VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

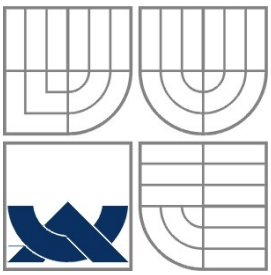
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ FOTOGRAFIÍ PODLE OBSAHU

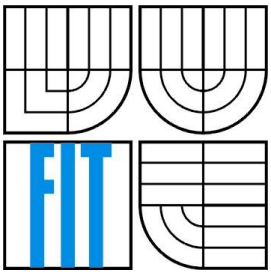
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE RADEK BAŘINKA
AUTHOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPGICS AND MULTIMEDIA

VYHLEDÁVÁNÍ FOTOGRAFIÍ PODLE OBSAHU

CONTENT BASED PHOTO SEARCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADEK BAŘINKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL, PH.D.

BRNO 2013

Abstrakt

Tato práce se zabývá problematikou vyhledávání fotografií podle obsahu a existujících aplikací zabývajících se touto oblastí. Cílem je lokálně pracující aplikace pro vyhledávání fotografií podle obsahu zadaného vzorem. Součástí řešení je jednoduché grafické rozhraní, podpora ukládání a načítání dat z přenosné, lokální databáze. Aplikace vyhledává ve zvolené sadě fotografie, které jsou obsahem podobné zadanému vzoru. Výsledky pak visuálně předloží uživateli. Extrakce příznaků a detekce fotografie podle obsahu je řešena pomocí algoritmu SURF, visuálního slovníku vytvořeného metodou k-means a popisu obsahu fotografií jako bag of words. Dále je vyhledávání fotografií podle kosinové podobnosti vektorů doplněno o samostatný výpočet homografie a selekci hledaných regionů ve vzorové fotografii. V závěru technické zprávy jsou zmíněny výsledky testů.

Abstract

This thesis deals with the problematics of searching of photographs by the content and existing applications dealing with this subject. The aim is the local working application for searching of photographs by the content given by a pattern. The solution consists of the simple graphical interface, the support of saving data and the reading of data from the transferable local database. The application searches the photographs of a given set that are similar to the given pattern. The results are visually depicted to the user. Feature extraction and detection by photo content is solved by means SURF algorithm, visual vocabulary created by method k-means and a description of photography as a bag of words. In addition,the searching of photographs by cosine similarity of vectors enriched with the independent calculation of homography and the selection of regions searched in an example photography. At the end of the technical report the results of testing are presented.

Klíčová slova

Vyhledávání fotografií podle obsahu, deskriptory obrazu, klíčové body obrazu, algoritmus SURF, algoritmus k-means, visuální slovník, bag of words, kd-tree, homografie, kosinová podobnost.

Keywords

Content based image search, image descriptors, image keypoints, SURF algorithm, k-means algorithm, visual vocabulary, bag of words, kd-tree, homography, cosine similarity.

Citace

Radek Bařinka: Vyhledávání fotografií podle obsahu, bakalářská práce, Brno, FIT VUT v Brně, 2013

Vyhledávání fotografií podle obsahu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla, Ph. D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Radek Bařinka
Datum 15.5 2013

Poděkování

Tímto chci poděkovat panu Ing. Michalu Španělovi, Ph.D. za podnětné rady a řádné vedení při tvorbě mé bakalářské práce.

Obsah

1 Úvod

Tato práce se zabývá vývojem aplikačního řešení pro vyhledávání fotografií podle zadaného vzoru. Na na dnešním softwarovém trhu existuje celá řada produktů, které se touto oblastí zabývají. Jmenovitým příkladem je tak například Google Images, který je založen hlavně na vyhledávání napříč internetovou sítí. Jeho používání je poskytováno zdarma, avšak pro vyhledávání je nutné aktivní připojení k internetu. Ze známých komerčně provozovaných aplikací můžu zmínit aplikaci *Adobe* Photoshop Elements 11.

Cílem této práce bylo vytvořit aplikaci pro vyhledávání fotografií podle obsahu zadaného vzoru. Z hlediska vyhledávání se metodologie zakládá na některém ze známých postupů. Aplikace podporuje lokální zpracování a ukládání obrazových dat. Z implementačního hlediska se skládá z volně dostupných knihoven a uživatel této aplikace není komerčně vázán jejím používáním. Její provoz je čistě lokálního charakteru a nepotřebuje internetové připojení. Ovládání realizuje jednoduché GUI se základním nastavením pro tvorbu a mazání zdrojových dat. Konečné výsledky vyhledávání jsou předkládány v přijatelné grafické podobně uživateli.

Následující *kapitola 2* se zaobírá teorií detekce a extrakce obsahu fotografických dat. Popisuje části zvolené nastudované metodiky pro vyhledávání podobných obrázků a s tím související algoritmy. *Kapitola 3* pojednává o existujících aplikacích v oblasti vyhledávání fotografií a obrázků. Koncem této kapitoly nastiňuji některé nejnovější algoritmy, které se při vyhledávání používají. V *kapitole 4* podrobně popisuji návrh mnou vytvořené aplikace. *Kapitola 5* popisuje implementační detaily a realizaci jednotlivých částí aplikace. Jsou zde také zmíněny stěžejní třídy a metody, které jsou v nich využity. V *Kapitole 6* navrhuji testování a předkládám dosažené výsledky zaměřené na vyhledávání a paměťovou náročnost databáze. Dále diskutuji nad dalším možným vývojem projektu. Závěrem práce hodnotím celkovou funkčnost aplikace.

2 Vyhledávání fotografií podle obsahu

Už při visuálním vyhledávání korespondencí mezi fotografiemi se na klade důraz na jejich obsah z hlediska podobnosti. Základním kamenem vyhledávání je tedy detekce obsahu fotografie. Při jeho analýze je potřeba zajistit takový druh informací, které by obsahovaly všechny vzorky prohledávané sady a na jeho základě určit podobnost.

Takovou informaci představují obrazové příznaky, které jsou tvořeny číselnými vektory. Mohou být lokálního nebo globálního charakteru. Tzn. popisují celek nebo část obrazu či fotografie. Za důležitou vlastnost detekovaných příznaků se považuje invariance. Konkrétně myšlena je stálost příznaku vůči rotaci, změna velikosti, zkreslení obrazu či úhlu natočení nebo změna osvětlení atd. [1].

2.1 Detekce klíčových bodů v obraze

S lokálními příznaky koresponduje detekce klíčových (významných) bodů v obraze, viz. obr. 1. Tyto body jsou charakterizovány svým okolím, které tvoří vektory lokálních příznaků. V detekovaném obraze je můžeme najít například v podobě hran, blobů nebo T-Křižovatkek [2].

K nalezení těchto příznaků se používají různé algoritmy (SIFT,SURF,..), které účelně spojují implementaci detektorů klíčových bodů, tak charakteristiku deskriptorů. Detektory v konečné fázi identifikují odpovídající příznakové vektory mezi různými obrázky. Jejich hledání se zakládá na Eukleidovské nebo Mahalanobisově vzdálenosti.

Nejdůležitější vlastností těchto detektorů je opakovatelnost (repeatability). Opakovatelnost znamená, že daný detektor najde stejný počet klíčových bodů po opakované změněně podmínek pohledu. Další důležitou vlastností je rozměr vektorů deskriptorů, který ovlivňuje časovou náročnost výpočtu. Čím vyšší rozměr je, tím se spotřebovává více času, a proto jsou jeho nižší rozměry žádoucí [2].

Obrázek 1: Na fotografii jsou zeleně zvýrazněny detekované klíčové body za použití metody SURF.

2.1.1 SURF

SURF (Speeded-Up Robust Features*)* [2] je algoritmus pro detekci a extrakci lokálních obrazových příznaků. Jeho vzniku předcházela myšlenka pro vývoj výkonného, robustního detektoru a deskriptoru. SURF takové vlastnosti splňuje s ohledem na opakovatelnost, invarianci, rozlišovací způsobilost a hlavně rychlost. Vznikl jako sloučení již několika existujících funkčních metod a vychází ze staršího algoritmu SIFT. SURF je nezávislý na barvách daného obrazu, využívá pouze intenzity bodů v obsahu obrazu ve formě integrálních obrazů. Detekce se zakládá na *determinantu Hessovy matice.* Determinant je použit jak při extrakci tak i detekci příznaků [2] [3] [4].

Integrální obraz je reprezentací vstupního obrazu, jehož hodnoty jasu jednotlivých pixelů se kumulativně sčítají v řádcích a sloupcích. Tohoto konceptu se využívá k redukci výpočetního času detekce klíčových bodů v detektoru SURF. Existuje-li obraz $I_\Sigma(x)$ v bodě x = (*x*, *y*), pak jeho reprezentace vypadá následovně [4].

$$I_\Sigma(x)=\sum_{i=0}^{i\leq x}\sum_{j=0}^{j\leq y} I(i,j) \tag{1}$$

Detekce je založena na *Hessově matici*, protože toto řešení podává dobrý výpočetní čas a výkon. Matice použitá v detektoru vypadá následovně. Je dán bod x = (*x*, *y*) v obrazu I, pak je Hessova matice H(x, σ) v bodě x při měřítku σ definována podle rovnice (2). L_{xx}(x, σ) je konvolucí druhé derivace Gaussovy funkce dle rovnice (3) s obrazem I a podobně pro L_{xy}(x, σ) a L_{yy}(x, σ) [2].

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xx}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{2}$$

$$\frac{\partial^2}{\partial x^2} g(\sigma) \tag{3}$$

2.2 Tvorba a použití visuálního slovníku

Visuální slovníky představují ekvivalenci ke slovníkům lexikografickým, avšak jejich použití spočívá ve vytváření popisů obsahů fotografií. Metodologie je podobná jako u textových dokumentů. V případě zájmu o analýzu textu, je třeba vyhledat existující slova ze slovníku a sledovat např. jejich počet. Visuální slovníky však nejsou naplněny lexikografickými daty, ale deskriptory pocházejícími z detekce a extrakce obrazu. Deskriptory tak přestavují visuální slova. U obsahu fotografie se sleduje jejich výskyt.

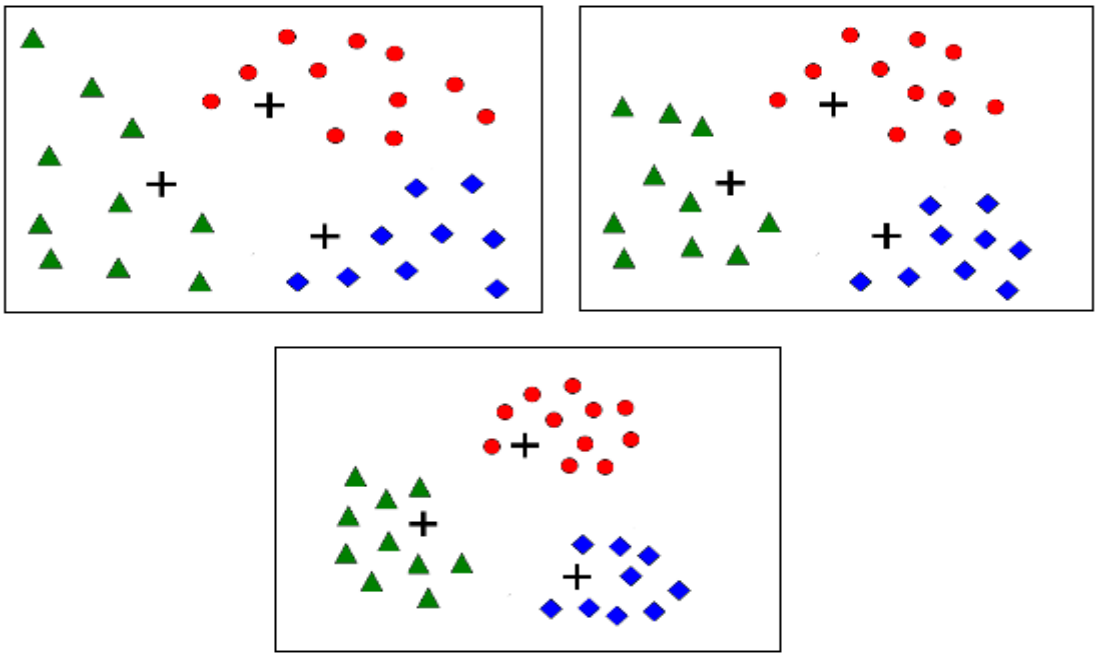
Obsah fotografií se od sebe většinou liší, a proto se liší i deskriptory. Není možné vytvářet slovník všech slov ze všech fotografií. Deskriptory na různých fotkách mohou být podobné. Je tedy vhodné vytvořit takovou reprezentaci slovníku visuálních slov, aby popisem pojala celou množinu obrazových dat a bylo při tom dosaženo dostatečné rozlišitelnosti.

Tvorba visuálních slovníků z deskriptorů se zakládá na algoritmech shlukování. Z extrahovaných deskriptorů jednotlivých fotografií se vytvoří shluky, které realizují slova pro celé množiny. K tomu účelu dobře slouží algoritmus k-means.

2.2.1 K-means

Patří mezi nehierarchické algoritmy pro setřiďování dat do shluků. Na začátku je zadán počet středů, kterými se určí počet shluků. Ten musí být menší, než počet objektů určených ke shlukování. Práce algoritmu spočívá v tom, že každý objekt přiřadí k nejbližšímu středu. Grafické znázornění iterací algoritmu je prezentováno na obr. 2. Středy se také v každé iteraci přepočítají jako aritmetické průměry bodů shluku. Konečnou podmínkou algoritmu je dosáhnou co nejmenších rozdílů uvnitř shluků. Výpočet tohoto minima zachycuje vzorec (4), kde k určuje počet množin bodů S_i, x_i je vektor reprezentující i-tý bod a μ_i je středem shluku [5] [6].

$$V_{min}=\sum_{i=1}^{k}\sum_{x_j\in S_i}\|x_j-\mu_i\|^2 \tag{4}$$



Obrázek 2: Průběh metody k-means.

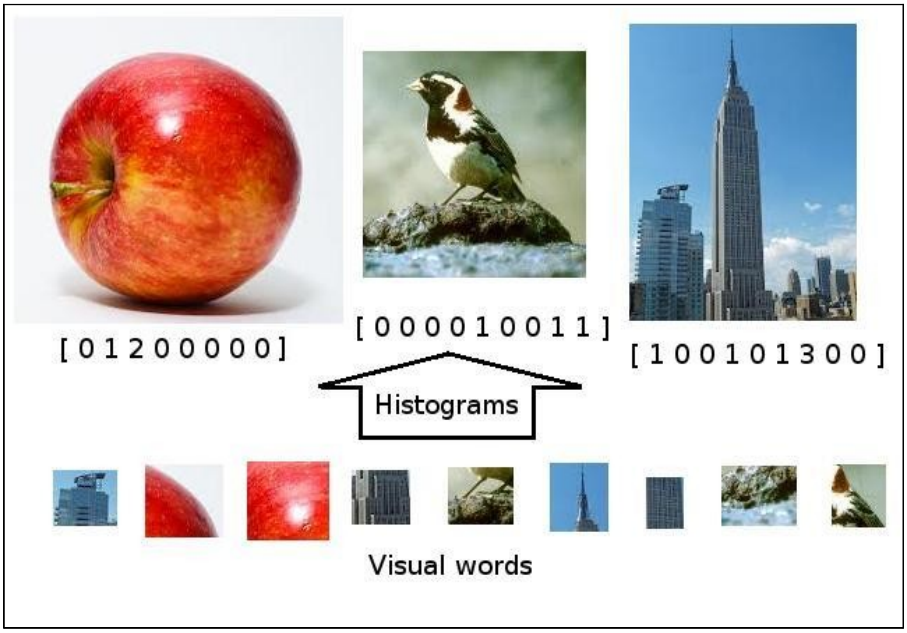
Obecný postup algoritmu k-means:

1. Je vybráno *k* objektů, které jsou považovány za středy shluků (náhodně, či dle metodiky).
2. Každý objekt je přiřazen k nejbližšímu centru shluku. Centra shluků jsou poté přepočítána.
3. Opakování přiřazení objektů.
4. Pokud je splněna podmínka (např. zadaná přesnost vzdálenosti shluků od středu, určený počet iterací apod.) shlukování se ukončí.

2.3 Bag of words

BoW (Bag of Words) je metoda původně určená pro popis obsahu textových dokumentů. Dokumenty se dají analyzovat na základě lexikografických slovníků. Bude-li sledován počet jednotlivých druhů slov ze slovníku vyskytujících se v náhodně vybraném dokumentu, tak výsledkem se stává histogram hodnot takovéto početnosti. Tento histogram je v podstatě reprezentací BoW. Pokud tento postup aplikujeme na celou množinu dokumentů, získáme tím jejich popis, na jehož základu lze sledovat míru podobnosti obsahu mezi jednotlivými dokumenty.

Tato metoda našla využití i při popisu obsahu obrazových dat. Její použití je úzce svázáno s visuálním slovníkem, viz. kapitola 2.2. Vytvořené shluky z visuálního slovníku pak reprezentují hledaná slova. Sledován je jejich počet v obsahu jednotlivých fotografií.



Obrázek 3: Visuální ukázka histogramů bag of words.

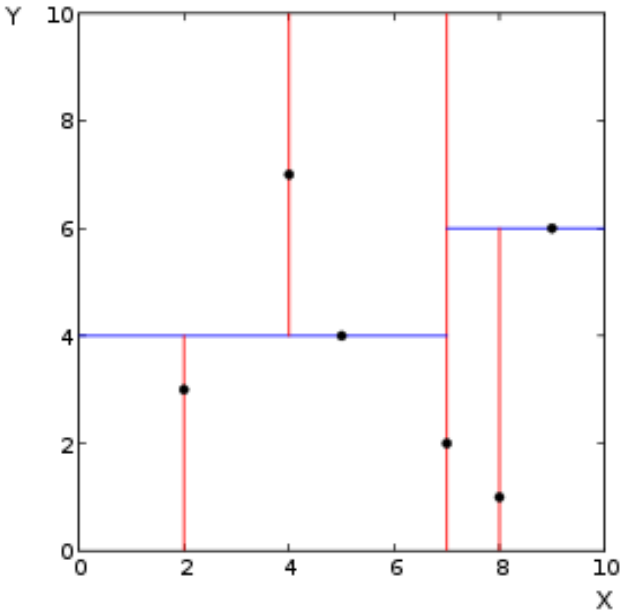
Visuální slova představují reprezentaci pro celou množinu obrazových dat. Je tedy nutné při hledání jejich výskytu přikročit k aproximaci, protože slova na fotografiích se spíše svou číselnou reprezentací blíží k požadovaným hodnotám, než že by přesně odpovídala. Toho je například docíleno vhodným řešením problému hledání nejbližšího objektu *A* v prostoru vzhledem k objektu *B*. V případě BoW jsou za objekty považovány číselné vektory histogramů.

2.3.1 Kd-tree

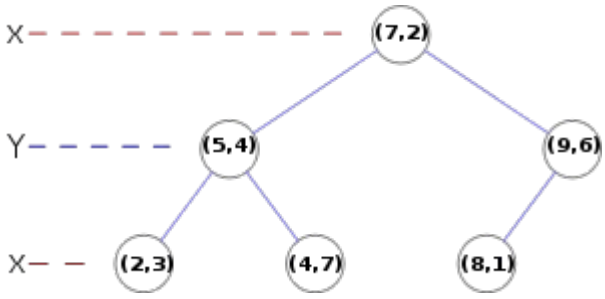
Kd-tree (k-dimensional tree) v překladu k-dimenzionální strom je datová struktura pro uspořádání bodů v k-rozměrném prostoru, viz. obr 4. Uspořádání se podobá binárnímu stromu, viz obr. 5, kde každý uzel je k-dimenzionálním bodem, který dělí rovinu, jíž prochází, na dvě podroviny. Bod reprezentují číselné hodnoty uspořádané k-tice. Tyto k-tice se používají jako vyhledávací klíč při pohybu stromem [7].

Tato datová struktura je tedy vhodná pro ukládání a vyhledávání k-dimenzionálních vektorů napříč prostorem. Toho se používá při tvorbě BoW, kdy je visuální slovník jako množina vektorů reprezentován tímto stromem. Nad ním poté probíhají specifikované dotazy (queries). Jedním

z takových dotazů je i hledání nejbližšího souseda (nearest neighbour search). Tento dotaz nad stromovou reprezentací slovníku vrací výsledek v podobě vyhledání nejbližšího slova (vektoru) k slovu obsaženému v obsahu fotografie a řeší tak nastíněný problém v kapitole 2.3 [7] [8].



Obrázek 4: Příklad kd-tree v prostoru při k = 2 [8].



Obrázek 5: Kd-tree realizován jako strom při k = 2 [8].

2.4 Podobnost fotografie na základě BoW

Dle kapitoly 2.3 jsou BoW reprezentovány histogramy. Histogram se dá vyjádřit jako vektor kladných celých čísel, přičemž jednotlivé hodnoty značí velikosti sloupců. Při hledání podobnosti

mezi BoW se vychází z myšlenky podobnosti těchto vektorů. Jedna z možností se nabízí ve výpočtu eukleidovské vzdálenosti, kdy se v eukleidovském prostoru pro každý bod vektoru *p* spočítá vzdálenost od jeho protějšku *q*. Čím více se vzdálenost mezi vektory snižuje, tím více se podobají viz rovnice (5).

$$distance(p,q)=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2} \tag{5}$$

Další možností je spočítat kosinovou podobnost těchto vektorů. Výsledné číslo udává míru podobnosti mezi vektory *A* a *B* v prostoru skalárního součinu na základě kosinu úhlu, který svírají, viz. rovnice (6).

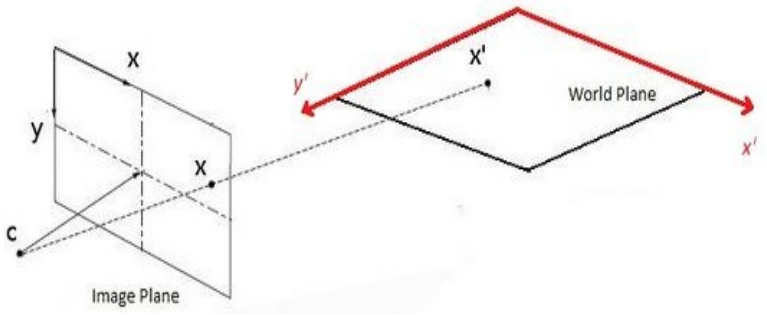
$$similarity=\cos(0)=\frac{A*B}{\|A\|*\|B\|}=\frac{\sum_{i=1}^{n}(A_i*B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2}*\sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{6}$$

2.5 Homografie

Homografie je projektivní transformací mezi dvěma zadanými prostory stejné dimenze, přičemž každý v bod x_i v prvním prostoru je mapován na odpovídající bod x_i' v prostoru druhém a naopak. Pokud oba prostory reprezentují roviny, mluví se o 2D homografii, viz. obr. 6. Ta najde uplatnění při transformaci významných bodů mezi fotografiemi, kdy se jejich obsah liší pouze v úhlu pohledu na scénu. Pak lze tyto korespondence využít pro logické sloučení jejich obsahu tak, aby utvořili jednu souvislou fotografii. Samotné mapování probíhá pomocí transformační matice. Konkrétní zápis lze zapsat jako $x_i' = Hx_i$, kde *H* je transformační maticí o rozměru 3x3 viz. rovnice (4) [9] [10] [11] [12].

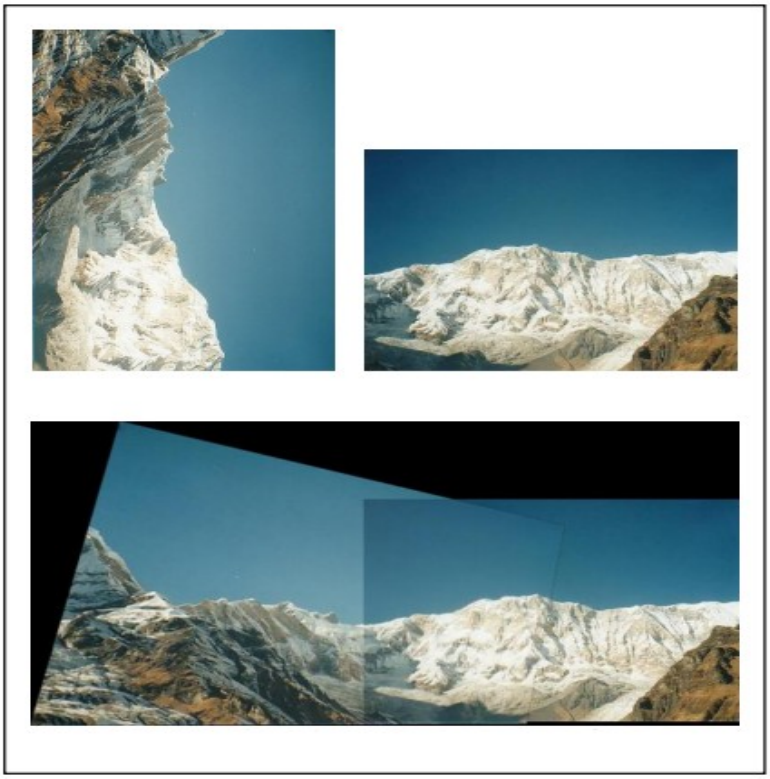
$$\boldsymbol{x}_i' = H\boldsymbol{x}_i = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i' \\ y_i' \\ w_i' \end{pmatrix} \tag{4}$$

Homografii lze také uplatnit při hledání podobnosti fotografií na základě jejich obsahu. Jako příklad jsou uvedeny fotografie *A* a *B*, mezi nimiž má být nalezena podobnost. Nejprve je detekován a extrahován jejich obsah. Dále jsou vyhledány párové dvojice deskriptorů. Tzn. množina deskriptorů, jejíž každý člen má svého podobného zástupce *A* a zároveň v *B*.



Obrázek 6: Ukázka homografie v 2D prostoru [11].

Na základě těchto dvojic se vypočítá transformační matice *H*. Poté dojde k transformaci významných bodů v *A* na odpovídající body v obsahu *B*. Visuálně tak dojde k překrytí fotografie *B* fotografií *A* v místech, kde se obsah mezi *A* a *B* shoduje obr. 7. Podobnost je hledána v místech překrytí, kdy by se podobnější vzorky měli překrývat větší plochou, než vzorky mezi nimiž podobnost není.

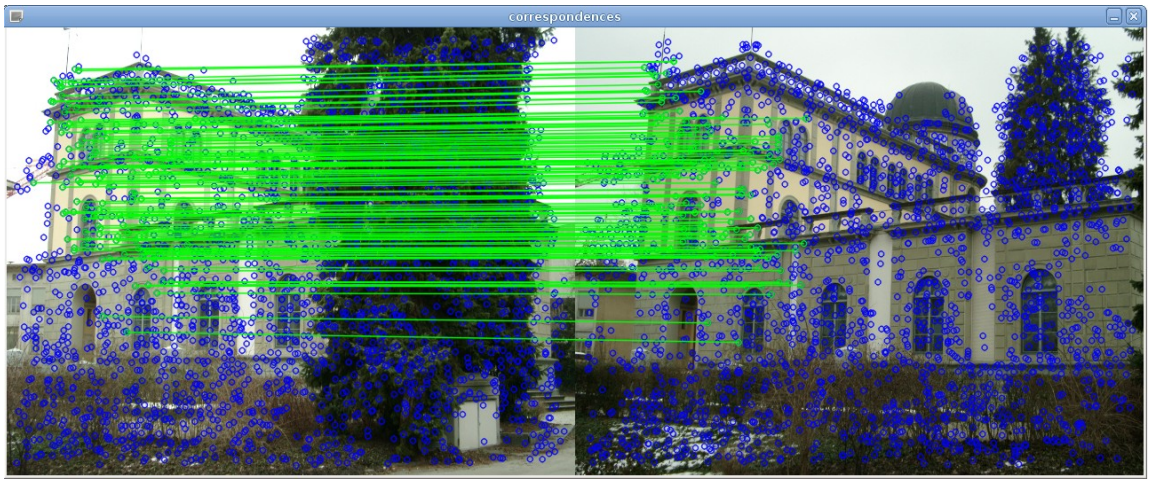


Obrázek 7: Příklad transformace pomocí homografie [12].

Tato plocha pak odpovídá nějakému počtu významných bodů tzv. *inliers*. Body, jenž do této oblasti nepatří se nazývají *outliers*. Za hledaný výsledek se považuje poměr *inliers* k celkovému počtu klíčových bodů fotografie *A*. Je tak tedy řečeno do jaké míry se fotografie *B* podobá fotografii *A*.

Pro vyhledání podobnosti však není nutné transformovat všechny body fotografie. Postačí pouze ty, jejichž deskriptory jsou párové. Celý postup by se tak dal shrnout do několika bodů:

1. Detekce a extrakce deskriptorů fotografií *A* a *B*.
2. Vyhledání párových dvojic deskriptorů mezi *A* a *B*.
3. Výpočet homografie mezi *A* a *B* na základě vyhledané množiny.
4. Transformace párových bodů v *A* pomocí matice homografie do prostoru obsahu *B*.
5. Výpočet *inliers* a *outliers* bodů.
6. Výpočet podobnosti.



Obrázek 8: Nalezené korespondence mez dvěma fotografiemi budov za pomocí homografie. Zeleně jsou spojeny inliers body.

3 Současné aplikace a algoritmy

V dnešní době existuje více než jedna cesta k řešení problému vyhledávání fotografií. Komerční i nekomerční sféra tak obsahuje mnoho funkčních používaných aplikací. V této kapitole se popisuji některé z používaných programů pro vyhledávání podle obsahu a zároveň uvádím některé zajímavé novinky.

3.1 Používané komerční a nekomerční aplikace

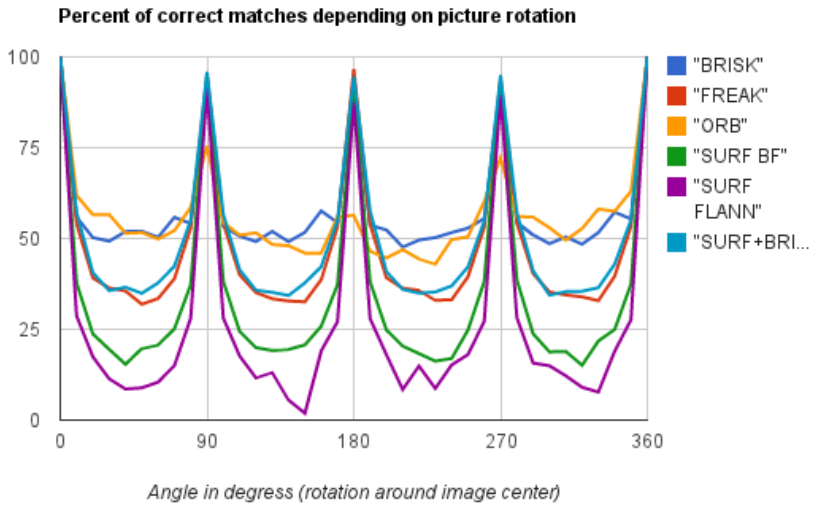
Google images je nejznámějších z používaných aplikací pro vyhledávání obrázků či fotografií napříč internetem. Tato služba se poskytuje zdarma. Od roku 2009 je schopná vyhledávat podobné fotografie dle zadaného vzoru na základě obsahu. Vzory mohou být zadávány URL adresou, či jednoduše cestou k fotografii, která je lokálně uložena v počítači. Google images získává průběžně metadata o obrázcích z webových stránek. Ty jsou následovně ukládány a indexovány do rozsáhlé databáze. Pokud přijde dotaz na vyhledání vzorové fotografie, prochází se indexy a porovnávají dotazy s uloženou informací na daném indexu. Výsledky se pak vrací v pořadí relevance.

Adobe Photoshop Elements 11 je nástrojem pro lokální editaci, organizaci a vytváření obrázků a fotografií. Za zajímavou součást se považuje, vzhledem k této práci, vyhledávání podobných fotografií. Uživatel předloží vzor, podle kterého aplikace uskuteční vyhledávání. Poté je vytvořen index v závislosti na podobnosti se vzorem. Následně vrátí seřazené výsledky dle relevance a u každé fotografie vypíše procentuální shodu. Ukázka aplikace zobrazena na obr. 9. Vyhledávání je možné doplnit o filtry pro specifikaci užší množiny výsledků např. barva, tvar, typ souboru, selekce regionu atd.

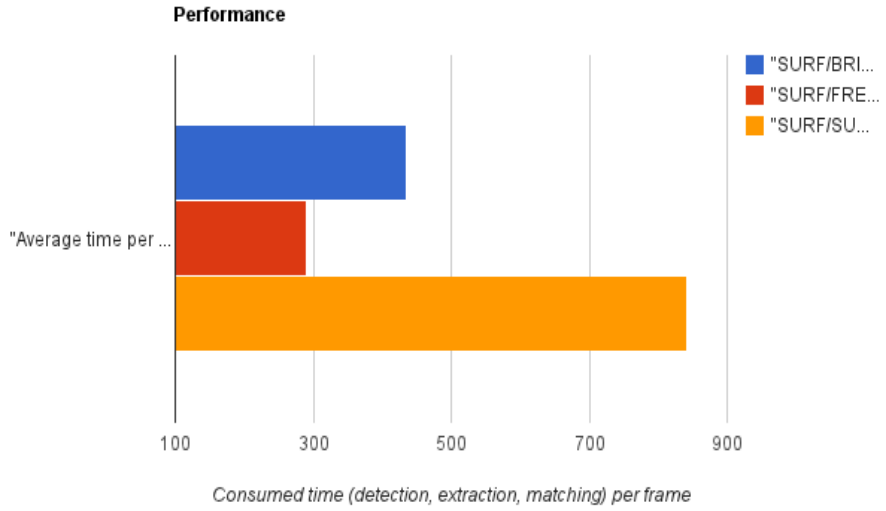
Obrázek 9: Výsledky při použití Adobe Photoshop Elements 11 [13].

3.2 Nové algoritmy

BRISK (Binary Robust Invariant Scalable Keypoints) je novým algoritmem v oblasti detekce a extrakce významných bodů v obraze. Algoritmus je robustní. Vyznačuje se dobrou invariancí vůči změně měřítka a rotací. Jeho autoři prohlašují, že výsledná výpočetní rychlost při použití tohoto algoritmu je řádově vyšší, než u standardně používaného SURF, viz. kapitola 2.1. Jeho rychlost spočívá v použití nového detektoru založeného na FAST (Features from Accelerated Segment Test) v kombinaci s deskriptory jako binárními řetězci s využitím porovnání intenzity významných bodů a cíleným vzorkováním jejich okolí [14].

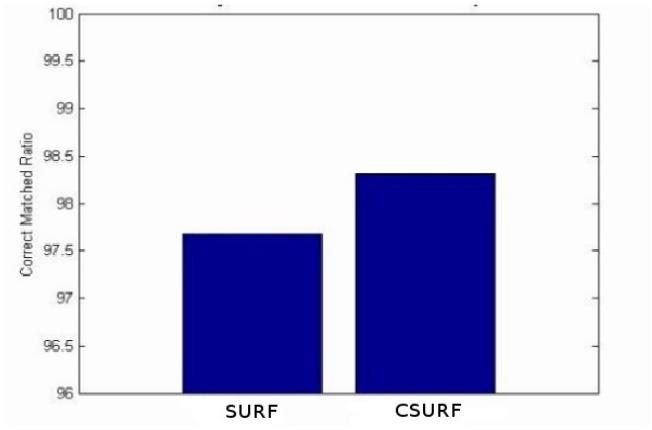


Obrázek 10: Procenta úspěšného vyhledání při použití různých metod a změně rotaci obrázku. Jsou zde uvedeny výsledky i pro algoritmus BRISK [15].



Obrázek 11 Graf spotřebovaného času při detekci a extrakci. Modře zobrazen spotřebovaný čas pro algoritmus BRISK [15].

C-SURF (Colored Speeded Up Robust Features) je vylepšenou formou algoritmu SURF. Princip C-SURF je podobný standardní algoritmu SURF, viz. kapitola 2.1.1. Rozdíl mezi nimi hraje ohled na barevnou složku analyzovaného obsahu. C-SURF tento obsah respektuje a do své detekce zahrnuje barevnou složku okolí detekovaných významných bodů. Vznikají tak deskriptory nesoucí barevnou informaci o velikosti112 dimenzí[16].



Obrázek 12: Test vyhledání při použití SURF a CSURF na barevném obsahu bez změny rotace [16].

4 Návrh aplikace pro vyhledávání fotografií

Jak bylo nastíněno v úvodu tak výsledkem této práce má být aplikace umožňující vyhledávání fotografií podle vzoru. Na několika řádcích níže rozeberu, co by aplikace měla umožňovat a co je zapotřebí pro její korektní provoz. Součástí tohoto popisu je pak detailnější náhled na důležité části.

Základ aplikace by měl spočívat ve zpracování základních zdrojů, což jsou fotografie. Pro získání některých později důležitých vstupů je nutné stanovit trénovací sadu. Tou bude složka obsahující fotografie. Sady reprezentují potenciální množinu fotografií, které budou prohledávány na shodu obsahu ze zadaným vzorem. Uživatel může mít více trénovacích sad a jeho, rozhodnutí ovlivní, zda použije některé pouze jako podklad pro tvorbu visuálních slovníků, nebo jsou vypočteny jejich BoW pro pozdější vyhledávání. Každá sada bude mít pro identifikaci klíčové pojmenování. Uživatel může obecně za vzor pro vyhledávání použít jakoukoliv fotografii, která nemusí být součástí trénovací sady. Vzorová fotografie se pro přehlednost uživateli zobrazí v grafickém výstupu.

U jednotlivých fotografií ze sady proběhne pomocí algoritmu SURF analýza obsahu, konkrétně se zaměřením na deskriptory a klíčové body. Dalším fáze bude vytvoření visuálního slovníku. Ten by měl být vytvořen na základě informací z předešlé analýzy zvolené sady. Velikost takové sady by pro optimální výsledky měla být v řádu tisíců fotografií. Konkrétnější atributy pro jeho tvorbu jsou ve stanoveném rozsahu nechány na uživateli. Z informací získaných ze sady a slovníku by došlo k reprezentaci fotografií na jednotlivé BoW. Pro lepší manipulaci budou BoW sdružovány do pojmenovaných množin (matic), kde jedna množina souvisí s obsahem celé jedné trénovací sady. V případě vzoru je množina tvořena pouze jedním záznamem. Klíčová vypočtená data budou ukládána do lokální databáze, k jejímuž vytvoření dojde automaticky po spuštění aplikace v lokálním souboru. Pokud by existovala v tomto souboru databáze z předešlých výpočtů, nebude utvářena znovu a její data budou nabídnuta uživateli. Uživatel tak může vynechat jednotlivé kroky zpracování sady, či celou etapu výpočtu podkladů a rovnou přejít k vyhledávání. Výjimkou v ukládání jsou pouze informace získané ze vzoru. Ty prochází výpočtem BoW za pomocí stejného slovníku jako byl použit pro sadu, ale jeho data se neukládají do databáze. Vyhledávání bude založeno na metodě kosinové podobnosti dvou vektorů, doplněno o volitelný výpočet homografie.

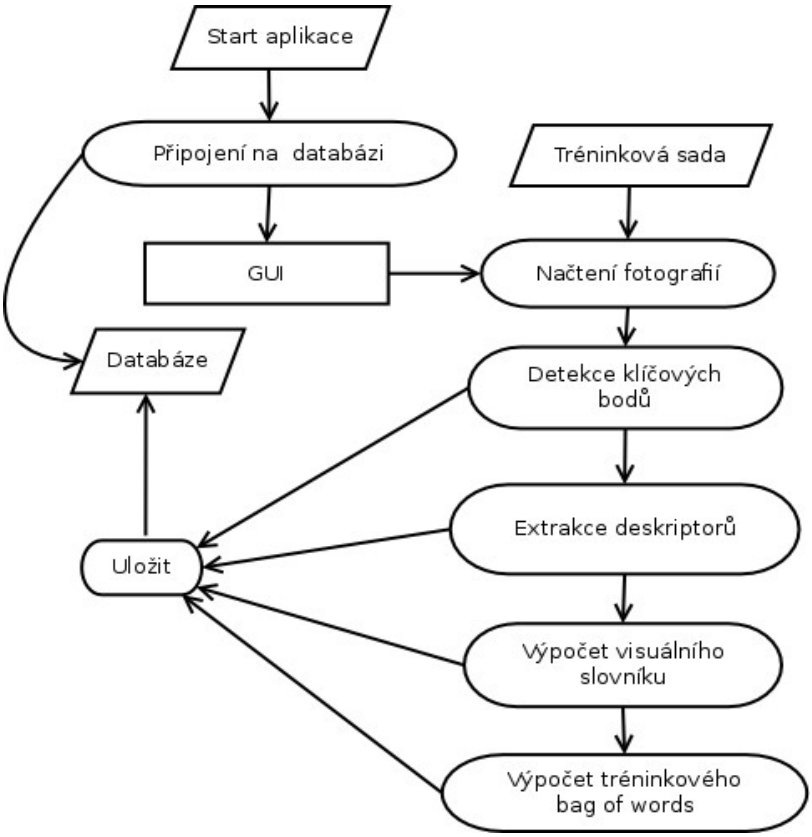
Z logického hlediska se aplikace v návrhu rozdělí na dvě části, mezi kterými existuje závislost. Každá je charakterizována jiným druhem výstupu. Obě sdílí stejný počáteční stav a jejich průběh může být paralelní. Jejich jednotlivý rozebraný návrh pomocí stavových diagramů je zaznamenán níže na obr. 13 a obr. 14.

- *1. Vytvoření vstupů pro vyhledávání*

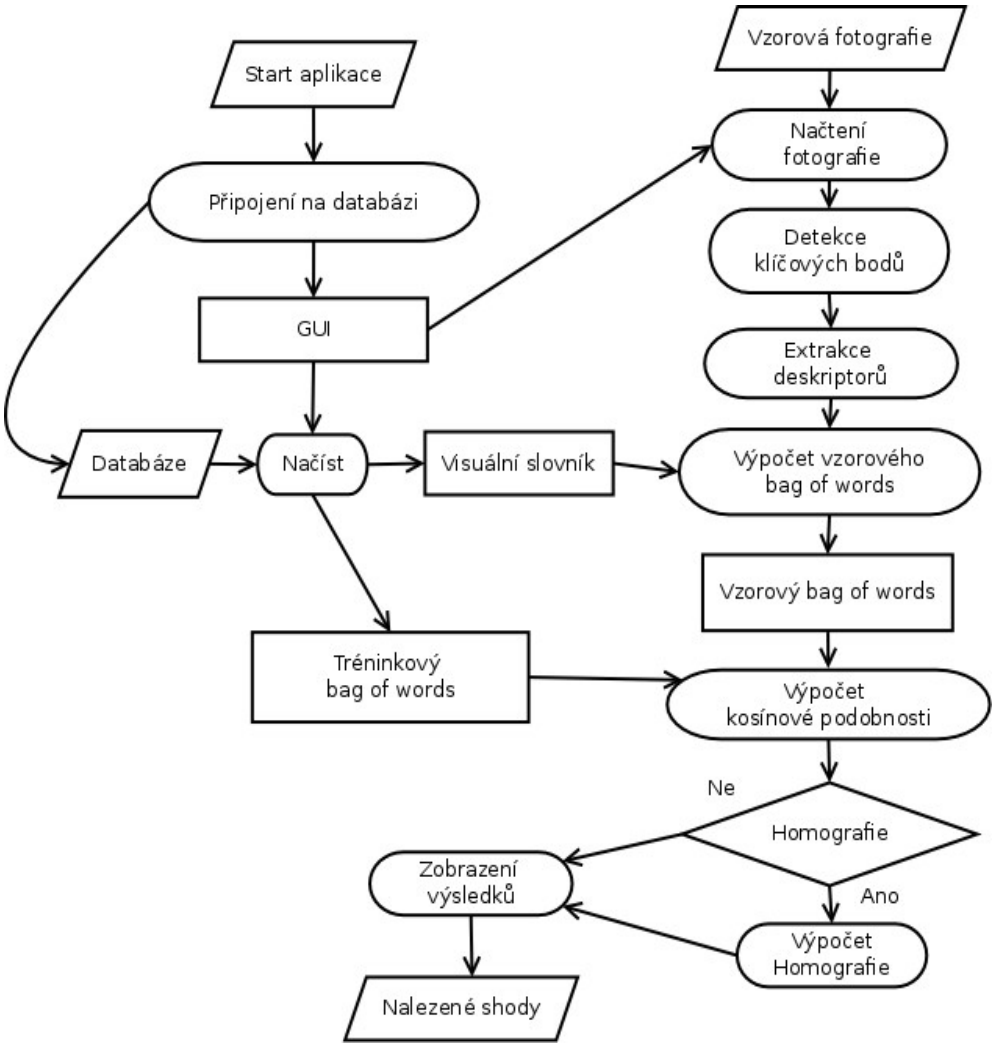
  Tato část načítá a vytváří nutné podklady pro vyhledávání dle vzoru. Jedná se tak o zpracování vstupní tréninkové sady fotografií na výstupy v podobě visuálního slovníku, BoW a dalších klíčových dat. Tyto výstupy jsou poté uloženy k pozdějšímu užití.

- *2. Výpočet vzorových dat a vyhledávání*

  Tato část předpokládá již existující vstupní podklady minimálně jednoho z předešlých průběhů části 1. Vyhledává podobné fotografie dle zadaného vzoru a výsledky zaznamenává na výstup. Dále se pak stará o vzorová data, která jsou vytvořena vždy před začátkem vyhledávání. Vyhledávání může být jednoduché nebo může být rozšířené o další metodu, která je spouštěna volitelně.



Obrázek 13: Vytvoření vstupů pro vyhledávání.



Obrázek 14: Výpočet vzorových dat a vyhledávání.

4.1 Získání klíčových informací

Získání informací začíná detekcí klíčových bodů v obraze a následném extrahování jejich deskriptorů. Pro tyto účely jsem zvolil algoritmus SURF. Hlavním důvodem jeho použití je dobrá rychlost výpočtu. To je příznivé z hlediska zpracování většího počtu fotografií.

Po extrakci následuje natrénování visuálního slovníku a BoW. Slovník pro svoje vytvoření používá vyextrahované deskriptory a zachovává jejich původní rozměr (konstantně 128 dimenzí). Vypočítá se pomocí metody shlukování středů k-means. Se slovníkem úzce souvisí BoW. Respektive BoW vzniká za pomocí visuálního slovníku. Jeho tvorba je založená na vyhledávání ve vícerozměrném prostoru, z toho plyne využití vhodné datové struktury. Pro tyto účely je slovník umístěn do stromové struktury k-dimenzionálního stromu (kd-tree). Úlohu prohledávání toho stromu pak zastává hledání nejbližšího souseda. Aby se však docílilo vyšší rychlosti při hledání ve stromě, je

využito paralelního prohledávání. Strom se náhodně rozdělí na určený počet menších podstromů, které jsou prohledávány současně.

4.2 Vyhledávání fotografií

Možnost jak vyhledat podobné fotografie plyne z určení podobnosti mezi vzorovým vektorem *A* a vektorem *B* z trénovací sady. To znamená, že hledanou hodnotou je výpočet kosinové podobnosti mezi těmito vektory. Jelikož hodnoty tvořící vektory budou vždy nulové či kladné, očekávaný výsledek kosinové podobnosti jedné fotografie se bude pohybovat v rozpětí <0,1>. Čím více se výsledek blíží k hodnotě 1, tím je relevantnější k zadanému vzoru. Tento zmíněný postup doplňuje nepovinný samostatný výpočet homografie mezi dvěma fotografiemi. Ta se počítá nezávisle na BoW a uživatel ji může nebo nemusí do vyhledávání zařadit. Hodnota podobnosti homografie se pohybuje také v rozsahu <0,1>. Pokud uživatel zvolí hledání pomocí obou metod, tak jsou jejich výsledky sečteny a rozpětí se pohybuje v rozsahu <0,2>. Všechny konečné ohodnocení jsou na konci seřazeny dle nejvyššího algoritmem quicksort.

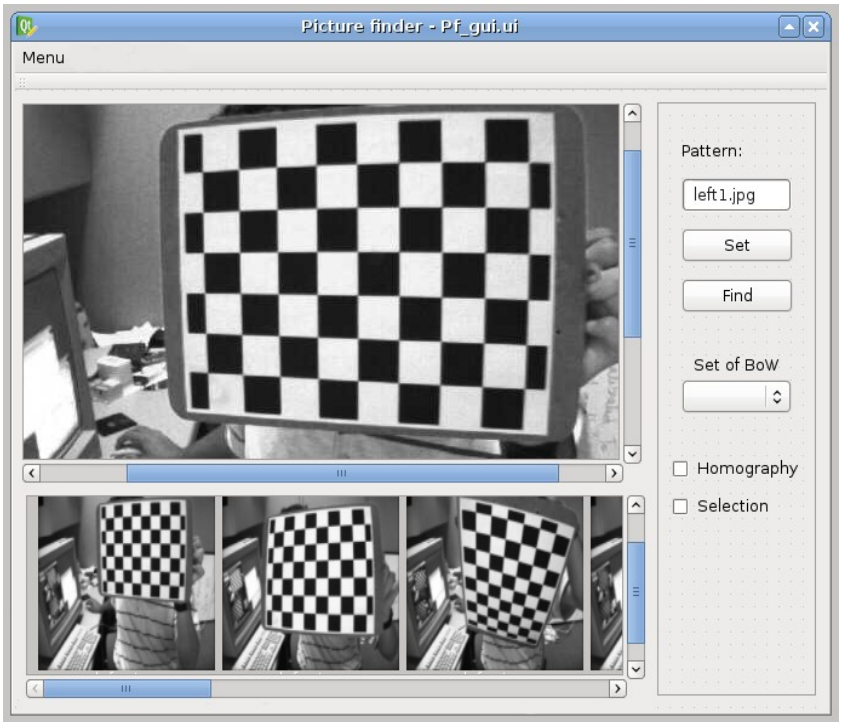
4.3 Vytvoření databáze

Zvážíme-li množství fotografií, které by mělo být prohledáno a s tím spojené výpočetní úkony, dojdeme k jasnému závěru. Bylo by časově náročné a nevýkonné některé tyto operace provádět znovu. Zvláště pokud se jedná o extrakci a výpočet zdrojových dat. Pokud uživatel aplikace tuto extrakci za určitých podmínek nad tréninkovými daty provede, neměl by být nucen za stejných podmínek a stejnou množinou dat tuto operaci opakovat. Jedná se hlavně o klíčové informace jednotlivých fotografií, BoW a visuálního slovníku. K tomuto účelu je vhodná databáze, avšak s tím je spojený nutný provoz serveru a síťového připojení. Přítomnost něčeho takového není vhodná.

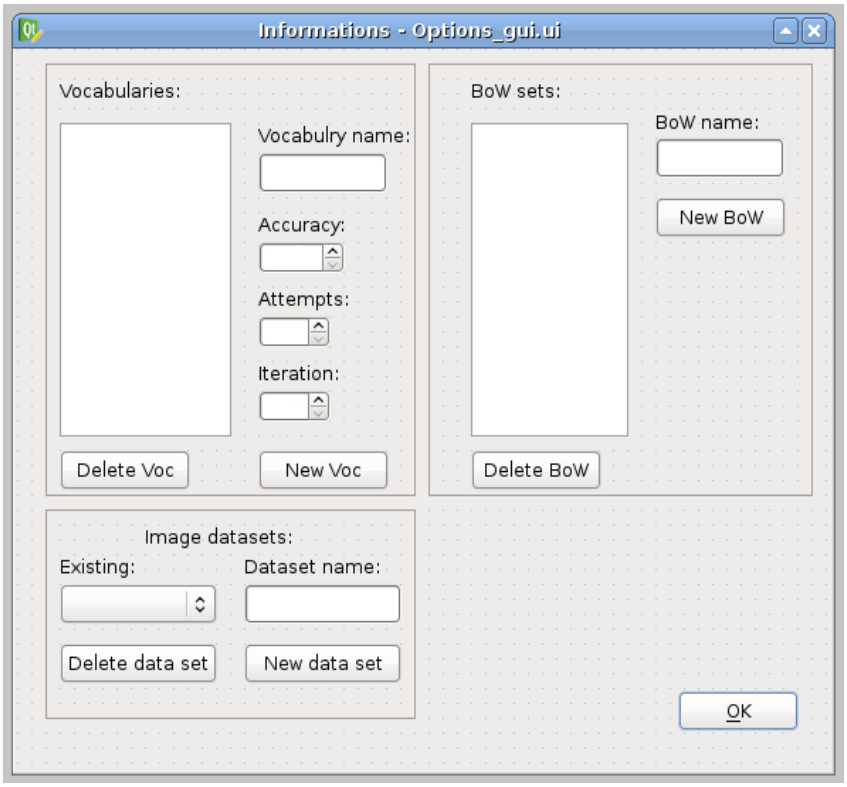
4.4 Grafické rozhraní

Grafické rozhraní by mělo být pro uživatele srozumitelné a jednoduché. Mělo by podporovat editaci databáze zdrojových dat, nastavení některých specifických údajů pro vyhledávání. V potaz se bere i volba z více visuálních slovníků a BoW. Z časových důvodů jsem se rozhodl nenavrhovat příliš detailně propracovanou grafiku. Z hlediska vzorových vstupních a vyhledaných výstupních dat musí existovat možnost si je prohlédnout v grafické podobě. Uživatele nemusí nutně zajímat fotografie jako celek. Byla navržena možnost selekce volitelných čtvercových regionů vzoru. Tím se omezí vyhledávání jenom na selektovaný region. Uživatel má potom možnost vybrat pouze tu část fotografie, která ho zajímá. To může zrychlit rychlost a relevantnost celkového výsledku. Grafický

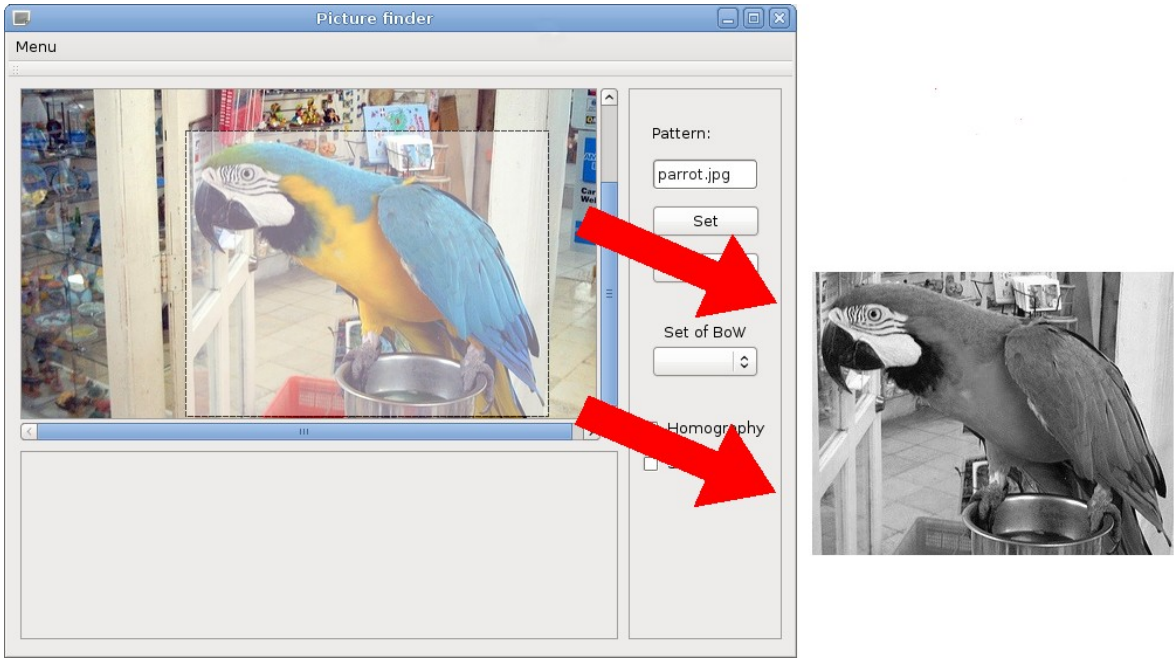
návrh rozhraní je zobrazen na obr. 15 a obr. 16. Na obr. 17 je zobrazen návrh selekce z části fotografie.



Obrázek 15: Hlavní okno aplikace.



Obrázek 16: Sekundární okno aplikace.



Obrázek 17: Selekce vybraného regionu v GUI a převod na jeho vnitřní reprezentaci.

5 Implementace aplikace

Tato kapitola by měla popsat realizaci jednotlivých etap zmíněných v návrhu. Dále shrnout prostředky použité při realizaci programové části aplikace.

Pro implementaci jsem si zvolil jazyk C++. Samotná aplikace byla napsána v programovacím prostředí Qt Creatoru verze 2.6.0. Daný jazyk a prostředí jsem si zvolil hlavně kvůli předchozím zkušenostem s těmito prostředky. Dále pak kvůli možnostem objektového návrhu, podpoře knihovny OpenCV pro C++ a vytváření grafického prostředí pomocí knihoven Qt. Se zvoleným implementačním jazykem souvisí i volba databáze. Výběr padl na databázi sqlite, která nepotřebuje pro svůj provoz síťové připojení. Zároveň také obsahuje aplikační rozhraní v jazyce C.

5.1 Použití dostupných knihoven

Jak jsem nastínil tak je v práci hojně využita knihovna OpenCV verze 2.4.2. Jedná se o rozsáhlou knihovnu pro práci s grafickými daty. Implementuje celou škálu algoritmů, struktur, funkcí s tím spojené a poskytuje plně objektově rozhraní pro jazyk C++. Pro vznik databáze je použito knihovny sqlite3, jenž poskytuje sadu funkcí pro vytvoření a práci s binárních souborem tvořící databázi. Dále je pak využito knihoven boost/serialization pro serializaci objektů na binární data.

5.2 Extrakce a výpočet vstupních dat

Extrakce a výpočet zahrnuje celé načtení vstupních dat pro aplikaci. Na začátku je každá fotografie v určeném adresáři načtena funkcí imread() do objektu třídy Mat, který zprostředkovává uložení číselné reprezentace obrazu uložené v matici. Z těchto objektů se algoritmem SURF, implementovaného knihovními funkcemi OpenCV, detekují klíčové body a z nich jsou extrahovány deskriptory. Ty jsou taktéž ukládány do samostatných objektů třídy Mat, ale s konstantními šířkou matice 128 sloupců. Klíčové body, deskriptory, název a souborová cesta k fotografii jsou předány funkcím databáze k jejich uložení.

V další části se deskriptory získané při extrakci načítají z databáze do paměti v podobě jedné souvislé matice. Poté probíhá výpočet visuálního slovníku pomocí funkce kmeans() z knihovny OpenCV. Tato funkce implementuje shlukování středů matice. Pokud jde o nastavení parametrů, je nastaven počet středů, které má funkce vypočítat. Ten je úměrný počtu slov, jenž chceme aby slovník obsahoval. Dalším parametrem jsou ukončovací kritéria. Ty se zadávají pomocí struktury TermCriteria, kterou říkáme kdy má být výpočet ukončen. Do ní stanovuje uživatel pomocí GUI před výpočtem slovníku číselná kriteria, jako je přesnost, počet iterací a počet pokusů. Defaultní nastavení

obsahuje ukončující podmínka CV_TERMCRIT_EPS + CV_TERMCRIT_ITER a inicializace středů KMEANS_PP_CENTERS. Říkáme tím, že algoritmus ukončí stanovený počet iterací nebo pokud se dosáhne požadované přesnosti. Inicializace středů probíhá dle metodiky Arthura a Vassilvitskiho.

Pokud existuje visuální slovník, je načten z databáze a použit pro výpočet BoW. Jádro BoW je založeno na prohledávání k-dimenzionálního stromu vybudovaného ze slovníku. Strom realizuje třída flann::Index opět z knihovny OpenCV. O samotné prohledávání se stará metoda knnSearch() (hledání nejbližšího souseda). Nalezené shody jsou cyklicky přepočteny na výskyt jednotlivých slov v histogramu. Výsledek je přidán jako řádek reprezentující jeden BoW do matice.

5.3 Kategorizace dat

Jak jsem zmínil v návrhu, viz. kapitola 4, je účelné některá data ukládat pro opětovné výpočty. Pomocí knihovny sqlite3 jsem implementoval třídu, jenž představuje abstraktní vrstvu pro práci s lokální databází sqlite. Databáze podporuje číselné i textové datové typy. Bohužel postrádá datové struktury typu pole, které jsem chtěl využít. Jako náhrada byl stanoven datový typ BLOB, který realizuje ukládání objektů, struktur či souborů jako binární data. Následně je však nutná podpora serializace.

Textové informace jako názvy, cesty souborů a pod. ukládám jako datový typ TEXT. Pro ukládání objektů obsahující deskriptory, slovníky, klíčové body či BoW využívám datového typu BLOB. Všechny tyto objekty jsou charakterizovány maticemi, jenž obsahují klíčová data. Ty procházejí před před vložením do databáze binární serializací. Stejně tak při načtení de-serializací na původní objekt. Obě metody jsou realizovány pomocí knihoven boost/serialization a hotového řešení podle Christopha Heindla [17]. Bez de-serializace není umožněn přímý přístup ke konkrétním datům. Databáze je realizována v souboru Info.db, který se vytváří automaticky po spuštění aplikace.

5.4 Přehled vytvořených tříd

Tato kapitola slouží jako přehled vytvořených tříd při implementaci aplikace. Každá z nich obsahuje stručný popis její funkce.

- **Error** - Třída realizující systém chybových kódů v aplikaci. Chybu reprezentuje kód a zpráva.

- **Vocabulary** - Třída realizující visuální slovník. Obsahuje metody pro výpočet visuálního slovníku pomocí vstupů z databáze. Dále realizuje uložení a načtení slovníku z databáze.

- **Bow** - Třída realizující sadu bag of words. Implementovány jsou metody pro výpočet BoW pomocí vstupů z databáze a výpočet vzoru z datové struktury. Realizováno je uložení i načtení sady do databáze. Dále obsahuje metodu Find pro určení kosinové podobnosti a statickou metodu *CheckHomography* pro určení homografie;

- **Strike** - Třída pro záznamy výsledků vyhledávání. Udržuje výsledky a dohledává případné cesty k výsledným souborům.

- **Mat_database** - Třída reprezentující abstraktní vrstvu nad databází. Implementovány jsou zde metody pro načítání,mazaní a ukládání položek v databázi. S tím je spojená realizace dotazů a vytváření tabulek.

- **Pf_gui** -Třída implementuje hlavní grafické okno aplikace, kde je uživatelem řízeno vyhledávání. Jsou zde také metody pro grafické zobrazení vzoru a výsledných nalezených fotografií.

- **Options_gui** - Třída implementuje sekundární grafické okno. Obsahuje uživatelské rozhraní pro vytváření a mazání podkladů pro vyhledávání.

- **PatternLabel** - Třída implementuje grafický rám pro zadaný fotografický vzor. Obsahuje metody pro selekci označeného regionu.

6 Dosažené výsledky

Výsledek této práce se zakládá na splnění funkčnosti vytvořené aplikace. Míru do jaké se shodují zadané předpoklady s některými cíli je ověřeno testováním v následující kapitole.

6.1 Návrh testování

Cílem testování je ověřit funkčnost aplikace z hlediska vyhledávání nad danou tréninkovou sadou a sledovat míru úspěšnosti v závislosti na hlavním datovém podkladu, kterým je visuální slovník. Dále je sledován rozdíl úspěchu mezi dvěma variantami vyhledávání. Tzn. použití základní varianty s kosinovou podobností, nebo kosinovou podobnost doplněnou o samostatný výpočet homografie.

Pro účely testování byla zvolena testovací sada ZuBuD [18], která obsahuje 1005 fotografií, které jsou rozděleny do 201 pojmenovaných množin budov. Každá množina obsahuje 5 podobných fotografií stejné budovy nafocené z různých úhlů. Vyhodnocení úspěšnosti vyhledávání pak spočívá v počtu relevantních nálezů, při volbě výstupu 5 nejlepších vyhledaných výsledků. Protože tato sada byla zbudována právě pro testování hledání podobných fotografií, rozhodl jsem se ponechat vhodně navržené pojmenování souborů navržené autory a to ve tvaru objekt[číslo_objektu].pohled[číslo_pohledu].přípona. Abych doplnil různorodost testovací sady ZuBuD, tak jsem navrhl její obohacení o dalších 495 různých, náhodně vybraných fotografií z testovací sady VOC2012 [19]. Celkově tak pro testování bylo použito 1500 fotografií.

Dále je testování doplněno o sledování růstu spotřeby datového prostoru lokální databází na disku. Toho je docíleno postupnou inkrementací počtu uložených pokladů do databáze, jako jsou vstupní informace z analyzovaných trénovacích sad. Zvolený inkrement pro test byl 150 fotografií.

Obrázek 18: Ukázka jedné množiny podobných fotek ze sady ZuBuD [18].

6.2 Výsledky testování

Realizace veškerých testů proběhla na osobním notebooku značky MSI CX620 s procesorem Intel Core i3-330M a DDRIII 4G RAM se systémem Debian GNU/Linux 6.0.6. Testovaní vyhledávání bylo navrženo s ohledem na výsledky ovlivněné tvorbou visuálního slovníku, jak bylo zmíněno v kapitole 6.1. Již při počátečních pokusech o vytvoření podkladů pro vyhledávání fotografií, jsem došel k jednoznačnému závěru. Vytváření slovníku je časově nejnáročnější etapou v celkovém průběhu aplikace. Proto byla zaznamenána doba jeho výpočtu. Dále je důležitá jeho velikost (počet středů), která ovlivňuje výstupní výsledky. Při těchto testech byly vytvářeny různě velké slovníky s konstantními vstupními atributy pro jejich tvorbu. Sledovanou hodnotou se stala průměrná procentuální úspěšnost vyhledávání nad celou testovací sadou v závislosti na velikost slovníku u jednotlivých variant pro vyhledávání.

Slovník	Počet slov	Doba výpočtu
Voc1	325	2h 49min 36s
Voc2	750	6h 34min 33s
Voc3	1500	13h 00min 41s
Voc4	2500	22h 53min 56s

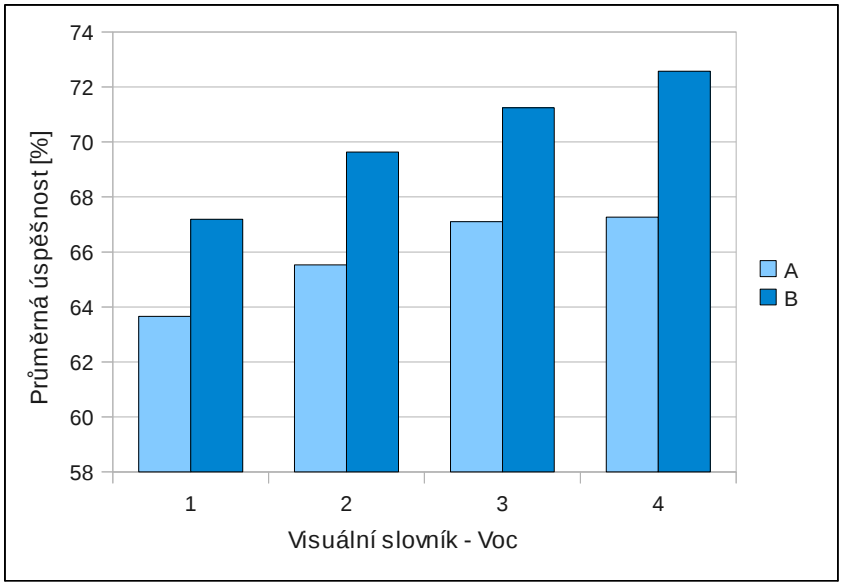
Tabulka 1: Vytvořené slovníky s atributy iterace = 100, přesnost 0.01, počet pokusu 3 .

Slovník	Průměrná úspěšnost [%]	Varianta
Voc1	63,66	A
	67,19	B
Voc2	65,53	A
	69,63	B
Voc3	67,1	A
	71,24	B
Voc4	67,27	A
	72,57	B

Tabulka 2: Přehled naměřené průměrné úspěšnosti

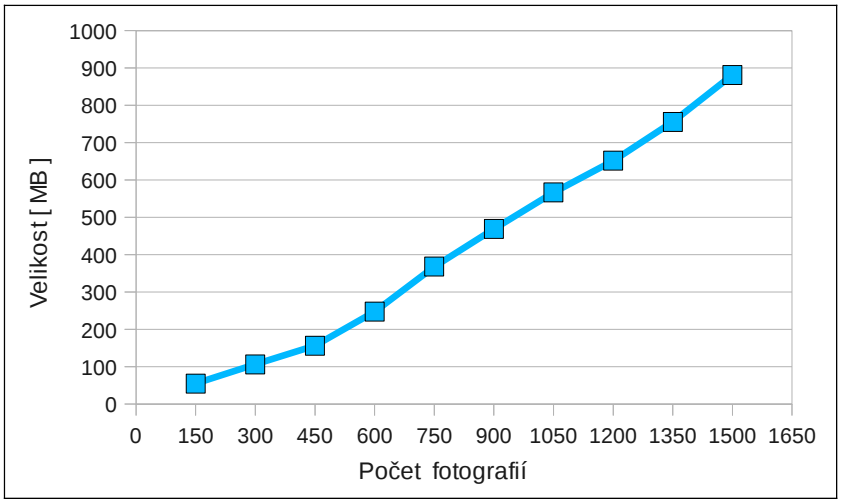
A - kosinová podobnost

B - kosinová podobnost + homografie



Obrázek 19: Graf průměrné úspěšnosti vyhledávání. Zobrazeny jsou obě možné varianty vyhledávání, při použití různě velikých visuálních slovníků

V dalším průběhu testovaní jsem se zaměřil na velikost databáze. Ta s přibývajícími podkladovými daty a informacemi získaných z analyzovaných fotografií rostla. Podle návrhu jsem testovací sadu rozdělil na 10 dílů po 150 fotografiích a u každého z nich extrahoval pouze klíčová data (deskriptory, klíčové body atd.). Po každé extrakci bylo provedeno měření aktuální velikosti souboru databáze. Výsledný průběh byl pak zanesen do grafu.



Obrázek 20: Graf spotřeby datového prostoru databází

6.3 Rozšíření aplikace

V této kapitole popisuji možný budoucí vývoj aplikace. Tu lze z hlediska funkčnosti obohatit hned v několika směrech. Prvním možným vylepšením je nahrazení stávajícího algoritmu pro detekci a extrakci SURF rychlejším algoritmem BRISK. Od verze 2.4.3 je totiž BRISK součástí knihoven OpenCV a ve verzi 2.4.2 která byla zvolena na začátku vývoje jeho implementace chybí. Dalším vylepšením se stává distribuce výpočtu slovníku do několika paralelních částí, což by přispělo k celkové rychlostí jeho vytvoření.

Dále navrhuji několik změn z hlediska GUI. Mělo by být více senzitivní, příjemnější vzhled. Další změna je posun oznámení o indexaci do oblasti pracovních panelů a doplnění aplikace o autodetekci přidaných podkladů. Aplikace by tak registrovala nově přidané fotografie a po překročení zvoleného limitu by začala počítat nové podklady. Zajímavým nápadem by bylo zaměřit se na vyhledávání nejen podle obsahu fotografie, ale i podle zapamatovaného obrysu objektu, který by mohl na fotografii být. Uživatel by tak v jednoduchém editoru nakreslil přibližný obrys hledaného objektu a v kombinaci se selekcí by vyhledával fotografie s obsahem podobným tomuto obrysu.

7 Závěr

Cílem této práce bylo vytvořit aplikaci pro vyhledávání fotografií podle obsahu. Podle nastudovaných metod vyhledávání a zpracování fotografií jsem tuto aplikaci navrhl a implementoval Ze známých metod pro její tvorbu je použita detekce a extrakce obsahu fotografie pomocí algoritmu SURF. Dále jsem ve své práci využil vytváření visuálního slovníku pomocí metody shlukování k-means. Pro vyhledávání jsem implementoval popis fotografií jako bag of words, hledání pomocí kosinové podobnosti vektorů a homografie.

Aplikace dokáže vyhledávat fotografie ze vstupní tréninkové sady, podle zadaného vzoru. K tomuto účelu aplikace vytváří datové podklady, které ukládá do lokální přenosné databáze. Uživatel má přístup k jednoduchému grafickému rozhraní, skrze které aplikaci ovládá. Má možnost výběru, tvoření či mazání těchto datových podkladů, včetně nastavení některých důležitých atributů pro jejich tvorbu. Dále má možnost nastavení libovolné fotografie jako vzor pro vyhledávání, který se zobrazí v uživatelském prostředí. Zde se nabízí možnost selekce jen určeného regionu oblasti zájmu, např. určitý objekt ve vzorové fotografii. Aplikace umožňuje vyhledávat dvojím způsobem a to pomocí kosinové podobnosti, která je základní metodou vyhledávání nebo rozšířeným vyhledáváním se samostatným výpočtem homografie. Konečné nalezené výsledky jsou v pořadí relevance a zmenšeném formátu uživateli graficky zobrazeny.

Aplikace byla testována z hlediska průměrné úspěšnosti vyhledávání v závislosti na velikosti visuálního slovníku, který byl použit pro výpočet BoW. V testech dosahovala nad vytvořenou datovou sadou v rozšířené variantě až 72,52% úspěšnosti což je o 5,3 % lepší oproti základní variantě. Z testů je jasně vidět, že velikost slovníku kladně ovlivňuje procentuální úspěšnost. Optimálního výsledku se docílí již od velikosti slovníku 1500 slov. Další zvětšování slovníku sice přinese mírné zlepšení výsledků, avšak za cenu velikého nárůst spotřebovaného času pro jeho výpočet. V druhé části testů jsem se věnoval sledování spotřeby datového prostoru databází. Z výsledné křivky je vidět, že zvětšování databáze v závislosti na příbytku klíčových dat je lineární. Lze tak očekávat úměrné zvětšování databáze s počtem klíčových dat fotografií analyzovaných tréningových sad, které bude dosahovat do řádu jednotek až desítek GB.

Vylepšení aplikace v budoucnosti by spočívalo ve výměně algoritmu SURF za algoritmus BRISK, což by mohlo zvýšit rychlost zpracování klíčových dat. Dále rozšířit GUI o nové prvky a funkci hledání fotografie podle obrysu objektu.

Literatura

[1] JURÁNEK, R. *Extrakce obrazových příznaků* [online]. Ústav počítačové grafiky a multimédií FIT VUT, 2010 [cit. 2013-05-07]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/IKR/public/stare_prednasky_2010/04_obrazove_priznaky_2010/2010-IKR-ImageFeatures.pdf>

[2] BAY, H., ESS, A., TUYELAARS, T., VAN GOOL, L. SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding (CVIU)* [online]. ETH Zurich, 2008, 3 ,110 [cit. 2013-05-07]. Dostupné z: <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>

[3] EVANS, Christopher. Notes on the OpenSURF Library. In Notes on the OpenSURF Library [online]. [s.l.] : [s.n.], 2009 [cit. 2013-05-07]. Dostupné z: <http://opensurf1.googlecode.com/files/opensurf.pdf>

[4] Computer Vision – The Integral Image. *Computer science source* [online]. 2010 [cit. 2013-05-07]. Dostupné z: <http://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>

[5] KUČERA, Jiří. *Shluková analýza* [online]. [cit. 2013-05-08]. Dostupné z: <http://is.muni.cz/th/172767/fi_b/5739129/web/web/kmeans.html>

[6] MATTEUCCI, Matteo. *A Tutorial on Clustering Algorithms* [online]. [cit. 2013-05-08]. Dostupné z: <http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html>

[7] KINGSFORD, C., kd-Trees[online]. School of Computer Science, Carnegie Mellon University, [cit. 2013-05-07]. Dostupné z: <http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/kdtrees.pdf>

[8] K-d tree. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2 December 2006 - 2 May 2013 [cit. 2013-05-08]. Dostupné z: <https://en.wikipedia.org/wiki/K-d_tree>

[9] Homography. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2 November 2009 - 4 April 2013 [cit. 2013-05-08]. Dostupné z: <http://en.wikipedia.org/wiki/Homography>

[10] BENEDA, Martin. Homografie a epipolární geometrie. *Časopis Trilobit* [online]. 2010, roč. 2010, č. 2 [cit. 2013-05-08]. Dostupné z: <http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie#_ftn1>

[11] DAL PONT, Mattia, SILVESTRI, Andrea. 2D Homography: Algorithm and Tool. *MultiMedia Signal Processing and Understanding LAB* [online]. 2011, 26 September 2011 [cit. 2013-05-08]. Dostupné z: <http://mmlab.disi.unitn.it/wiki/index.php/2D_Homography:_Algorithm_and_Tool>

[12] YILDRZ C. An Implementation on Recognizing Panoramas [online]. Dept. of Computer Engineering Bilkent University Ankara, Turkey, 2011[cit. 2013-05-07]. Dostupné z: <http://cs.bilkent.edu.tr/~cansin/projects/cs554-vision/image-stitching/image-stitching-paper.pdf>

[13] What are different searching options in Elements Organizer?. MOHIT GUPTA, VAISHALI. *TIPS AND TRICKS FOR PHOTOSHOP ELEMENTS* [online]. 2012 [cit. 2013-05-12]. Dostupné z: <http://photoshopelementstips.blogspot.cz/>

[14] LEUTENEGGER, S., CHLI, M., Y. SIEGWART, R. BRISK: *Binary Robust Invariant Scalable Keypoints* [online]. ETH Zurich : [s.n.], 2011 [cit. 2013-05-07]. Dostupné z: <http://www.asl.ethz.ch/people/lestefan/personal/BRISK>

[15] A battle of three descriptor: SURF, FREAK and BRISK. *Computer Vision Talks* [online]. 2012 [cit. 2013-05-08]. Dostupné z: <http://computer-vision-talks.com/2012/08/a-battle-of-three-descriptors-surf-freak-and-brisk/>

[16] FU, Jing, Xiaojun JING, Songlin SUN, Yueming LU a Ying WANG. C-SURF: Colored Speeded Up Robust Features.*Communications in Computer and Information Science* [online]. 2013, 320 [cit. 2013-05-12]. Dostupné z: <http://link.springer.com/content/pdf/10.1007%2F978-3-642-35795-4_26.pdf>

[17] HEINDL, Christoph. Serialization of cv::Mat objects using Boost. *Cheind* [online]. 2011, 7 January 2012 [cit. 2013-05-08]. Dostupné z: <http://cheind.wordpress.com/2011/12/06/serialization-of-cvmat-objects-using-boost/>

[18] Zurich Building Image Database. *The Computer Vision Laboratory* [online]. c1999, *September 6, 2004* [cit. 2013-05-08]. Dostupné z: <http://www.vision.ee.ethz.ch/showroom/zubud/>

[19] Visual Object Classes Challenge 2012. *Pascal2: Pattern Analysis, Statiscical Modelling and Computational Learning* [online]. 2012 [cit. 2013-05-08]. Dostupné z: <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/index.html#testdata>

Seznam příloh

1. Obsah DVD

- zdrojové soubory
- README
- databáze s předpřipravenými slovníky
- tréninková sada fotografií
- elektronická verze technické zprávy
- elektronická verze plakátu

2. Plakát


Picture Finder
- autor: Radek Bařinka
- vedoucí: Ing. Michal Španěl, Ph. D.
- FIT VUTBR 2013
- aplikace pro vyhledávání fotografií nebo obrázků na základě jejich obsahu
- vyhledávání pomocí kosinové podobnosti doplněné o volitelnou homografii či selekci regionů
- v implementaci využito SURF, bag of words, visualního slovníku