VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ ZÁJMOVÝCH OBJEKTŮ V OBRAZOVÉ SADĚ

SEARCH OF OBJECTS OF INTEREST IN THE IMAGE SET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JURAJ MEDVEC
AUTHOR

VEDOUCÍ PRÁCE — Ing. BERAN VÍTĚZSLAV, Ph.D.
SUPERVISOR

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Medvec Juraj**
Obor: Informační technologie
Téma: **Vyhledávání zájmových objektů v obrazové sadě**
Object Instance Search in Image Dataset
Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte metody pro získání popisu významných oblastí v obraze a jejich porovnání, kvantizace a vyhledávání.
2. Vyberte vhodné metody a navrhněte systém, který vytvoří obrazové příznaky pro sadu obrazů tak, aby se v této sadě daly efektivně vyhledávat obsahově podobné obrazy.
3. Připravte a anotujte sadu obrazů pro vývoj a testování řešení.
4. Implementujte knihovnu funkcí a navržený vyhledávací systém s využitím dostupných knihoven a technologií.
5. Proveďte testy na přesnost, rychlost a využitelnost řešení. Výsledky diskutujte.
6. Vytvořte plakát a krátké demonstrační video reprezentující Vaše řešení.

Literatura:

- M. Sonka, V. Hlaváč, R. Boyle. *Image Processing, Analysis, and Machine Vision*, CL-Engineering, ISBN-13: 978-0495082521, 2007.
- Gary R. Bradski, Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*, ISBN 10: 0-596-51613-4, September 2008.
- J. Sivic, A. Zisserman. *Video Google: A text retrieval approach to object matching in videos,* 2013.
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2, 3 a částečně 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Táto práca je zameraná na vyhľadávanie objektu/scény v sade obrazov. Na úvod sú čitatelovi predstavené existujúce riešena, ktoré sa už aj používajú v praxi. Ďalšie kapitoly sú zamerané na vytvorený framework a popis práce na ňom. Na záver je predstavená demo aplikácia v ktorej bol navrhnutý framework pužitý, pomocou ktorého aplikácia vyhľadáva knihy na základe fotky ich prednej strany.

Abstract

This project is focused on the selected techniques for image searching. In the introduction are presented all existing solutions, which are used in daily use. The other chapters are focused on creating resulting framework. On the end is introduced the demo application, which is using the framework for searching books by front page image.

Kľúčové slová

Vizuálny slovník, SIFT, SURF, Vyhľadávaní obrazu, Obrazové Príznaky, Bag of Words

Keywords

Visual Dictionary, SIFT, SURF, Image Search, Image Featuers, Bag of Words

Citácia

MEDVEC, Juraj. *Vyhledávání zájmových objektů v obrazové sadě.* Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Beran Vítězslav, Ph.D.

Vyhledávání zájmových objektů v obrazové sadě

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pana Ing. Vítězslava Berana, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

. . . . . .
Juraj Medvec
16. mája 2017

Poďakovanie

Chcel by som poďakovať Ing. Vítězslavovi Beranovi, Ph.D za jeho pomoc a za to, že mi pomohol pri štúdiu potrebnej literatúry pre vypracovanie tejto práce. Bez neho by nebolo možné dosiahnúť takého výsledku aký sa mi podarilo dosiahnúť.

Obsah

Kapitola 1

Úvod

Cieľom tejto práce bolo navrhnutie frameworku, ktorý vyhľadáva scénu alebo objekt zobrazený na obrázku v dátovej sade orbazov. Navrhovaný framework by mal dokázať nájsť konkrétnu časť obrazu a to pomocou známych techník pre vyhľadávanie pomocou obrazu. Samotná myšlienka vyhľadávania pomocou obrazu sa objavila prvýkrát v roku 1992 pod pojmom „content-based image retrieval“, čo v preklade znamená vyhľadávanie obrázkov pomocou ich obsahu. S touto myšlienkou prišiel T. Kato, ktorý ju definoval ako vyhľadávanie obsahu obrazu pomocou hrán a farieb. Postupom času došlo k vyvynutiu ďalších metód pre vyhľadávanie obrazu, ktoré sa líšia či už spôbom spracovania obrazu, alebo v práci s týmito informáciami.

Pre riešenie mojej úlohy exituje veľa techník, pomocou ktorých je možné daný framework navrhnúť. Ja som sa rozhodol použiť jednu zo známych metód pre vyhľadávanie alebo kategorizáciu obrazu a to vyhľadávanie pomocou bag of words. Táto technika bola použitá pri vyriešení niekoľkých problémov v oblasti počítačovej grafiky ako napríklad objektová kategorizácia. Samotný navrhovaný framework je možné rozdeliť do niekoľkých logických častí, na časť ktorá sa zaoberá spracovaním obrazu, vytvorením bag of words a vizuálneho slovníka a na priestorovú verifikáciu.

Práca sa zaoberá problematikou vyhľadávania obrazu a spôsobmi, ktoré sa používajú pri riešení konkrétnych problémov. Z dôvodu, že som sa rozhodol vytvoriť framework pre vyhľadávanie obrazu pomocou metódy bag of words, tak sú v práci spomenuté problémy, ktoré sa vyskytujú pri práci s touto technológiou. V každej kapitole je čitateľovi predstavená teória, ktorá je zameraná na konkrétnu časť algoritmu a ako sa daná teória používa v praxi. Následne je na konci každej kapitoly uvedené, ako je daná problematika zahrnutá v navrhovanom frameworku. Po vytvorení vyhľadávacieho frameworku som sa rozhodol vytvoriť demo aplikáciu, v ktorej by bolo možné daný vyhľadávaci nástroj použiť. V závere práce sú popísané testy, ktoré prebehli na navrhovanom frameworku, kde je vysvetlené, prečo sa jednotlivé testy vykonali a ďalej je uvedená podrobnejšia analýza týchto testov.

Kapitola 2

Existujúce frameworky

Od začiatku devedesiatych rokov je vyhľadávanie obrazu na základe jeho obsahu veľmi aktívne skúmaná oblasť. Veľa vyhľadávacích systémov, ako komerčné tak aj vedecké, boli vytvorené [19]. Väčšina vyhľadávacích systémov postavených na základe vyhľadávania pomocou obsahu obrazu podporujú jednu alebo viac možností vyhľadávania:

- náhodné prehľadávanie
- vyhľadávanie pomocou príkladu
- vyhľadávanie pomocou skice
- vyhľadávanie pomocou textu
- navigácia s prispôsobenými obrazovými kategóriami

V dnešnej dobe je možné vidieť bohatú sadu možností vyhľadávania. Teraz si predstavíme časť týchto riešení.

2.1 QBIC

Patrí medzi prvé komerčné vyhľadávacie algoritmy, ktoré vyhľadávajú na základe obsahu obrazu. Funkcionalitu QBIC (Query by image content) je možné rozdeliť do dvoch komponentov, na spracovanie databázy a databázový dotaz. Počas spracovania databázy sú obrázky, videá spracované pomocou extrahovania bodov záujmu, ktoré popisujú ich osah ako farbu, textúru, ostrosť a pri kamere aj pohyb objektov [15]. Následne sú extrahované body záujmu uložené v databáze. Vo fáze dotazu na databázu sa užívateľ dotazuje graficky, to znamená, že body záujmu sú extrahované z dotazovaného obrazu a sú vstupom pre porovnávací nástroj, ktorý nájde obrázky alebo videá s podobnými bodami záujmu.

2.2 Virage

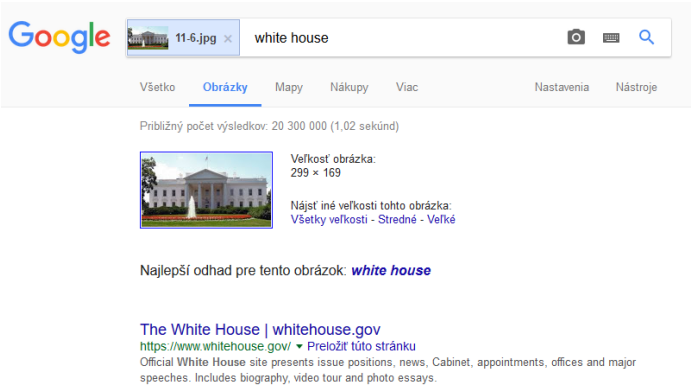
Jedná sa o algoritmus pre vyhľadávanie obrazov pomocou ich obsahu, ktorý bol vyvinutý firmou Virage Inc. Je podobný QBIC, takže Virage podporuje taktiež vizuálne dotazy založené na farbe, kompozícii, textúre a štruktúre obrazu. Virage ale zachádza o trochu ďalej ako QBIC, a to tak, že podporuje váhovanie vyššie uvedených dotazov. Samotné váhovanie záleží na užívatelovi.

2.3 Photobook

Photobook je set interaktívnych nástrojov pre vyhľadávanie obrázkov vyvynutý na MIT. Pohtobook pozostáva z troch podkníh, z ktorých je príslušne extrahovaná ostrosť, textúra a tvárove body záujmu. Užívateľ potom môže vytvárať dotaz založený na príslušných príznakoch v každej z jednej podkníh.

2.4 Google Image Search

V roku 2001 uviedla firma Google [1] na trh framework, ktorý umožnuje svojím užívateľom vyhľadávať obrázky. Kľúčovými bodmi pre vyhľadávanie boli meno súboru, text odkazu na obraz a text vedľa obrazu. V tej dobe nešlo o vyhľadávanie pomocou obrazu, ale to čo vtedy vzniklo sa stalo základným kameňom dnešného vyhľadávacieho frameworku.



Obr. 2.1: Obrázok zobrazujúci výsledok vyhľadávania pomocou Google Image Search. V zobrazovanom výsledku sa nachádzajú viaceré obrazy s podobným obsahom.

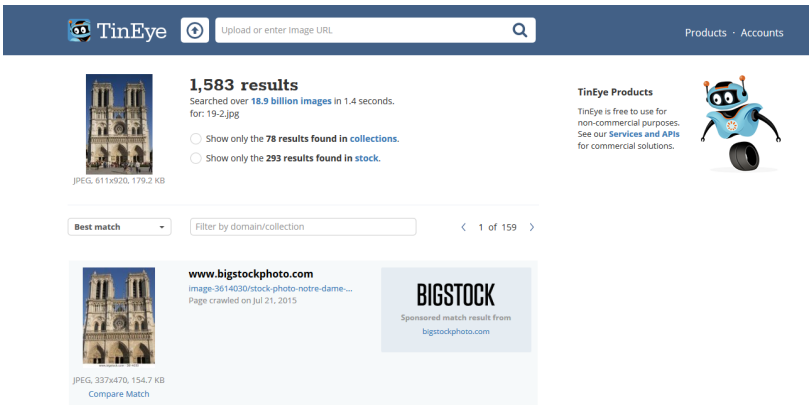
Užívateľia boli s týmto nástrojom spokojní a počet indexovaných obrázkov rástol. Z toho dôvodu sa Google v roku 2007 rozhodol priniesť novú možnosť uživateľom, a to vyhľadávanie pomocou obrazu. V dnešnej dobe tento framework používa pre vyhľadávanie podobných obrazov techniku, ktorá sa nazýva „reverse image search“. Táto technika funguje na princípe vytvorenia takzvaného „image qeury“, pomocou ktorého je daný obraz vyhľadaný. Na základe týchto dotazov je možné veľmi efektívne vyhľadávať vo veľkých referenčných sadách obrazov, čo je v konečnom dôsledku veľmi dôležité, keďže práve rýchlosť vyhľadávania a jeho presnoť sú základné piliere dobrého vyhľadávacieho nástroja. Google Image Search má oproti ostatým vyhľadávacim nástrojom tú výhodu, že disponuje obrovskou databázou obrázkov. Dokáže užívateľovi poskytnúť pri väčšine vyhľadávaní sadu dobrých výsledkov.

2.5 TinEye

TinEye [2] je vyhľadávací framework, korý sa taktiež ako Google Image Search radí medzi nástroje, ktoré sú postavené na metóde vyhľadávania s názvom „reverse image search“. Radí sa medzi prvé webové vyhľadávacie nástroje, ktoré namiesto používania bodov záujmu, metadat a pod. používa obrazovú identifikáciu. TinEye nerozpoznáva objekty alebo tváre,

[1] www.google.com
[2] https://www.tineye.com/

ale pozerá sa na obraz ako na celok. Dokáže spracovať typy obrazov ako JPEG, GIF, PNG ale nedokáže pracovať s formátom Adobe Flash [16].



Obr. 2.2: Ukážka vyhľadania obrazu pomocou framworku TinEye. Ako je možné vidieť na obráku, vo výsledku vyhľadávania sa nachádzajú viaceré výsledky s podobným obsahom.

Algoritmus pre porovnávanie je založený na vytváraní hashu pre každý jeden obraz. Výsledný hash je nezávislý na veľkosti obrazu, zaostrení obrazu či dokonca na zmene farieb.

Kapitola 3

Návrh vlastného frameworku

Ako bolo spomenuté v úvode, cieľom tejto práce je navrhnutie a vytvorenie vlastného vyhľadávacieho frameworku, ktorý bude vyhľadávať scény alebo objekty v obrazovej dátovej sade. V dnešnej dobe existuje niekoľko metód, ktoré sa zaoberajú touto problematikou. Tieto algoritmy je možné rozdeliť do dvoch hlavných skupín a to na CBIR (Content-based imahe retrieval) a na CBVIR (content-based visual information retrieval). Väčšina algoritmov, ktoré dnes sa radia do prvej menovanej skupiny, ako napríklad spomenutý Google Imase Search. Tieto algoritmy pracujú na princípe popisovania algoritmov pomocou reťazcov, ktoré boli vytvorené z bodov záujmu extrahovaných z obrázkov.

Medzi populárne metódy pre kategorizáciu a vyhľadávanie obrazu sa v dnešnej dobe radí metóda Bag of Words. Na internete je možné nájsť mnoho odborných článkov, ktoré popisujú použitie danej metódy v praxi. Jedná sa o veľmi zaujímavý spôsob spracovania obrazu, a preto som sa rozhodol použiť tento koncept pre navrhovaný framework. To znamená, že výsledný algoritmus je možné rozdeliť do niekoľkých logických častí ako spracovanie obrazu, tvorba vizuálneho slovníka a bag of words, vyhľadávanie kandidátov a priestorová verifikácia. Každá jednotlivá časť frameworku, je detailnejšie popísaná v nasledujúcich kapitolách.

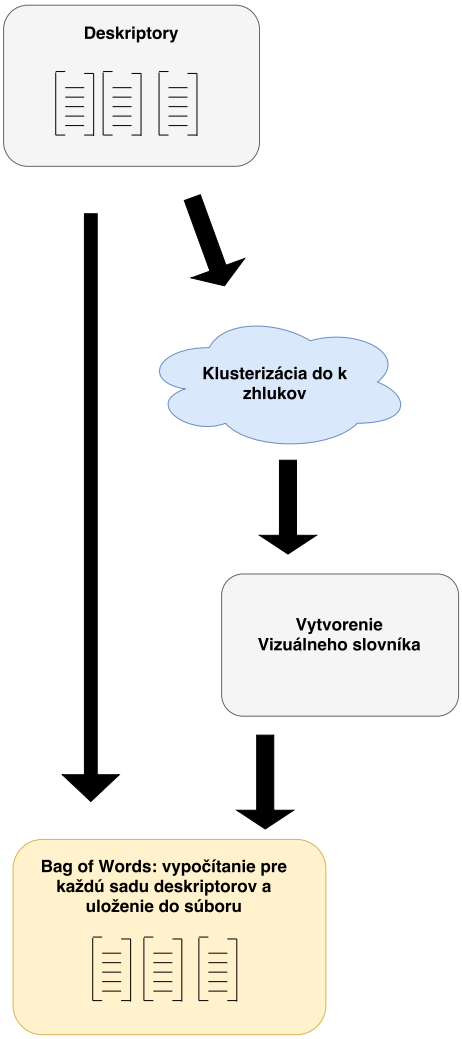
3.1 Spracovanie obrazov

Každý framework, ktorý je zameraný na vyhľadávanie obrazu potrebuje ako prvé načítať hľadaný obraz a taktiež obrazy, v ktorých bude vyhľadávať. Spracovanie či už obrazu z databázy, alebo vyhľadávaného obrazu je proces, ktorý prebieha rovnakým postupom ako je to znázornené na obrázku 3.1. Ako je možné vidieť, na vstupe sú spomínané obrazy a na výstupe tejto časti algoritmu sú deskriptory, ktoré sú nevyhnutné pre ďalšie výpočty. Predtým ako bude algoritmus schopný vypočítať deskriptory, je potrebné extrahovať body záujmu. To znamená, že celý proces spracovania obrazu v mnou navrhovanom frameworku je možné rozdeliť do troch krokov, tak ako je to možné vidieť na obrázku 3.1.

Obr. 3.1: Diagram, ktorý popisuje proces spracovania ako obrazu z databázy, taktiež hľadaného obrazu.

3.2 Tvorba slovníka

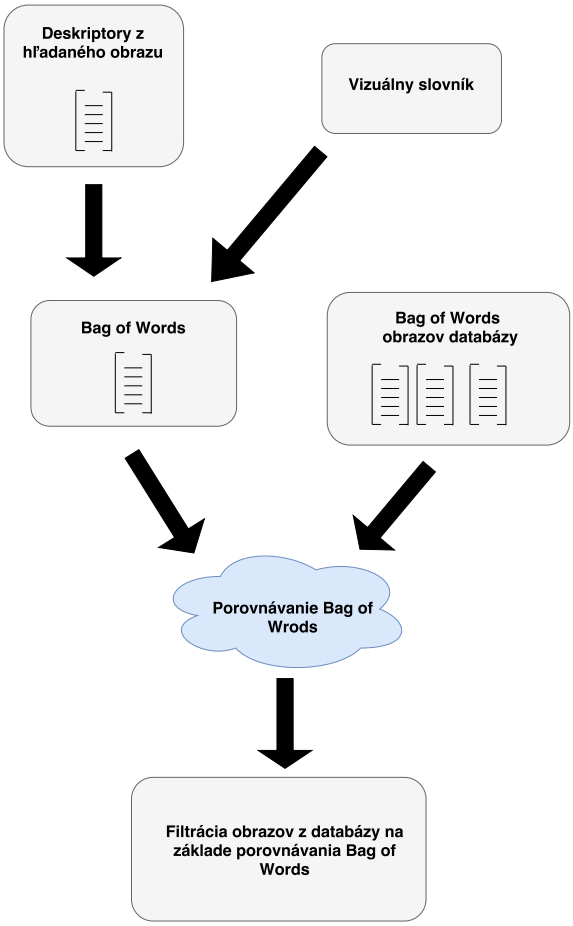
Ako som spomenul v úvode tejto kapitoly, rozhodol som sa vytvoriť vyhľadávací framework, ktorý bude postavený na vyhľadávaní pomocou bag of words. Na to, aby bol framework schopný vytvoriť spomínané histogramy, je potrebné vytvoriť vizuálny slovník z dát ktoré sú uložené v databáze. Na obrázku 3.2 môžete vidieť, že vizuálny slovník je možné vytvoriť z vypočítaných deskriptorov. Ďalej je potrebné klasterizovať deskriptory, z ktorých budem slovník vytvárať. Po úspešnej klasterizácii bude framework disponovať potrebným vizuálnym slovníkom, pomocou ktorého bude môcť dokázať vypočítať bag of words či už pre obrazy z databázy takiež aj pre vyhľadávaný obraz. Je dôležité spomenúť to, že slovník sa vytvára klasterizáciou výhradne deskriptorov, vypočítaných z obraz z databázy a k jeho opätovnému vypočítaniu dochádza len pri zmene dát v databáze.



Obr. 3.2: Diagram, ktorý popisuje proces tvorby vizuálneho slovníka a následne ako dochádza k vytváraniu bag of words pre obrazy z databázy.

3.3 Vyhľadávanie kandidátov

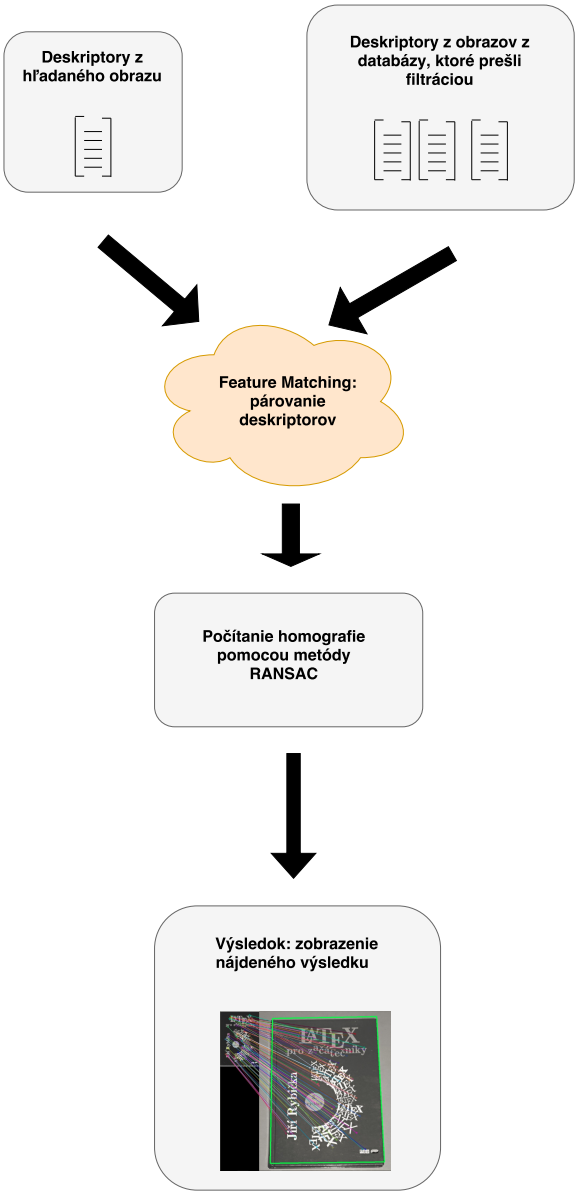
Po tom, čo vytvorí framework bag of words pre všetky obrazy z databázy je potrebné vytvoriť to isté aj pre hľadaný obraz a následne ich medzi sebou porovnať. Ja som sa rozhodol túto časť algoritmu dať do jedného celku, kedy bude dochádzať k vytvoreniu bag of words pre hľadaný obraz a následné porovnanie s bag of words obrazov z databázy. Celý tento postup je zobrazený na obrázku 3.3, kde je možné vidieť, že vizuálny slovník, ktorý bol vytvorený pomocou deksriptorov obrazov z databázy bude následne použitý pre vypočítanie bag of words pre hľadaný obraz. Po porovnaní potrebných histogramov dôjde k zoradeniu výsledkov od najlepšieho, kedy dôjde k filtrácii obrazov z databázy. Táto filtrácia znamená, že framework bude ďalej pracovať len s určitým počtom obrazov z databázy, ktoré budú vybraté na základe porovnania bag of words.



Obr. 3.3: Diagram, ktorý popisuje proces tvorby bag of wrods pre hľadány obraz a následne porovnanie bag of words hľadaného obrazu a obrazov z databázy.

3.4 Priestorová verifikácia

Priestorová verifikácia bude posledná časť algoritmu. V tejto časti bude dochádzať k porovnaniu deskriptorov hľadaného obrazu a všetkých vybratých obrazov z databázy. Samotné porovnanie nedokáže určiť, či sú dané obrazy podobné alebo zobrazujú podobnú scénu či objekt. Podobnosť obrazu sa bude v navrhovanom frameworku počítať pomocou homografie. Homografia bude mať za úlohu zistiť, ktoré časti obrazov sa zhodujú. Celý priebeh časti frameworku s názvom priestorová verifikácia je zobrazený na obrázku 3.4, kde je možné vidieť, čo je vstup a výstup jednotlivých častí a ako na sebe jednotlivé časti závisia.



Obr. 3.4: Diagram, ktorý popisuje proces priestorovej verifikácie, kedy dochádza k párovaniu deskriptorov hľadaného obrazu a obrazu z databázy. Následne po párovní týchto deskriptorov dochádza k nájdeniu homografie medzi týmito dvoma obrazmi a konečné vyhodnotenie, či zobrazujú rovnakú scénu či objekt alebo nie.

Kapitola 4

Spracovanie obrazu

Pri vyhľadávaní pomocou fotografií je veľmi dôležité sa nepozerať na obraz ako na celok ale ako na počet bodov, z ktorých sú niektoré viac alebo menej špecifické pre daný obraz. V odbornej literatúre je možné sa stretnúť aj s názvami ako body záujmu (keypoints) alebo príznaky (features). Tento pojem je možné definovať ako zaujímavú časť obrazu, ktorá by mala byť určitým spôsobom špecifická pre daný obraz [17].

4.1 Body záujmu

Príznaky sú základným prvkom mnohých algoritmov v oblasti počítačovej grafiky. Detekovanie príznakov je nízkourovňový proces spracovania obrazu a zväčša k nemu dochádza na začiatku algoritmov [14]. Body záujmu nie sú vhodné pre samotné porovnávanie, pretože reprezentujú len bod v obraze, to znamená, že sa jedná o ukazateľe na určité body v obraze ako je možné vidieť na obrázku 4.1.

Obr. 4.1: Ukážka bodov záujmu vykreslených priamo na obraze. Na obraze sú vykreslované ako farebné kruhy a ich pozícia je určená príslušnou metódou, ktorou boli detekované.

V počítačovej grafike bol koncept bodov záujmu alebo taktiež nazývané keypoints použitý na vyriešenie veľa problémov pri rozpoznávaní objektov. Pri práci s keypoints sa stretávame s myšlienkou práce s obrazom nie ako s celkom, ale ako prácou so špecialnými bodmi a vykonávaním výpočtov nad nimi. Tieto body zväčša predstavujú špecifické časti obrazu čo môžu byť napríklad hrany, rohy, škvrny alebo hrebene. To ktorý typ z bodu záujmu je vybraný, vždy závisí akú metódu algorimus používa. Pri extrahovaní bodov záujmu je možné použiť radu algoritmov, ktoré sa špecializujú len na určitý druh bodov. Zoznam týchto algoritmov je zobrazený v tabuľke 4.1 . Z tejto tabuľky je možné vyčítať, že nie všetky algoritmy pre detekciu bodov záujmu z obrazu sa zaoberajú všetkými typmi daných bodov. K najčastejšie používaným algritmom pre extrahovanie týchto bodov patrí Hessian Determinant. Tento spôsob je veľmi rýchly a účinný. Je možné sa s ním stretnúť pri práci s metódou SURF (Speede Up Robust Features).

Feature detector	**Edge**	**Corner**	**Blob**
Canny	**X**		
Sobel	**X**		
Kayyali	**X**		
Harris & Stephens	**X**	**X**	
SUSAN	**X**	**X**	
Shi & Tomasi		**X**	
FAST		**X**	**X**
Laplacian of Gaussian		**X**	**X**
Difference of Gaussian		**X**	**X**
Determinant of Hessian		**X**	**X**

Tabuľka 4.1: Detektory Bodov Záujmu a oblasti na ktoré sa zameriavajú

Pri porovnávaní príznakov extrahovaných z dvoch obrazov je možné naraziť na častý porblém ako zmena veľkosti objektu na jednom obraze oproti objektu na druhom obraze. To vzniká tak, že objekt, ktorý chceme nájsť môže byt odfotený z rôznych vzdialeností alebo aj pod rôznym uhlom. Ak sa potom snažíme porovnať dva obrazy používaním pevnej veľkosti okolia bodu záujmu, dochádza potom k nezhode kvôli zmenenej veľkosti objektu.

Tento problém je možné vyriešiť pomocou konceptu klúčových bodov obrazu nezávislých na zväčšení (scale-invariant features). Hlavná myšlienka tohto konceptu je mať faktor zmeny (scale-factor) spojený s každým jednotlivým detekovaným bodom záujmu. Medzi metódy, ktoré pracujú s týmto typom bodov záujmu sa radí napríklad SURF a SIFT (Scale-invariant feature transform) [5]. Tieto metódy sú určitým spôsobom odlišné, a z toho dôvodu vypočítavajú body záujmu nezávislé na zväčšení odlišným spôsobom. Zatiaľ čo pri metóde SURF sa tento typ bodov záujmu počíta pomocou determinantu Hessianovej matice, tak metóda SIFT pracuje na základe DoG (Difference of Gaussians) [11].
Rozdiel medzi bodmi záujmu, ktoré nie sú invariatné k zväčšeniu a rotácii, vidieť hneď na prvý pohľad. Keď sa detailnejšie pozrieme na obrázok 4.1 a obrázok 4.2 môžeme vidieť rozdiel v tom, že zatiaľ čo na obrázku 4.1 sú všetky body rovnakého priemeru, tak na obrázku 4.2 majú body záujmu rozdielny priemer. [14]

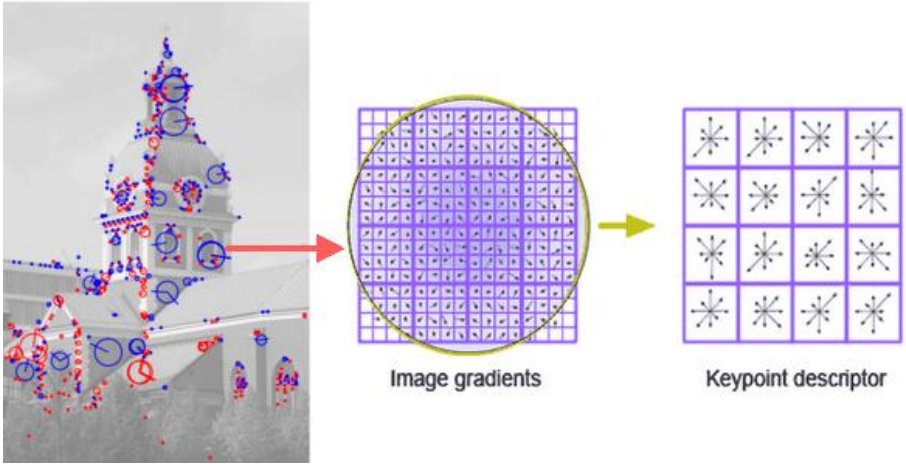
Obr. 4.2: Na obrázku je možné vidieť body záujmu, ktoré sú typu „scale-invariant“, čo je potvrdené tým, že veľkosti kruhov zobrazených na obraze nie sú rovnaké. Prevzaté z [6]

4.2 Deskriptory

Ako bolo spomenuté v predošlej podkapitole, tak pri práci s obrazom je veľmi dôležité extrahovať body, pomocou ktorých je možné daný obraz popisovať. Tieto body sa nazývajú body záujmu, ale nie je možné s nad nimi vykonávať potrebné operácie pre porovnávanie obrazu, pretože predsavujú len ukazovatele do obrazu. Deskriptory sú reprezentované ako n dimenzionálne vektory, ktoré popisujú elementárne charakteristiky obrazu ako napríklad hrany, farbu, textúru alebo pohyb. Pre každý jeden extrahovaný bod záujmu je vypočítaný deskriptor. Deskriptory sa považujú za prvý krok pri zisťovaní spojenia medzi pixelami obsiahnutými v digitálnom obraze [13]. Vizuálne deskriptory je možné rozdeliť do dvoch skupín:

- Deskriptory všeobecných informácií (General information descriptors) – Tento typ deskriptorov väčšinou popisuje elementárne vlastnosti ako napríklad farbu, textúru, ostrosť, lokáciu. S týmto typom deskriptorov sa môžeme stretnúť pri bežných algoritmov zameraných na spracovanie obrazu.

- Deskriptory špecifických informácií o doméne (Specific domain information descriptors) – Špecifické deskriptory obsahujú informáciu o objektoch, udalostiach v scéne, no ich nevýhodou je, že nie sú ľahko extrahovateľné. Používajú sa v aplikáciách, ktoré sú zamerané na rozpoznávanie tvári.



Obr. 4.3: Postup výpočtu deskriptorov z extrahovaných bodov záujmu, Na obrázku je znázorné extrahovanie bodov záujmu z obrazu a následné vypočítanie deskriptorov z extrahovaných bodov. Prevzaté z [2].

4.3 Známe techniky pre získanie bodov záujmu

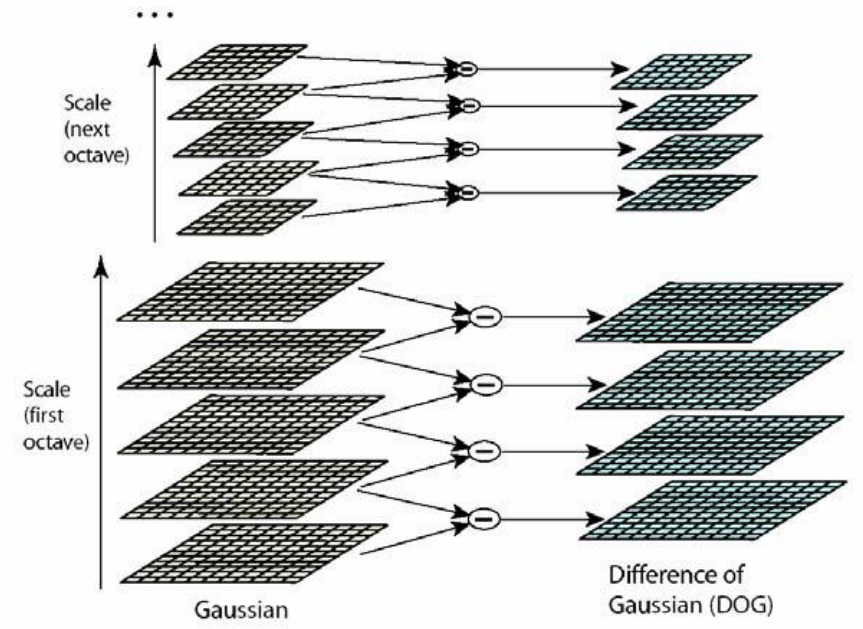
Extrahovaním a detekciou príznakov z obrazu sa v dnešnej dobe zaoberá veľa techník. Medzi tie najpoúživanejšie sa radia SIFT a SURF (Speeded-Up Robust Features). Obe tieto metódy sú implementované vo voľne dostupnej knižnici OpenCV.

SIFT

SIFT je algoritmus, ktorý sa zameriava na detekovanie a popisovanie lokálnych príznakov z obrazu. Tento algoritmus bol patentovaný v Spejných Štátoch Amerických na Univerzite v Britskej Kolumbii. V roku 1999 bol publikovaný Davidom Lowiem.

Detekovanie bodov záujmu v metódy SIFT prebieha pomocou počítania DoG (Difference of Gaussian). Jedná sa o použitie Gaussianovej pyramídy k tomu, aby bolo možné pre každý bod záujmu určiť veľkosť faktoru zväčšenia. DoG je počítaný z rozdielu matíc LoG (Laplacian of Gaussian), čo je aj vizuálne znázornené na obrázku 4.4. Vďaka tomu dokáže metóda pracovať s bodami záujmu, ktoré sú invariantné k zväčšeniu či rotácii [4].

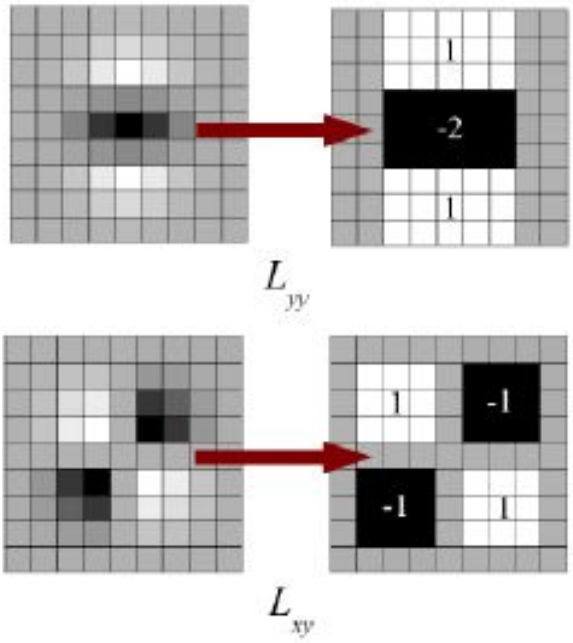
Popisovanie extrahovaných bodov záujmu prebieha v metóde SIFT tak, že okolo bodu záujmu je zobratá matica o veľkosti 16x16 [4]. Táto matica je rozdelená do podblokov o veľkosti 4x4. Pre každý podblok je vytvorený histogram s ôsmimi stĺpcami. To znamená, že v konečnom dôsledku pracuje metóda SIFT s 128 dimenzionálnymi vektormi.



Obr. 4.4: Postup výpočtu Difference of Gaussian. Prevzaté z [10].

SURF

SURF sa radí medzi metódy, ktoré vyhľadávajú a popisujú kľúčové body v obraze. Táto metóda vychádza z predchádzajúcej metódy SIFT. Pretože metóda SIFT bola celkom pomalá pri svojích výpočtoch a bolo potrebné tieto výpočty zrýchliť, tak v roku 2006 došlo v publikovaniu novej rýchlejšej metódy SURF. Bola publikovaná tromi autormi : Bay, H., Tuytelaars, T. and Van Gool, L. [11] Ako bolo spomenuté, tak SURF vychádza zo staršej metódy SIFT, ale čo sa týka počítania faktoru zväčšenia, tak SURF zachádza o niečo hlbšie a aproximuje LoG pomocou filtrov. Táto aproximácia je znázornená na obrázku 4.5. Tento princíp detekcie bodov záujmu je veľmi rýchly.



Obr. 4.5: Výpočet deskriptorov pre príslušné body záujmu pomocou metódy SURF. Prevzaté z [18].

SURF používa pre nájdenie bodov záujmu determinant Hessianovej matice. Tento determinant je použitý na vypočítanie lokálnej zmeny okolo bodov a následne sú vybraté také body, ktoré kde determinant dosahuje maxima.

4.4 Použitie vo výslednom Frameworku

Je predpokladom, že navrhovaný framework pre vyhľadávanie obrazu bude pracovať nad veľkym objemom dát. Tieto obrázky budú obsahovať rôzne objekty odfotené pod viacernými uhlami a z rôznej vdialenosti. Z toho vyplýva, že typ bodov záujmu použitých pre moje riešenie by mal byť invariantný k rotácii a zmene veľkosti obrazu. Preto som sa rozhodol použiť pre extrahovanie bodov záujmu a vypočítanie deskriptorov metódu SURF, ktorá je vhodná pre navrhované riešenie. Pre SURF som sa rozhodol na základe toho, že rovnako ako SIFT pracuje so bodmi záujmu, ktoré nie sú ovplyvňované zväčšením obrazu, ale táto metóda je vhodnejšia pre prácu s veľkým objemom dát, pretože dokáže spracovať veľké množstvo dát v pomerne kratšej dobe ako SIFT.

Implementáciu samotnéj metódy som v tomto projekte nemusel riešiť, keďže je voľne dostupna knižnica OpenCV, v ktorej sú implementované všetky potrebné funkcie pre spracovanie obrazu. Pre vytvorenie detekora a deskriptora bodov záujmu používam funkcie:

```
Ptr<FeatuerDetector> detector (new SurfFeatuerDetector());
```

Pomocou vytvoreného detektora je možné extrahovať body záujmu a taktiež vypočítať potrebné deskriptory pomocou funckií:

```
detector.detect(input_img,keypoints);
detector.compute(input_img, keypoints, descriptors);
```

Spracovanie obrazu je výpočtovo náročný proces. Preto som sa rozhodol ukladať vypočítané deskriptory pre obrazy z databázy do súborov typu .yml . Tieto dáta môžu byť neskôr použité pri obrazovej verifikácii.

Kapitola 5

Bag of Words

V oblasti počítačovej grafiky je možné narábať s bodami záujmu (image features) ako so slovami. Táto myšlienka vychádza s techniky vyhľadávania textových dokumetov, kde každý dokument je popísaný pomocou výskytu špecifikých slov z určitého slovníka v ňom. V mojom prípade nenarábam s textom, ale s obrazom, a preto namiesto slov pracujem so špecifickými bodmi obrazu. Na to aby som bol schopný vytvoriť slovník musím vypočítať deskriptory z extrahovaných bodov záujmu. Kedže vychádzam z predpokladu, že tieto body sú špecifické pre každý jeden obraz, na základe nich je možné vytvoriť vizuálny slovnik, pomocou ktorého bude možné vytvárať histogramy pre jednotlivé obrazy. Vytvorené histrogramy popisujú, ako sa jednitlivé slová (v mojom prípade deskriptory) vyskytovali v danom obraze. Proces vytvárania vizuálneho slovníka pozostáva z troch častí:

- Detekcia bodov záujmu
- Vypočítanie deskriptorov
- Vytvorenie vizuálneho slovníka

Detekcia bodov zuáujmu a následne vypočítanie deskriptorov je detailnejšie popísané v predošlej kapitole. V tejto kapitole viac sa zameriam na vytvorenie vizuálneho slovníka a následne na vypočítanie bag of words.

5.1 Vizuálny Slovník

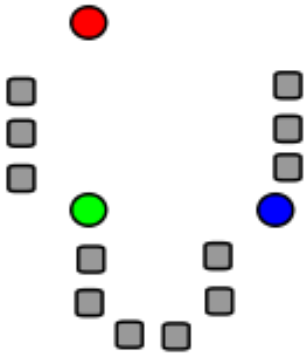
Vizualny slovník je možné definovať ako dátovu štruktúru, ktorá obsahuje špecifické slová, pomocou ktorých je možné popisovať jednotlivé obrazky. V mojom prípade pracujem s deskriptormi, ktoré boli extrahované z každého obrazu z databázy. Pročet deskriptorov pre jendotlivé obrazy je rôzny, a preto je nutné tieto vektory zhlukovať do určitého počtu skupín (clusterov). Pre túto úpravu deskriptorov sa najčastejšie používa algoritmus K-Means, ktorý je veľmi jednoduchý, ale napriek tomu pomerne účinný. Výsledný slovník je repezentovaný ako matica, ktorá obsahuje všetky výpočítané stredy jednotlivých klastrov. Pre klasterizáciu deskriptorov je možné použiť niekoľko metód ako napríklad :

- Conectivity-based clustering
- Centroid-based clustering (K-Means)
- Distribution-based clustering

Väčšina existujúcich algoritmov používa pre klasterizáciu práve metódu K-Means, preto som sa rozhodol použiť práve túto metódu.

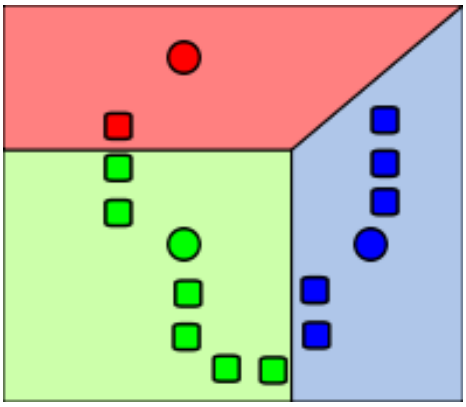
K-Means

K-Means je metóda, ktorá sa bežne používa pre automatické rozdelenie dát do `k` skupín. Je postavená na selekcii `k` počiatočných stredov skupín a iteraktívnom prepočítavaní daných stredov skupín [3].



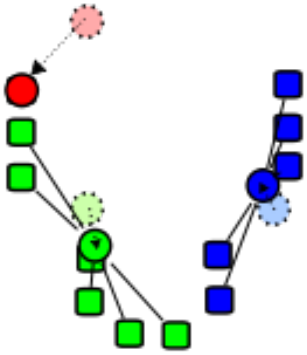
Obr. 5.1: Počiatočný stav. Prevzaté z [7]

Ako je možné vidieť na obrázku 5.1, tak pred začiatkom samotnej klasterizácie došlo k voľbe určitého počtu skupín do ktorých budú dané body rozdelené.



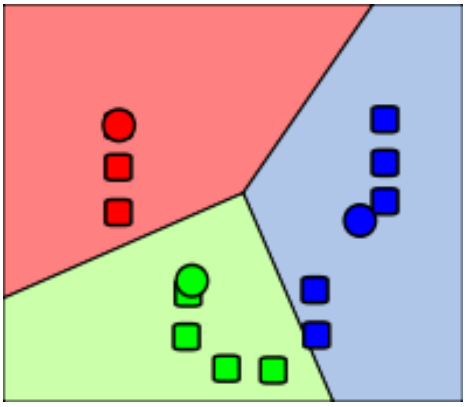
Obr. 5.2: Prvotné priradenie do klustrov. Prevzaté z [7]

Na obrázku 5.2 je zobrazené prvotné roztriedenie bodov do klasterov. Stredy, ktoré boli určené na začiatku celého výpočtu, nereprezentujú stredy daných klusterov, preto musí dôjsť k ich posunu.



Obr. 5.3: Prepočítavanie stredov. Prevzaté z [7]

Ďalším krokom algoritmu je prepočítanie stredov pre každý jeden klaster. Tieto stredy sú prepočítané na základe pozície priradených bodov do daného klastra. Táto časť algoritmu je graficky zobrazená na obrázku 5.3 , kde môžeme vidieť ako došlo k posunu stredov klasterov.



Obr. 5.4: Zobrazenie výsledku po posunutí stredov zhlukov na základe pozície bodov, ktoré patria do konkrétneho zhluku.Prevzaté z [7]

K-Means klasterizácia je iteratívna metóda, ktorá sa opakuje pokiaľ nedosiahne určitej presnosti. V praxi je možné sa stretnúť s použitím tohto algoritmu, kedy počet opakovaní bol presne stanovený ale najlepší spôsob určenia konca výpočtov je podmienka, ktorá znie, že ak neexistuje žiadna daľšia zmena v prideľovaní bodov do klastrov, tak sme dosiahli konečný výsledok.

5.2 Bag Of Words

V počítačovej grafike pojem "bag of words"predstavuje vektor pomocou, ktorého je možné veľmi jednoducho popisovať obsah obrazu. Táto dátová štruktúra je vytvárana pomocou vizuálneho slovníka. Jedná sa o najdôležitejšiu súčasť dnešných vyhľadávacích algoritmov, ktoré pracujú s digitálnymi obrazmi. Táto myšlienka bola prevzatá z algoritmov ktoré pracujú s textovými dokumentami, kedy tieto algoritmy vytvárajú histogram výskytu špecifických slov v danom dokumente. Túto myšienku sa rozhodli aplikovať na digitálne obrazy, kedy sa vytvára histogram výskytu špecifických deskriptorov v danom obraze. Výsledný

histogram dokonale popisuje obraz a vďaka nemu je možné vykonávať veľmi rýchlu klasifikáciu digitálnych obrazov [12]. Tento systém je vhodný pre vyhľadávacie algoritmy, ktoré pracujú nad veľkou databázou obrazov a je potrebné ich filtrovať pred tým, než dôjde k samotnému porovnávaniu deskriptorov.

5.3 Metódy pre porovnávanie

Bag of words predstavujú reťazce, pomocou ktorých je možné popisovať obrazy. Vyhľadávacie algoritmy potrebujú tieto reťazce porovnávať aby na základe tohto porovnania dokázali efektívne vyhľadávať vo veľkych dátových sadách. Na najnižšej úrvoni pracujú existujúce algoritmy s bag of words ako s vektormi. Pre porovnávanie vektorov alebo inak povedané matíc existuje veľa metód, pomocou ktorých je možné vypočítať podobnosť medzi dvoma vektormi. Medzi tieto metódy sa radia napiklad:

- L1 distance $D(h_1, h_2) = \sum_{i=1}^{N} \mid h_1(i) - h_2(i) \mid$
- χ^2 distance $D(h_1, h_2) = \sum_{i=1}^{N} \frac{(h_1(i)-h_2(i))^2}{h_1(i)+h_2(i)}$
- Quadratic distance (cross-bin) $D(h_1, h_2) = \sum_{i,j} A_{ij}(h_1(i) - h_2(j))^2$
- Cosine Distance

Pre svoju jednoduchú implementáciu a dostatočnú presnosť som sa rozhodol vo výslednom frameworku použiť pre porovnávanie bag of words cosinovú vzdialenosť.

Cosinová vzdialenosť

Pojem cosinová vzdialenosť sa spája s počítaním podobnosti medzi dvoma nenulovými vektormi. Podobnosť medzi týmito vektormi je určená pomocou cosinusového uhlu. Ako môžeme vidieť v rovnici 5.1, algoritmus má na vstupe dve matice A a B. Uhol podobnosti je počítaný pomocou všetkých prvkov vektorov. Pokial sa výsledok porovnávania rovná 1, dané dva vektory sú totožné. Čím je výsledok porovnánia menší, tým je menšia podobnosť. Vo výsledku to znamená, že algoritmus môže nadobúdať hodnoty od 1 po (-1), pričom (-1) predstavuje naprostú nezhodu.

$$\cos(\theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.1}$$

Vo výsledom algoritme je táto metóda implementovaná v jednej funkcii s názvom `CosineDist()`, ktorá má za úlohu porovnávať bag of words. Na základe tohto porovnania sú bag of words zoradené do príslušného poradia od najvačšej zhody po najmenšiu zhodu. Toto porovnávanie nám nezaručuje, že hľadaný obraz bude mať najlepší výsledok, ale mal by sa nachádzať medzi najlepšími. Pri tvorbe vyhľadávacieho frameworku som bol nútený určiť, na akej najhoršej pozícii sa môže hľadaný obraz nachádzať.

5.4 Vytváranie Bag of Wrods

Ak vychádzam z predpokladu, že výsledný framework bude pracovať nad veľkou databázou obrázkov, je potrebné tieto obrázky určitým spôsobom filtrovať. Pre filtrovanie som sa rozhodol použiť metódu vytvárania bag of words pre každý jeden obraz z databázy ako aj pre vyhľadávaný obraz.

Na to, aby som bol schopný vytvoriť bag of words, je potrebné zostaviť vizuálny slovník. Tento slonvík sa vytvára z extrahovaných deskriptorov, ktoré je potrebné klasterizovať. Pre kluaterizáciu deskriptorov som sa rozhodol použiť metódu K-Means, pretože sa vykytuje vo väčšine existujúcich riešení, ktoré pracujú s vizuálnym slovníkom. Pri vytváraní slovníka dochádza k spracovaniu celej databázy obrazov, a preto som sa rozhodol priebežne ukladať dáta, ktoré sú ďalej potrebné pre porovnávanie. To znamená, že pri spracovaní databázy dochádza k ukladaniu:

- Bodov Záujmu
- Deskriptorov
- Bag of Words
- Vizuálneho slovníka

K tomuto ukladaniu dochádza z dôvodu zrýchlenia výpočtu samotného frameworku, kedy nedochádza k opätovnému spracovaniu obrazu z databázy pri obrazovej verifikácii. Keby sa tieto potrebné data neukladali, tak pri porovnávani príznakov hľadaného obrazu a obrazu z databázy, by bolo potrebné tento obraz z databázy znova spracovať. Vypočítané dáta sa ukladajú do súboru .yml . Tento typ súboru je vhodný pre ukladanie vektorov alebo matíc. Vytváranie slovníka je v samotnom frameworku oddelené v samostatnej funkcii s názvom

```
void create_dictionary();
```

Táto funkcia si načíta potrebné dáta z databázy, spracuje ich a na základe spracovaných dát vytvorí požadovaný vizuálny slovník. Pre vytváranie vizuálneho slovníka som znova použil jednu z funkcii implementovaých v knižnici OpenCV. Táto knižnica disponuje funkciou:

```
BOWKMeansTrainer bowTrainer(dictionarySize, termcriteria, reties, flags);
```

ktorá vytvorí vizuálny slovník. Tento slovník je potrebné vytrénovať pomocou vypočítaných deskriptorov z obrazov z databázy. To je možné docieliť pomocou funckie:

```
bowTrainer.cluster(featureUnclustered);
```

Počítanie bag of words pre jednotlivé obrazy z databázy je oddelené v samostanej funkcii s názvom

```
void BoW();
```

Samotné vytváranie bag of words je realizované funkciami:

```
BOWImgDescriptorExtractor bowDE(extractor, matcher);
bowDE.setVocabulary(dictionary);
bowDe.compute(img, keypoints, bowDescriptors);
```

Kapitola 6

Verifikácia obrazu

6.1 Porovnávanie Príznakov

Porovnávanie príznakov alebo inak povedané porovnávanie obrazu je súčasťou mnohých aplikáci, ktoré sú zamerané na oblasťi počítačovej grafiky. Medzi takéto aplikácie patria napríklad aplikácie pre kalibráciu kamery, registráciu obrazu a deketcia objektov alebo scén v obraze. Bežný postup porovnávania obrazu pozostáva z detekovania záujmových bodov, pričom každý takýto bod je spojený s deskriptorom popisujúcim určitú oblasť v obraze. Akonáhle sú extrahované body záujmu z obrazu a vypočítané potrebné deskriptory, ďalší krok pri porovnávaní obrazov je stanovenie predbežných zhôd medzi deskriptormi.

Provnávanie dvoch obrazov je možné matematicky definovať ako : nech p je bod detekovaný detektorom v obraze, tak je to možné vyjadriť ako

$$\phi(p) \quad = \quad \{\phi_k(P) \mid k = 1, 2, ..., K\} \tag{6.1}$$

kde pre každé k existuje feature vektor ktorý je možné definovať ako

$$\phi_k(p) \quad = \quad (f^k_{1p}, f^k_{2p}, ..., f^k_{n_k p}) \tag{6.2}$$

Cieľom porovnávania je nájdenie najlepšieho odpovedajúceho deskriptora q v druhom obraze zo sady N záujmových bodov $Q \quad = \quad \{q_1, q_2, ..., q_n\}$ pomocou porovnávania deskriptorov $\phi_k(p)$ a $\phi_k(q)$, ktorá môže byť definovaná ako

$$d_k(p, q) \quad = \quad |d_k(p) - d_k(q)| \tag{6.3}$$

Na základe vzdialenosti d_k sú body zo sady Q sortované do určitého poradia nezávislého pre každý bod p

$$\psi(p, Q) \quad = \quad \{\psi_k(p, Q) | k = 1, 2, ..., k\} \tag{6.4}$$

takže výsledná definícia ma podobu

$$\psi_k(p, Q) \quad = \quad \{(\psi_k^1, \psi_k^2, ..., \psi_k^N) \in |d_k(p, \psi_k^i) \leq d_k(p, \psi_k^j), \forall i > j\} \tag{6.5}$$

Výsledok porovnávania párov bodov záujmu (p, q) je akceptovaný vtedy ak predstavuje najlepšiu zhodu p pre q zo všetkých bodov z prvého obrazu a taktiež ak q predstavuje najlešiu zhodu pre p zo všetkých bodov z druhého obrazu.

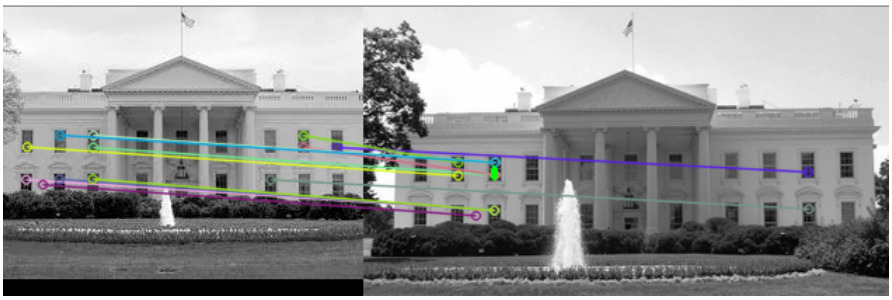
Existujúce algoritmy

Počet algoritmov, ktoré sa zaoberajú porovnávaním obrazov, existuje v dnešnej dobe niekoľko. Medzi takéto algoritmy sa radia napríklad :

- K-Nearest Neighbour
- Brute-Fore Mathcer
- Flann Based MAtcher
- Radius Matcher

Pre vysoké počty deskriptorov je najvhodnejší algoritmus Flann Based Matcher, ktorý funguje na princípe nájdenia najbližšieho suseda. Samotný algoritmus pozostáva z niekoľkých optimalizovaných algoritmov špecializovaných pre rýchle vyhľadávanie. Keďže algoritmy pre porovnávanie fungujú tak, že nachádzajú najbližšiu zhodu, dochádza k nepresnostiam vo výsledku. To znamená, že po provnaní deskirptorov je potrebné filtrovať tieto zhody na spravné zhody (inliers) a na zlé zhody (outliers).

Na to, ako filtrovať tieto zhody existuje mnoho spôsobov a je len na programátorovi, ktorý spôsob si vyberie. Rozdiel medzi filtrovaním a nefiltrovaním zhôd po porovnávaní deskriptorov je možné vidieť na obrázku 6.1 a 6.2 , kde je jednoznačne vidieť, že na obrázku 6.2 sa nachádzajú aj zhody, ktoré na prvý pohlaď nie sú správne.



Obr. 6.1: Ukážka zhody pri porovnávaní image features

Obr. 6.2: Ukážka nezhody pri porovnávaní image features

6.2 Homografia

V oblasti počítačovej grafiky, akékoľvek dva obrazy rovnakého rovinného povrchu v priestore sú prepojené pomocou homografie. Algebraická definícia Homografie znie : mapovanie z $P^2 \Rightarrow P^2$ je projekcia len a len vtedy, ak existuje non-singulárna matica $3x3$ H taká, že akýkoľvek bod s P^2 je reprezentovaný vektorom x, pričom je pravda, že mapovaný bod je zhodný s Hx [8]. Využitie homografie je veľmi všestranné, ako napríklad pri rekonštrukcii obrazu, vypočítanie pohybov kamery alebo rotácie a zvačšenia medzi dvoma obrazmi.

$$s_i \begin{bmatrix} x_i \\ y_i^{'} \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{6.6}$$

Existujúce algoritmy

V dnešnej dobe existuje veľký počet algoritmov, ktoré boli navrhnuté pre počítanie odhadu homografie. Medzi také algoritmy sa radia napríklad :

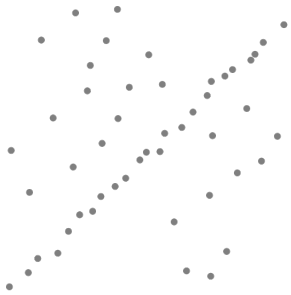
- Basic DLT algorithm
- Different cost functions
- Robust Estimation

Mojím cieľom je vytvorenie frameworku pre vyhľadávanie pomocou obrazu, nebudem detailne rozoberať tieto algoritmy, až na výnimku a tou je RANSAC. Túto metódu detailnejšie popíšem v nasledujúcej podkapitole, pretože sa jedná o metódu, ktorá je nevyhnutnou súčasťou vyhľadácích algoritmov.

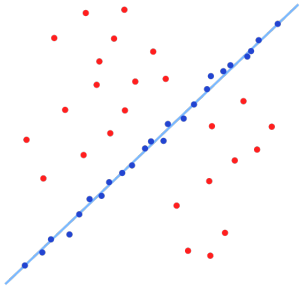
6.3 RANSAC

RANSAC je iteratívna metóda, ktorá sa používa pre odhad parametrov matematického modelu zo sady dát, ktoré obsahjú odchylky. Tento pojem je možné interpretovať ako detekciu

odchyliek. Jedná sa o nedeterministický algoritmus v tom zmysle, že produkuje rozumný výsledok len s určitou pravdepodobnosťou. Táto pravdepodobnosť sa zvyšuje vyšším počtom iterácií. Tento algoritmus bol prvýkrát publikovaný v roku 1981 a jeho autormi sú Fischler a Bolles, ktorý ho použili na vyriešenie LD (Location Determination) problému. Základná myšlienka algoritmu je, že dátová sada pozostáva z hodnôt, ktoré vyhovujú matematickému modelu a z hodnôt ktoré nevyhovujú [9]. V odbornej literatúre je možné sa stretnúť s pojmomami öutliersä ïnliers".



Obr. 6.3: Outliers, z ktorých musí byť detekovaná čiara. Prevzaté z [**?**]



Obr. 6.4: Nájdená čiara pomocou metódy RANSAC. Prevzaté z [1]

.

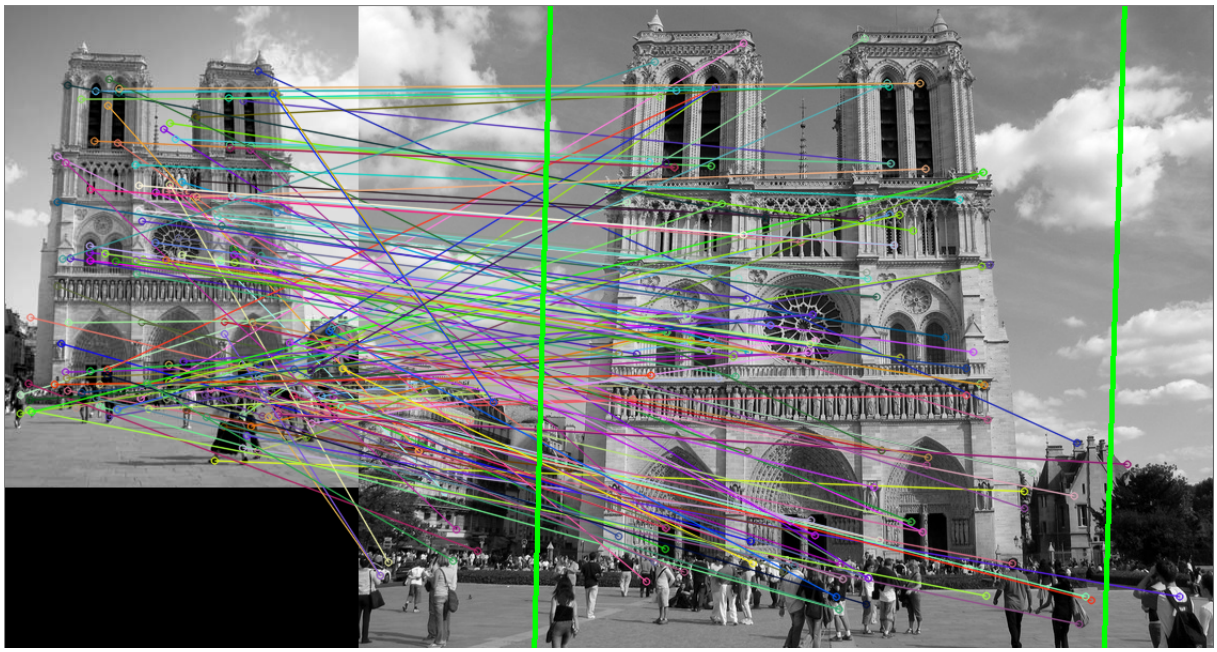
6.4 Porovnávanie a verifikácia nájdených výsledkov

Porovnávanie a verifikácia nájdených výsledkov je veľmi dôležitá časť navrhovaného algoritmu, pretože pokrýva hlavnú funkcionalitu algoritmu. Keďže som sa rozhodol implemetovať tento framework pomocou knižnice OpenCV, bol som obmedzený na metódy, ktoré obsahuje. Táto knižnica ponúka radu metód pre porovnávanie deskriptorov. Medzi tie najčastiejšie používané sa radia Flann Based Matcher a Brute Force Matcher. Ja som sa rozhodol použiť pre porovnávanie práve Flann Based Matcher, pretože je vhodnejší pre veľký počet deskriptorov. Tento typ metódy porovnávania deskriptorov je možné uskutočniť použitím nasledujúcich metód:

```
FlannBasedMatcher matcher;
matcher.match(descriptors_img, descriptors_db_img, matches);
```

Algoritmus urobí obrazovú verifikáciu po porovnaní deskriptorov. Tento proces pozostáva z nájdenia homografie medzi dvoma porovnávanými obrazmi pomocou výsledkov z porovnávania deskriptorov. Úlohou homografie je určiť, či sa medzi porovnávanými obrazmi

nachádza zhoda alebo nie. Výsledkom homografie je tranformačná matica, ktorá reprezentuje zmenu medzi hľadaným obrázkom a obrázkom z databázy. Vďaka tejto matici vie algoritmus presne označiť časť obrazu z databázy, ktorá sa zhodovala s hľadaným obrazom ako je to možné vidieť na obrázku 6.5 .



Obr. 6.5: Nájdený objekt pomocou metódy RANSAC, ktorý je zobrazený pomocou zelenej čiary

Vypočítanie homografie pre dva dané obrazy je možné realizovať pomocou funkcie

```
Mat H = findHomography(imgkeypoints, databaseimgkeypoints, CV_RANSAC, 3,mask);
```

Výsledok tejto funkcie je uložený v matici, pomocou ktorej je možné určiť, ktoré časti dvoch porovnávaných obrazov sa zhodujú.

Kapitola 7

Demo Aplikácia

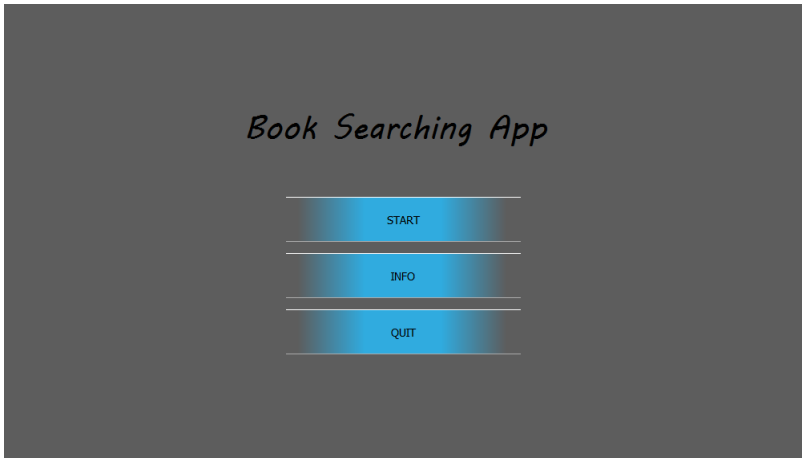
Po vytvorení výsledného frameworku som sa rozhodol ho následne použiť v nejakej aplikácii, ktorá bude daný framework využívať pre vyhľadávanie objektu alebo scény v obrazovej sade. Preto som sa rozhodl vytvoriť aplikáciu s názvom BookSearch, ktorá bude vyhľadávať knihu a základné informácie o knihe len pomocou fotografie predneho obalu knihy.

7.1 Návrh Aplikácie

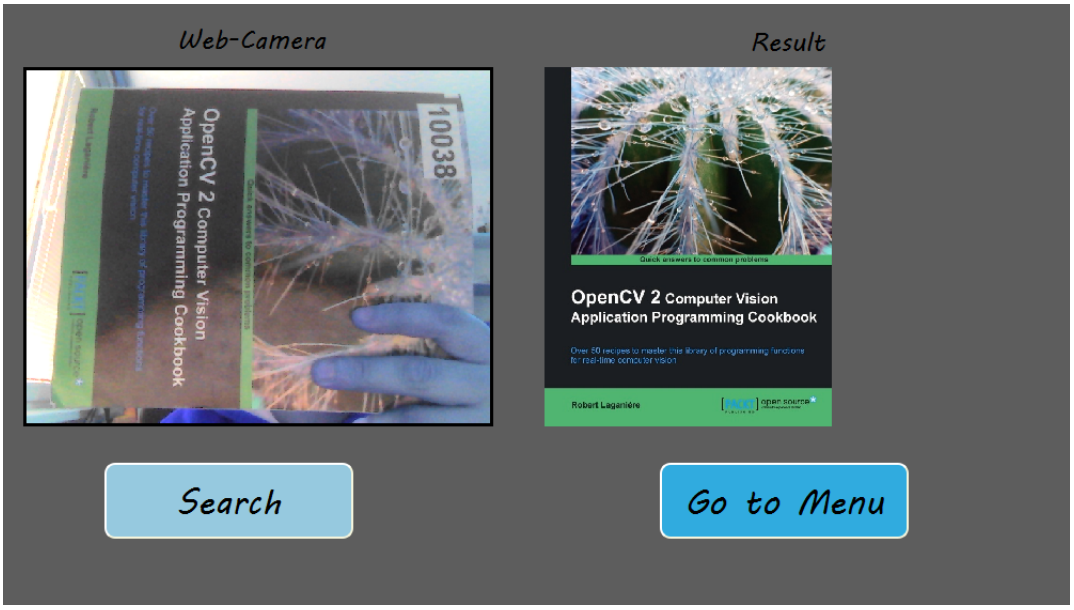
Aplikácia bola navrhnutá, tak aby jej grafické užívateľské rozhranie nebolo príliš zložité a zbytočne nezaťažovalo užívateľa pri jeho používaní. Na to, aby bola aplikácia schopná vyhľadávať pomocou obrazu, bolo potrebné ako prvé vytvoriť dátovú sadu obrazov predných strán kníh. Keďže sa jedná o desktopovú demo aplikáciu, tak ukladanie obrázkov do databázy je možné považovať ako zbytočnú komplikáciu návrhu aplikácie. Rozhodol ukladať databázu obrázkov do súboru k zdrojovým súborom aplikácie. Aplikácia je navrhnutá tak, že vstupom je obrázok knihy zachytený pomocou web kamery. Pre toto riešenie zachytenia obrazu som sa rozhodol z toho dôvodu, pretože v dnešnej dobe je web kamera súčasťou každého notebooku a aj väčšiny stolových počítačov. Samotná aplikácia je vytvorená pomocou frameworku Qt Creator. Na vývoj aplikácie bol použitý programovací jazyk c++ .

7.2 Vyhľadanie obrazu

Po spustení aplikácie má užívateľ možnosť prejsť stlačením tlačítka Start do časti aplikácie pre vyhľadávanie. V tejto časti aplikácie je spustená web-kamera, ktorej úlohou je zachytenie hľadanej knihy. Samotné vyhľadávanie nezačne pokiaľ užívateľ nestlačí tlačíko "search". Po stlačení daného tlačítka dôjde k zachyteniu fotografie a následne je spustené samotné vyhľadávanie. Aby aplikácia dokázala reagovať na užívatelove príkazy aj počas vyhľadávania, je vyhľadávanie spustené v novom vlákne. Po určitej dobe, keď skončí vyhľadávanie, je užívateľovi zobrazený výsledok vyhladávania ako je možné vidieť na obrázku 7.2 .



Obr. 7.1: Úvodne okno aplikácie BookSearch



Obr. 7.2: Nájdená kniha pomocou aplikácie BookSearch

Na obrázku 7.2 je možné vidieť úspešné najdenie knihy. Výsledok vyhľadávania je zobrazený vpravo od okna, ktoré zobrazuje vstup z web-kamery. Po ukončení vyhľadávania ma užívateľ znova možnosť spustiť ďalšie vyhľadávanie stlačením tlačítka "Search".

Obr. 7.3: Ukážka najdenia nesprávnej knihy

Kapitola 8

Dosiahnuté výsledky

Cieľom tejto práce bolo vytvorenie frameworku, ktorý bude vyhľadávať v dátovej sade obrazov. Výsledný program by mal dokázať vyhľadať obraz alebo len časť obrazu v databáze obrazov. Na to aby som bol schopný zistiť do akej miery je moje riešenie presné je potrebné otestovať jeho funkcinalitu. Rozhodol som sa vytvoriť sadu testov, ktoré budú skúmať do akej miery je vyhľadávanie konkrétneho obrazu úspešné.

8.1 Testovanie frameworku

Pre testovanie vytvoreného frameworku som sa rozhodol vytvoriť sadu testov v ktorých budem sledovať

- pozíciu hľadaného obrazu po porovnávaní bag of words
- priemernú euklidovskú vzdialenosť správnych výsledkov
- percentil úspešnosti nájdenia obrazu s rôznymi kvalitami obrazu.

Pozícia hľadaného obrazu po Bag of Words porovnaní

Vytváranie Bag fo words má za úlohu prefiltrovanie dát z databázy, aby do výslednej časti porovnávania nevstupoval veľký počet dát, pretože táto časť frameworku je veľmi výpočtovo náročná. Z toho dôbodu dochádza k porovnávaniu bag of words. Výsledky sú následne zoradené od najlepšieho po prvok s najhorším výsledkom. Týmto testom sa budem snažiť zistiť, na akej priemernej pozícii sa nachádza hladaný obraz po porovnáni bag of words. Vďaka týmto testom budem schopný povedať, koľko prvkov by malo vstupovať do fázy porovnávania features, tak aby to bolo čo najefektívnešie.

Priemerná euklidovská vzdialenosť

Pri porovnávaní deskriptorov dochádza k počítaniu euklidovskej vzdialenosti.Táto vzdialenosť reprezentuje, na koľko sú si dva body podobné. Všetky nástroje pre porovnávanie features pracujú na princípe nájdenia najbližšieho suseda. To znamená, že výsledky nie vždy reprezentujú zhodu, ale môžu obsahovať aj chybné dvojice bodov. Spôsob, ako je možné

tieto body filtrovať na dobré zhody a zlé zhody, je možné uskutočniť pomocou počítania priemer veľkosti euklidovskej vzdialenosti. Pomocou tohto výpočtu som schopný zistiť, či došlo k porovnaniu dvoch úplne odlišných obrazov alebo tieto dva obrazy môžu zobrazovať podobnú scénu a je potrebné zistiť, ktoré časti obrazov sa zhodujú. Rozhodol vytvoriť testy pre vypočítanie priemernej euklidovskej vzdialenosti správnych výsledkov. Vďaka týmto testom budem schopný nastaviť výsledný algoritmus tak, aby dokázal efektívne selektovať správne zhody od nesprávnych.

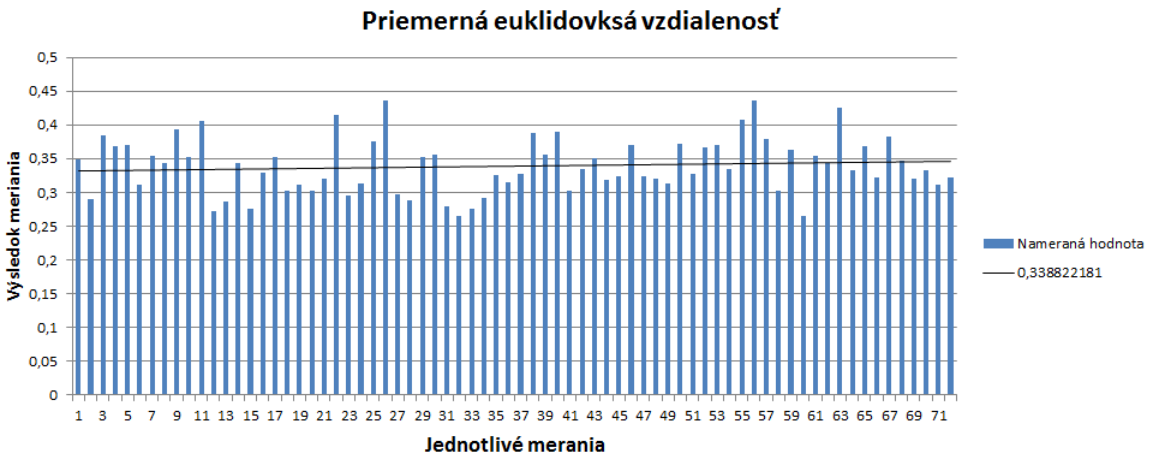
Percentil úspešnosti nájdenia obrazu

Vytvorený framework bol navrhnutý pre vyhľadávanie scén alebo objektov z dátovej sady obrazov. Je dôležité zistiť na koľko je navrhované riešenie úspešné pri vyhľadávaní obrazu. Rozhodol som sa testovať výsledný framework sadou testov, kde som skúmal, koľko percent hľadaní končí úspešne.

8.2 Analýza testov

Výsledky testovania priemernej euklidovskej vzdialenosti

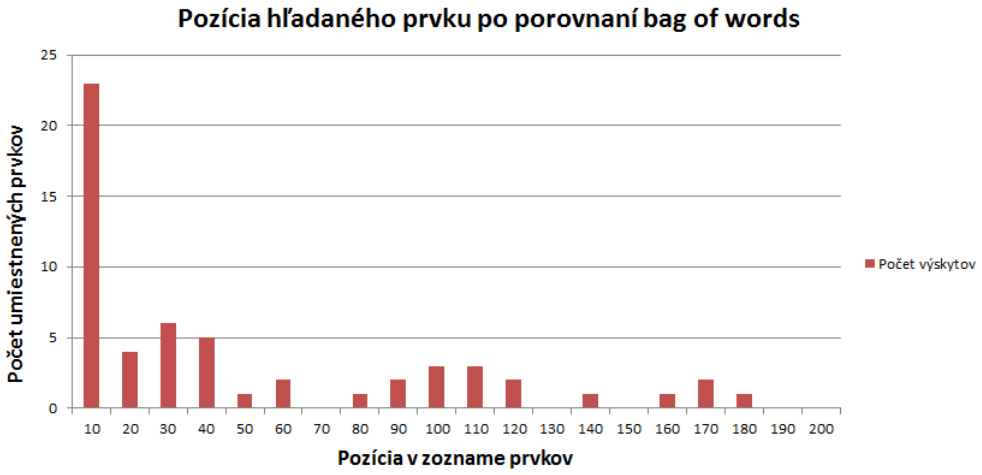
Z výsledkov testovania, ktoré sú zobrazené v tabuľke 8.1 je možné vidieť, že priemerná veľkosť vypočítanej euklidovej vzdialenosti medzi bodmi dvoch obrazov zobrazujúcich rovnakú scénu sa pohybuje na hranici 0.3108 . Vďaka tomuto zisteniu som schopný určiť počas vyhľadávania, kedy sa jedná o nezhodu a zhodu porovnávania dvoch obrazov. Napriek tomu stanoveniu tejto hranice euklidovej vzdialenosti nie je stopercentná filtrácia nezhôd a zhôd, pretože môže nastať prípad, kedy dva obrazy, ktoré nezobrazujú rovnakú scnénu môžu dosiahnuť menšiu priemernu euklidovskú vzdialenosť ako je testami určená hranica.



Obr. 8.1: Graf popisujúci priemernú hodnotu euklidovsej vzdialenosti pri porovnávaní dvoch obrazov s rovnakou scénou alebo objektom.

Výsledky testovania pozície hľadaného prvku v zozname obrazov z DB

Výsledky testovania pozície hľadaného obrazu z databázy v zozname obrazov zoradených podľa najväčšej podobnosti bag of words, som zistil, že väčšina hľadaných obrazov sa vyskytovala medzi pozíciami 1-20. Tieto výsledy je možné vidieť aj na grafe 7.2 , kde je možné vidieť ako s narastajúcou pozíciou klesá počet výskytov. Vďaka tomuto zisteniu je algoritmus schopný sa zamerať na určitý počet obrazov po porovnávaní bag of words. Keďže porovnávanie bag of words obrazov nie je 100% metóda a hľadaný obraz z databázy sa nie vždy vyskytoval s najlepším výsledkom porovnania, bolo potrebné zisiť, s akým najčastejším najhorším výsledkom je potrebné pracovať po porovnávaní bag of words obrazov z databázy s hľadaným obrazom.



Obr. 8.2: Graf popisujúci umiestnenie hľadaného obrazu v zozname obrazov z databázy po porovnaní bag of words.

Výsledky testu úspešnosti nájdenia obrazu

Vytvorený framework som testoval s databázou obrázov o veľkosti 1000 obrazov. Do tejto databázy bolo cielene umiestnených pár obrazov, ktoré som sa neskôr pokúsil vyhľadať. Pri vyhľadávaní som vždy použil obraz, ktorý zobrazoval rovnáky objekt alebo scénu, ale bol odfotený z inej vdzialenosti alebo uhla. Z 20 vyhľadávaní bolo úspešných 13. To znamená, že na základe týchto výsledkov má mnou vytvorený framework 65% úspešnosť.

Kapitola 9

Záver

Cieľom mojej praće bolo navrhnutie a vytvorenie frameworku, ktorý bude vyhľadávať obraz na základe jeho obsahu v dátovej sade obrazov. Po preštudovaní potrebnej literatúry som bol schopný vytvoriť prvotný návrh celého frameworku, ktorý som následne aj implementoval. Pri riešení tejto úlohy som sa snažil čerpať čo najviac poznatkov z existujúcich riešení, ktoré používajú podobnú alebo rovnakú technológiu pre vyhĺadávanie obrazu.

Podarilo sa mi vytvoriť výsledny framework, ktorý dokáže vyhľadať obraz na základe jeho obsahu v referenčnej dátovej sade obrazov. Vytvorený vyhľadávací nástroj používa pre samotné vyhľadávanie metódu bag of wrods, vďaka ktorej filtruje dáta z databázy. Framework je navrhnutý tak, aby dokázal pracovať nad akoukoľvek veľkou databázou obrazov. Referenčná databáza je spracovaná iba vtedy, pokiaľ dôjde k zmene jej obsahu, inak pracuje algoritmus s vypočítanými hodnotami, ktoré sú uložené v súborovej štruktúre vytvoreného frameworku.

Po vytvorení vyhľadávacieho frameworku som sa rozhodol použiť tento nástroj v praktickej aplikácii, ktorá ho bude využívať pre vyhľadávanie obrazu. Z toho dôvodu som vytvoril aplikáciu, ktorá vyhľadáva knihu, na základe fotky jej prednej strany. Preto som sa rozhodol nazvať túto aplikáciu BookSearch. Jedná sa o desktopovú aplikáciu, ktorá využíva web kameru pre vytváranie obrazov, ktoré sú následne vyhľadávané. Vytvorená aplikácia disponuje vlastnou dátovou sadou obrazov predných strán kníh, v ktorej vyhľadáva. Jedná sa o jednoduchú aplikáciu, ale v budúcnosti by som chcel dorobiť niektoré zaujímavé funkcie ako je editovanie databázy obrazov užívateľom alebo zobrazovanie viac informácii o knihe, ktorá bola nádjená.

Samotný vyhľadávací framework bol testovaný na úspešnosť nájdenia obrazu, priemernú euklidovskej vzdialenosti pri porovnávaní obrazov so zhodným obsahom a pozíciu hĺadaného obrazu v zozname obrazov, po porovnávaní bag of words. Výsledky dvoch spomenutých testov boli aplikované v samotnom frameworku a na ich základe došlo k zlepšeniu samotného vyhľadávania obrazu.

V budúcnosti by som chcel zlepšiť celkovú rýchlosť vyhľadávania, ktorú by som chcel docieliť paralérnym vyhľadávaním alebo paralérnym spracovaním obrazov. Ďalej by som chcel zlepšiť presnosť vyhľadávania tak, aby vytvorený framework dokázal nájsť hľadaňy obraz alebo scénu aj pri menej kvalitnejších obrazoch alebo obrazoch kedy hľadaný objekt zaberá len malú časť celkového obrazu.

Literatúra

[1] [Online; navštíveno 17.04.2017].
URL https://en.wikipedia.org/wiki/Random_sample_consensus

[2] *Bag-of-Features Descriptor on SIFT Features with OpenCV (BoF-SIFT)* . [Online; navštíveno 18.04.2017].
URL https://www.codeproject.com/Articles/619039/Bag-of-Features-Descriptor-on-SIFT-Features-with-O

[3] *Constrained K-means Clustering with Background Knowledge*. [Online; navštíveno 15.04.2017].
URL https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf

[4] *Introduction to SIFT (Scale-Invariant Feature Transform)*. [Online; navštíveno 16.04.2017].
URL http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html

[5] *Introduction to SURF (Speeded-Up Robust Features)*. [Online; navštíveno 15.04.2017].
URL http://docs.opencv.org/trunk/df/dd2/tutorial_py_surf_intro.html

[6] *Introduction to SURF (Speeded-Up Robust Features)* . [Online; navštíveno 17.04.2017].
URL http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html

[7] *k-means clustering* . [Online; navštíveno 10.04.2017].
URL https://en.wikipedia.org/wiki/K-means_clustering

[8] *Random sample consensus* . [Online; navštíveno 11.04.2017].
URL https://en.wikipedia.org/wiki/Random_sample_consensus

[9] *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography* . [Online; navštíveno 11.04.2017].
URL http://www.dtic.mil/dtic/tr/fulltext/u2/a460585.pdf

[10] *SIFT vs SURF* . [Online; navštíveno 10.04.2017].
URL http://zhvillues.tumblr.com/post/119561114181/sift-vs-surf

[11] *SURF: Speeded Up Robust Features*. [Online; navštíveno 20.04.2017].
URL http://www.vision.ee.ethz.ch/~surf/eccv06.pdf

[12] *Visual Categorization with Bags of Keypoints* . [Online; navštíveno 10.04.2017].
URL https://people.eecs.berkeley.edu/~efros/courses/AP06/Papers/csurka-eccv-04.pdf

[13] *Visual descriptor*. [Online; navštíveno 15.04.2017].
URL https://en.wikipedia.org/wiki/Visual_descriptor

[14] Laganiére, R.: *OpenCV 2 Computer Vision Application Programming CookBook*. Packt Publishing Ltd, 2011, ISBN 978-1-849513-24-1.

[15] Myron Flickner, W. N. J. A. Q. H. B. D. M. G. J. H. D. L. D. P. D. S., Harpreet Sawhney; Yanker, P.: *Query by Image and Video Content: The QBIC System*. [Online; navštíveno 20.03.2017].
URL http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=410146

[16] TinEye: *Frequently asked questions*. 2010, [Online; navštíveno 18.03.2017].
URL https://www.tineye.com/faq#how

[17] Tsai, C.-F.: *Bag-of-Words Representation in Image Annotation: A Review*. Aug 2012, [Online; navštíveno 01.05.2017].
URL https://www.hindawi.com/journals/isrn/2012/376804/

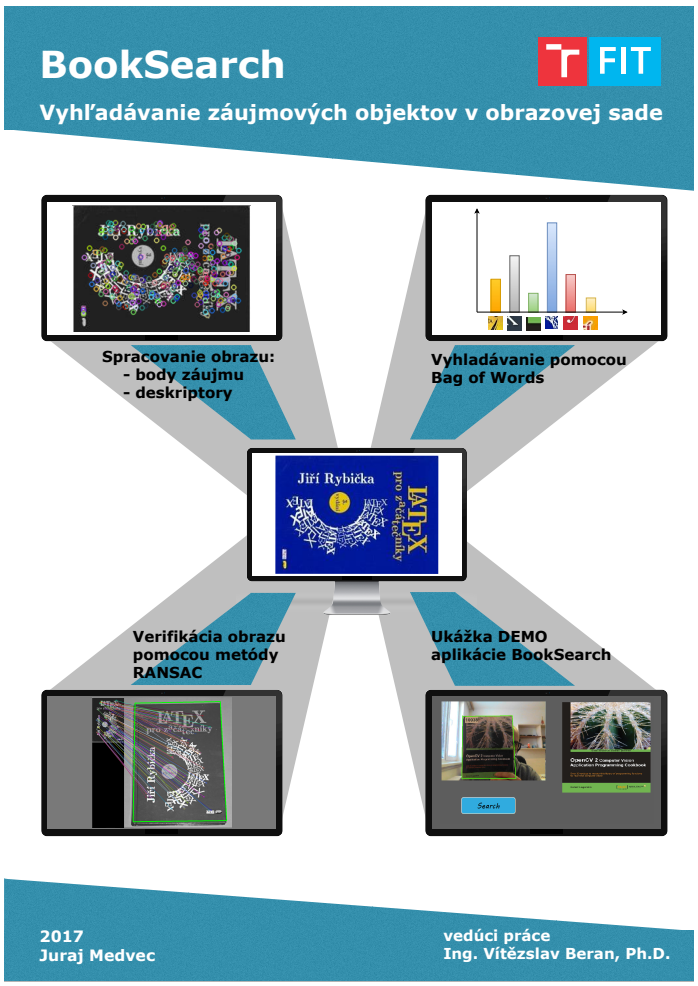
[18] Woodruff, C.: *Feature Detection and Matching* . [Online; navštíveno 10.05.2017].
URL https://courses.cs.washington.edu/courses/cse576/13sp/projects/project1/artifacts/woodrc/index.htm

[19] Yong Rui, T. S. H.: *Image Retrieval: Current Techniques, Promising Directions, and Open Issues*. 1998, [Online; navštíveno 20.03.2017].
URL http://ac.els-cdn.com/S1047320399904133/1-s2.0-S1047320399904133-main.pdf?_tid=87acf5fc-3443-11e7-94b7-00000aacb362&acdnat=1494285210_815bc209c7167cc5e687297eeb265467

Prílohy

Príloha A

Plagát



Príloha B

Obsah pameťového média

- Zdrojové súbory výsledného programu
- Plagát
- Prezentačné video