. *Histogram Equalization* [online]. 2019 [cit. 2020-02-20]. Dostupné z: https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html.
6. Difference between contrast stretching and histogram equalization. In: *Stack Overflow* [online]. 2017 [cit. 2020-02-26]. Dostupné z: <https://stackoverflow.com/questions/41118808/difference-between-contrast-stretching-and-histogram-equalization>.
7. *Perspective Transformation* [online]. 2020 [cit. 2020-02-21]. Dostupné z: https://www.tutorialspoint.com/dip/perspective_transformation.htm.
8. SHAH, Dr. Mubarak. Interest point detection [přednáška]. In: *Youtube* [online]. Orlando, 2012 [cit. 2020-02-28]. Dostupné z: https://www.youtube.com/watch?v=_qgKQGsuKeQ.

9. KOLLER, Daphne; SAHAMI, Mehran. Toward Optimal Feature selection [online]. In: *Stanford Digital Libraries* [online]. Stanford, 2005 [cit. 2020-02-28]. Dostupné z: <https://ai.stanford.edu/~koller/Papers/Koller+Sahami:ICML96.pdf>.
10. MORDVINTSEV, Alexander. *OpenCV*. Understanding Features [online]. 2013 [cit. 2020-02-24]. Dostupné z: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning.
11. What is a feature descriptor in image processing (algorithm or description)? In: [online]. 2015 [cit. 2020-02-24]. Dostupné z: <https://stackoverflow.com/questions/27595455/what-is-a-feature-descriptor-in-image-processing-algorithm-or-description>.
12. GEVERS, T.; SMEULDERS, Arnold. PicToSeek: Combining color and shape invariant features for image retrieval. *Image Processing, IEEE Transactions on* [online]. 2000, roč. 9, s. 102–119 [cit. 2020-05-03]. Dostupné z DOI: 10.1109/83.817602.
13. BROWN, Matthew; LOWE, David. Invariant Features from Interest Point Groups. In: [online]. 2002, sv. 13 [cit. 2020-05-03]. Dostupné z DOI: 10.5244/C.16.23.
14. DANIELSSON, Per-Erik. Euclidean Distance Mapping. *Graphical Models /graphical Models and Image Processing /computer Vision, Graphics, and Image Processing - CVGIP* [online]. 1980, roč. 14, s. 227–248 [cit. 2020-03-04]. Dostupné z DOI: 10.1016/0146-664X(80)90054-4.
15. OJALA, Timo; PIETIKÄINEN, matti; HARDWOOD, David. A comparative study of texture measures with classification based on featured distributions. In: [online]. 1996, sv. 29, s. 51–59 [cit. 2020-02-20]. Dostupné z DOI: 10.1016/0031-3203(95)00067-4.
16. AHONEN, Timo; HADID, Abdenour; PIETIKÄINEN, Matti. Face Recognition with Local Binary Patterns. In: [online]. 2004, sv. 3021, s. 469–481 [cit. 2020-02-18]. Dostupné z DOI: 10.1007/978-3-540-24670-1_36.
17. KOBAYASHI, Takuya; HIDAKA, Akinori; KURITA, Takio. Selection of Histograms of Oriented Gradients Features for Pedestrian Detection. In: [online]. 2007, s. 598–607 [cit. 2020-02-20]. Dostupné z DOI: 10.1007/978-3-540-69162-4_62.
18. HARALICK, Robert; SHANMUGAM, K.; DINSTEN, Ih. Textural Features for Image Classification. *IEEE Trans Syst Man Cybern* [online]. 1973, roč. SMC-3, s. 610–621 [cit. 2020-02-26]. Dostupné z: <http://haralick.org/journals/TexturalFeatures.pdf>.

19. RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. ORB: an efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision* [online]. 2011, s. 2564–2571 [cit. 2020-02-21]. Dostupné z DOI: 10.1109/ICCV.2011.6126544.
20. LOWE, David G. Object recognition from local scale-invariant features. In: *Proceedings of the seventh IEEE international conference on computer vision* [online]. 1999, sv. 2, s. 1150–1157 [cit. 2020-03-23]. ISBN 0-7695-0164-8. Dostupné z DOI: 10.1109/ICCV.1999.790410.
21. LOWE, David G. *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image* [online]. The University of British Columbia, 2004 [cit. 2020-04-28]. US6711293B1.
22. LOWE, David G. SIFT-The scale invariant feature transform. *Int. J.* [online]. 2004, roč. 2, s. 91–110 [cit. 2020-02-20]. Dostupné z: https://robo.fish/wiki/images/5/58/Image_Features_From_Scale_Invariant_Keypoints_Lowe_2004.pdf.
23. BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. SURF: Speeded up robust features. *Computer Vision-ECCV 2006* [online]. 2006, roč. 3951, s. 404–417 [cit. 2020-02-20]. Dostupné z DOI: 10.1007/11744023_32.
24. OKANAGAWA, Funayama; YANAGIHARA, Hiromichi; VAN GOOL, Luc; BAY, Herbert. *Robust interest point detector and descriptor* [online]. 2009 [cit. 2020-02-21]. US 2009/0238460A1.
25. HARRIS, C.; STEPHENS, M. A Combined Corner and Edge Detector. *Proceedings 4th Alvey Vision Conference* [online]. 1988, roč. 1988, s. 147–151 [cit. 2020-02-18]. Dostupné z DOI: 10.5244/C.2.23.
26. MIKOLAJCZYK, Krystian; TUYTELAARS, Tinne; SCHMID, Cordelia; ZISSERMAN, Andrew; MATAS, Jiri; SCHAFFALITZKY, Frederik; KADIR, Timor; VAN GOOL, Luc. A Comparison of Affine Region Detectors. *International Journal of Computer Vision* [online]. 2005, roč. 65, s. 43–72 [cit. 2020-02-18]. Dostupné z DOI: 10.1007/s11263-005-3848-x.
27. *OpenCV*. Introduction to SURF (Speeded-Up Robust Features) [online]. 2020 [cit. 2020-03-19]. Dostupné z: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html.
28. *OpenCV*. Feature Matching [online]. 2020 [cit. 2020-03-04]. Dostupné z: https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html.

29. MUJA, Marius; LOWE, David. Fast Matching of Binary Features. In: [online]. 2012, s. 404–410 [cit. 2020-03-04]. ISBN 978-1-4673-1271-4. Dostupné z DOI: 10.1109/CRV.2012.60.
30. *OpenCV*. Feature Matching with FLANN [online]. 2020 [cit. 2020-03-04]. Dostupné z: https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html.
31. FISCHLER, R.; BOLLES, M. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun ACM* [online]. 1981, roč. 24, s. 619–638 [cit. 2020-05-25]. Dostupné z DOI: 10.1145/358669.358692.
32. KARAMI, Ebrahim; PRASAD, Siva; SHEHATA, Mohamed. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. In: [online]. 2015 [cit. 2020-02-26]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf>.
33. TECHNICAL STANDARDIZATION COMMITTEE ON AV & IT STORAGE SYSTEMS AND EQUIPMENT. Exchangeable image file format for digital still cameras: Exif Version 2.2. *Japan Electronics and Information Technology Industries Association* [online]. 2002 [cit. 2020-03-23]. Dostupné z: <https://www.exif.org/Exif2-2.PDF>.
34. KUMAR, Apu. *Lot a data*. How precisely accurate is your geo-intelligence? [online]. 2017 [cit. 2020-02-12]. Dostupné z: <https://lotadata.com/blog/how-precisely-accurate-is-your-geo-intelligence/>.
35. OPENCV TEAM. *OpenCV*. OpenCV [online]. 2020 [cit. 2020-04-10]. Dostupné z: <https://opencv.org/>.
36. NUMPY DEVELOPERS. *NumPy*. NumPy [online]. 2020 [cit. 2020-04-10]. Dostupné z: <https://numpy.org/>.
37. LUNDH, Fredrik; CLARK, Alex; GITHUB CONTRIBUTORS. *Pillow*. Pillow [online]. 2020 [cit. 2020-04-10]. Dostupné z: <https://python-pillow.org/>.
38. MITCHEL, Justin. *Get Latitude and Longitude from EXIF using PIL* [online]. 2011 [cit. 2020-02-12]. Dostupné z: <https://gist.github.com/erans/983821>. [skript].

Seznam použitých zkratk

CLAHE Contrast limited adaptive histogram equalization

LBP Local binary patterns)

HOG Histogram of oriented gradients

GLCM Gray levelco-occurrence matrix

ORB Oriented FAST and Rotated BRIEF

FAST Features from accelerated segment test

BRIEF Features from accelerated segment test

SIFT Scale invariant feature transform

DOG Difference of Gaussian kernel

LOG Laplacian of Gaussian

SURF Speeded up robust features

BF Brute force

FLANN Fast Library for Approximate Nearest Neighbors

RANSAC Random Sample Consensus

EXIF Exchangeable image file format

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├── impl.....	zdrojové kódy implementace
│ ├── data.....	dataset budov
│ └── thesis.....	zdrojová forma práce ve formátu \LaTeX
text	text práce
└── thesis.pdf.....	text práce ve formátu PDF

Seznam budov

Většina budov byla nafocena během zataženého počasí. Při tvorbě datasetu proběhlo focení na třech místech: v Praze, Liberci a Chrastavě. Budovy č. 4 a 9 mají schválně špatné fotky v datasetu, aby demonstrovali chyby, kterých se vyvarovat při přidávání nové budovy. Budova č. 6 není obsažena v datasetu, protože nebylo možné vytvořit dataset z prvního focení. Mezitím byly přidány další budovy, tak bylo zachováno původní číslování.

Budova č.1

Název: Dům U Kamenného zvonu

Lokalita: Praha - Staroměstské náměstí

Fotografie budovy byla pořízena za dobrých světelných podmínek. Jedná se o jednu z prvních budov z datasetu, proto obsahuje i snímky z blízka a nemá dostatečný počet snímků z různých úhlů.



Budova č.2

Název: Dům U Rotta

Lokalita: Praha - Malé náměstí

Budova neobsahuje vstupní snímek. Před budovou je kašna, které zakrývá spodní část budovy (není na snímcích). Některé snímky v datasetu jsou méně kontrastní a zaniká na nich freska na celé přední části budovy.



Budova č.3

Název: U Zlaté konvice

Lokalita: Praha - Staroměstské náměstí

Budova neobsahuje vstupní snímek. Dataset obsahuje téměř všechny snímky z blízkosti a neobsahuje snímky z více úhlů. Špatné světelné podmínky v podloubí nejsou úplně optimální.



Budova č.4

Název: Štorchův dům

Lokalita: Praha - Staroměstské náměstí

Tato budova obsahuje téměř všechny snímky z velké blízkosti, obsahuje fotky pouze částí domu a slouží jako dobrá ukázka špatného datasetu. Snímky v datasetu by mohli být více kontrastní, aby vynikly výrazné prvky budovy.



Budova č.5

Název: Pražská městská pojišťovna

Lokalita: Praha - Staroměstské náměstí

Jedná se o jednu z prvních budov z datasetu, proto obsahuje i snímky z blízka a nemá dostatečný počet snímků z různých úhlů a vzdáleností. Snímky domu byly pořízeny za zataženého počasí.



Budova č.7

Název: Zelený rodinný dům

Lokalita: Liberec - Perštýn

Zelený dům byl vybrán jako zástupce rodinných/klasických domů, které nemají výrazně zdobenou omítku apod. Na vstupním snímku plot a značka zakrývá kus domu. V tomto případě stíny nejsou tak ostré, jako u jiných snímků domů.



Budova č.8

Název: Liberecká radnice

Lokalita: Liberec - Náměstí Edvarda Beneše

Snímky budovy byly pořízeny až v pozdních hodinách za slunečného dne, kde jsou na zemi tzv. ostré stíny a slunce osvětluje silně celou přední část budovy a pravá stěna je ve stínu. Budova radnice je významně větší než ostatní budovy v datasetu.



Budova č.9

Název: Residence Royal

Lokalita: Liberec - Lidové sady

Snímky budovy byly pořízeny až v pozdních hodinách za slunečného dne, kde jsou na zemi tzv. ostré stíny a slunce osvětluje silně celou přední část budovy. Vstupní snímek obsahuje značky a podpůrnou konstrukci budovy naproti, která zakrývá část domu.



Budova č.10

Název: Chrastavská radnice

Lokalita: Chrastava - Náměstí 1. máje

Fotografie budovy byly pořízeny za dobrých světelných podmínek. Na vstupním snímku je budova lehce zakryta stromy.



Budova č.11

Název: Dřevěný rodinný dům

Lokalita: Chrastava

Fotografie budovy byly pořízeny za dobrých světelných podmínek. Vstupní snímek obsahuje rušivý element v podobě jehličí. Samotná budova je téměř jednolitě hnědá s výraznými okny, ale neobsahuje bohatě zdobenou omítku/stěny.



Budova č.12

Název: Podstávkový dům

Lokalita: Chrastava

Fotografie budovy byly pořízeny za horších světelných podmínek, kde je bílá omítka budovy přexponovaná. Vstupní snímek obsahuje tmavé zábradlí, kde může být vypočítáno mnoho silných příznaků. Lokalita budovy nedovoluje focení z blízka, proto jsou snímky z datasetu relativně daleko a z méně úhlů.



Výsledky testování

V této příloze jsou všechny tabulky výsledků z testování a obrázků na předzpracování obrazu, selekce dobrých příznaků a minimální vzdálenost mezi klíčovými body, pro zvýšení přesnosti umístění snímku do scény.

Tabulka D.1: Výsledky testu bez úpravy histogramu

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	N	A	A	N	A	N	A
Přesnost	4	X	X	2	4	X	2	X	3

Tabulka D.2: Výsledky testu s úpravou histogramu

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	N	A	A	N	N	N	N
Přesnost	3	X	X	4	2	X	X	X	X

Tabulka D.3: Výsledky testu s adaptivní úpravou histogramu

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	N	A	A	A
Přesnost	2	X	1	1	2	X	2	1	3

D. VÝSLEDKY TESTOVÁNÍ



Obrázek D.1: Výsledné umístění budov bez úprav histogramu (pravý, levý): (1, 8), (7, 10), 12



Obrázek D.2: Výsledné umístění budov s ekvalizací histogramu: (pravý, levý): (1, 8), 7

Tabulka D.4: Výsledky testu s nastaveným poměrovým testem na 0,4

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	N	N	A	A	A	A	A	A
Umístění	A	X	X	A	A	X	A	X	X
Přesnost	2	X	X	3	2	X	1	X	X



Obrázek D.3: Výsledné umístění budov s adaptivní ekvalizací histogramu (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8

Tabulka D.5: Výsledky testu s nastaveným poměrovým testem na 0,5

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	X	A	X	A
Přesnost	2	X	1	2	2	X	2	X	3

Tabulka D.6: Výsledky testu s nastaveným poměrovým testem na 0,6

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	N	A	A	A
Přesnost	2	X	1	1	2	X	2	1	3

D. VÝSLEDKY TESTOVÁNÍ

Tabulka D.7: Výsledky testu s nastaveným poměrovým testem na 0,7

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	N	N	A	N	A	X	A
Přesnost	2	X	X	X	2	X	2	X	3

Tabulka D.8: Výsledky testu s nastaveným poměrovým testem na 0,8

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	N	A	A	N	A	N	A
Přesnost	2	X	X	2	2	X	1	X	3



Obrázek D.4: Výsledné umístění budov pro poměrový test s hodnotou prahu 0,4. Budovy na obrázku ze shora: 1, 7, 8, 10

Tabulka D.9: Výsledky testu s výběrem nejlepších bodů čistě podle vzdálenosti

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	N	A	A	A
Přesnost	4	X	4	4	1	X	1	1	4

Tabulka D.10: Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a minimální vzdáleností, kde práh je nastaven na 80 pixelů

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	N	A	A	N
Přesnost	2	X	1	2	2	X	2	1	X



Obrázek D.5: Výsledné umístění budov pro poměrový test s hodnotou prahu 0,5. Budovy na obrázku ze shora (pravý, levý): (1, 8), (4, 10), (5, 12), 7

Tabulka D.11: Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a minimální vzdáleností, kde práh je nastaven na 100 pixelů

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	A	N	A	A	A	N	A	A	A
Přesnost	2	X	1	1	2	X	1	1	3

Tabulka D.12: Výsledky testu s výběrem nejlepších bodů podle vzdálenosti a minimální vzdáleností, kde práh je nastaven na 120 pixelů

Budova	1	4	5	7	8	9	10	11	12
Rozpoznání	A	A	A	A	A	A	A	A	A
Umístění	N	N	N	A	A	N	A	X	A
Přesnost	X	X	X	1	4	X	3	X	4

D. VÝSLEDKY TESTOVÁNÍ



Obrázek D.6: Výsledné umístění budov pro poměrový test s hodnotou prahu 0,6. Budovy na obrázku ze shora (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8



Obrázek D.7: Výsledné umístění budov pro poměrový test s hodnotou prahu 0,7. Budovy na obrázku ze shora (pravý, levý): (1, 10), (8, 12)



Obrázek D.8: Výsledné umístění budov pro poměrový test s hodnotou prahu 0,8. Budovy na obrázku ze shora (pravý, levý): (1, 10), (7, 12), 8



Obrázek D.9: Výsledné umístění budov pro čtyři nejlepší klíčové body. Budovy na obrázku ze shora (pravý, levý): (1, 8), (5, 10), (7, 11), 12

D. VÝSLEDKY TESTOVÁNÍ



Obrázek D.10: Výsledné umístění budov s minimální vzdáleností 80 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (1, 8), (5, 10), (7, 11)



Obrázek D.11: Výsledné umístění budov s minimální vzdáleností 100 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (1, 9), (4, 10), (5, 11), (7,12), 8



Obrázek D.12: Výsledné umístění budov s minimální vzdáleností 120 pixelů mezi klíčovými body. Budovy na obrázku ze shora (pravý, levý): (7, 10), (8, 12)