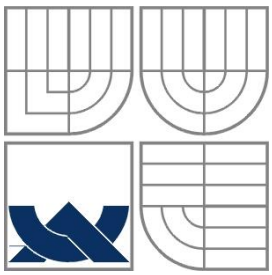
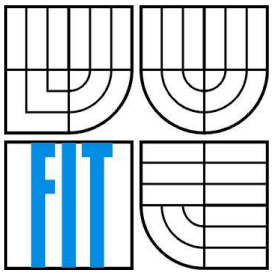


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÝ VÝBĚR REPREZENTATIVNÍCH FOTOGRAFIÍ

AUTOMATIC SELECTION OF REPRESENTATIVE PICTURES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Tomáš Bank
AUTHOR

VEDOUCÍ PRÁCE — Ing. Lukáš Polok
SUPERVISOR

BRNO 2015

Abstrakt

Tato práce je z oboru počítačového vidění. Zabývá se shlukováním fotografií podle jejich obsahu a dále výběrem dobré reprezentativní fotografie. K dosažení tohoto cíle je v práci popsáno několik metod a přístupů, ze kterých vychází návrh algoritmu. Příkladem použití výsledné aplikace může být výběr reprezentativní fotografie pro rozsáhlá alba.

Abstract

This paper belongs to field of computer vision. It deals with clustering photographs by content and selection of representative one. In this paper is described a few methods and approaches to reach the goal, and proposal of algorithm comes from those approaches. The example of usage this application can be selecting a representative photo from large albums.

Klíčová slova

počítačové vidění, výběr reprezentativní fotografie, shlukování fotografií, bag of words, algoritmus, OpenCV, slovník, obraz, fotografie, vizuální slovo, deskriptor

Keywords

computer vision, selection of representative photograph, clustering photographs, bag of words, algorithm, OpenCV, vocabulary, picture, photo, visual word, descriptor

Citace

Bank Tomáš: Automatický výběr reprezentativní fotografie, bakalářská práce, Brno, FIT VUT v Brně, 2014/2015

Automatický výběr reprezentativní fotografie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Poloka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Tomáš Bank
Datum 20.5.2015

Poděkování

Tímto děkuji vedoucímu bakalářské práce Ing. Lukáši Polokovi za užitečné rady i pomoc při zpracování této bakalářské práce.

Obsah

1 Úvod

Zadání této práce spadá do oboru počítačového vidění. Hlavní myšlenkou počítačového vidění je zjistit z obrazu co možná nejvíce informací a dále tyto informace zpracovat (rozeznávat, dávat je do souvislostí atd.) podobně jako lidský mozek. Počítačové vidění, jako obor i vše co pod něj patří, se v poslední době stává velice důležitým. Hlavním důvodem je velké využití obrazových dat a s tím související obrovské množství vznikajících obrazových dat. Tato data je třeba nějakým způsobem zpracovat, a právě v tuto chvíli přichází na řadu počítačové vidění. Je velice užitečné i důležité získat z obrazu informace o konkrétních objektech, které obsahuje. Správná kategorizace obsahu může vést například k efektivnímu vyhledávání, doporučení dalších obrazů nebo k různým jiným adekvátním reakcím. Podobnou a častou úlohou počítačového vidění může být také nalezení spojitostí mezi dvěma obrazy zachycující stejnou scénu nebo objekt. Tohoto se využívá například při kalibraci fotoaparátu/kamery. Takovéto zpracování tedy usnadňuje a zrychluje lidskou práci. Například aplikace pro identifikace objektů lze uplatnit v různých oblastech lidské činnosti a to v průmyslu, v medicíně nebo i v armádním sektoru. Toto zpracování musí zároveň zvládnout změny v obraze typické pro reálný svět, jako je nasvícení, otočení, šum apod.

V dnešní době sociálních sítí a s nimi spojeného obrovského množství fotografií začíná být nutností rychlé vyhledávání v těchto fotografiích a určení oné reprezentativní. Například výběr reprezentativní fotografie pro rozsáhlé album může člověku zabrat i desítky minut (za předpokladu, že si na tom dá opravdu záležet a chce vybrat tu nejlepší). Základní myšlenka, ze které vychází koncept tohoto výběru je, že to co se nejvíce opakuje, bývá často nejdůležitější. Stejně tak je to i při výběru reprezentativních fotografií. Nejčastěji fotografované objekty budou s větší pravděpodobností reprezentativní. Dalo by se tedy říct, že jakýmkoli výběrem (například náhodným) z největšího shluku dostaneme (s určitou mírou) dobrou reprezentativní fotografii.

V rámci této bakalářské práce byl navržen a vytvořen algoritmus pro seskupení obrazů zachycující podobný či stejný objekt. Program je zapsán v jazyce C++ a využívá open source knihovnu OpenCV. Algoritmus pracuje s modelem Bag of Words, který je známý a rozšířený v oboru počítačového vidění. Tento model využívá slovník vizuálních slov. Pro sestavení slovníku je nejprve nutné detekovat klíčové body v obraze a následně z něj extrahovat deskriptory. K tomuto je v této práci použit algoritmus SIFT, který je invariantní vůči rotaci i měřítku. Z extrahovaných deskriptorů se shlukováním vytvoří zmíněný slovník. Vizuální slova jsou středy těchto shluků. Pro seskupení se s použitím slovníku vypočítají vektory vizuálních slov pro každý z obrazů. Z těchto vektorů jsou vytvořeny shluky nejbližších vizuálních slov. Tímto je proces seskupení obrazů u konce a je na řadě výběr reprezentativní fotografie. Tento výběr funguje na principu popsaném výše a to, že nejpodobnější fotografie budou dobrými kandidáty na nejlepší reprezentativní fotografii. Obrazy z každého vytvořeného shluku se tedy porovnají, a ta která získá největší skóre v podobnosti s ostatními, se stane reprezentativní.

2 Zpracování Obrazu

Zpracováním obrazu se v této práci myslí rozpoznávání objektů, jejich shlukování do tříd a následný výběr reprezentativní fotografie z této třídy. V této kapitole budou popsány teoretické principy a metody zpracování obrazu v souvislosti se zadáním práce.

2.1 Počítačové vidění

Jak již bylo zmíněno, zpracování obrazu začíná u počítačového vidění (computer vision). Computer vision je vlastně opačný proces k vytváření obrazových dat z informací popisujících zachycené objekty. Cílem je získat požadované informace ze zachyceného obrazu, což se ukázalo jako velice obtížný úkol, na kterém posledních několik desetiletí pracovala velká řada lidí. I navzdory tomu jsme od sestavení obecně použitelného počítačového vidění stále daleko [4].

Součástí problému sestavení obecně použitelného algoritmu počítačového vidění je také komplexnost obrazových dat. Obrazová data mohou mít mnoho forem, jako například snímky videa, zdánlivě nesouvisející obrázky či multidimenzionální data z lékařských zařízení. Tato data se mohou svou složitostí výrazně lišit. Právě tato složitost je často překážkou k vytvoření efektivního a třeba i obecně platného algoritmu.

Počítačové vidění je rozsáhlý obor, který je členěn na množství podoborů. Tyto podobory zahrnují například rekonstrukci scény, detekci dějů, určování podobnosti, rozpoznávaní objektů, indexování nebo i rekonstrukci obrazu. Příkladem, který všichni dobře známe, mohou být například digitální fotoaparáty, většina z nich obsahuje algoritmy pro detekci obličeje či dokonce úsměvu. V této práci se budu zabývat rozpoznáváním objektů, určováním podobnosti obrazů a jejich následným zpracováním (seskupení, výběr reprezentativní zástupce atd.).

2.2 Podobnost

Pro rozpoznávání objektů (vzorů) a následné určování jejich podobností v obrazových datech existuje mnoho metod a modelů. K této problematice existuje kromě modelů níže uvedených spousta dalších přístupů. Tyto metody jsou aplikovatelné na jakákoliv obecná obrazová data. Některé metody (například metoda Segmentace) jsou však vhodné zejména pro určitý typ obrazových dat.

2.2.1 Registrace obrazu

Registrace obrazu je proces, při kterém hledáme vhodnou transformaci jednoho obrazu tak, aby se co nejvíce přiblížil obrazu druhému. Oba obrazy (jak registrovaný tak vztažný) obvykle zachycují stejnou scénu. Jsou ovšem pořízeny z odlišného pohledu v jiném čase nebo jiným zařízením. Registrace má za úkol právě tyto rozdíly odstranit. Tato metoda je často využívána v lékařství (CT sken nebo 3D mapování mozku a další) nebo i v armádě (rozpoznávání pro automatické zaměřování) [1].

Definice registrace

Definici registrace obrazu lze vyjádřit následovně [13]:
Registrací rozumíme nalezení geometrických transformací T_i mezi souřadnicemi jednotlivých obrazů A_i a zvoleného referenčního obrazu B. Kdy pro obrazy A_i a B při $i = 1, 2, 3, \dots, n$ platí:

$$B = f(Y) \tag{2.1}$$

$$A_i = g(Y) = f(X_i) \tag{2.2}$$

Přičemž souřadnice X_i vzniknou transformací souřadnic Y:

$$X_i = T_i(Y) \tag{2.3}$$

Metody registrace

Metody registrace lze rozdělit podle několika kritérií. Uvedu zde ty nejzákladnější [1].

a) **Dimenzionalita**

Jak samotný název napovídá, kritériem je zde počet dimenzí registrovaných dat. V nejčastějších případech jsou oba obrazy dvoudimenzionální, jedná se tedy o 2D-2D registraci (například porovnávání rentgenových snímků). Častým případem je i 3D-3D registrace. Registrace lze však provádět i mezi dimenzemi, například porovnávání tomografických dat s rentgenovými snímky je 2D-3D registrací. Jednou z dimenzí může být i čas, v takovém případě jsou porovnávány stejné snímky z různých časových okamžiků.

b) **Typ transformace**

Jedním z nejdůležitějších kroků samotné registrace je transformace. Proto je třeba vybrat řešené úloze odpovídající typ transformace. K nejčastěji používaným transformacím patří transformace tuhého tělesa, afinní transformace a elastická transformace.

- **Transformace tuhého tělesa** – používá se nejčastěji při registraci pevných objektů, které se nedeformují (například budovy). Lze ji však použít i pro registraci objektů, které mírnější deformaci podléhat mohou (např. lidské tělo). Jsou zde zachovány přímky a jejich rovnoběžnost, velikosti úhlů i poměr délek stran. Tato transformace zahrnuje posunutí, otočení a změnu velikosti.
- **Afinní transformace** – zachovává přímky a jejich rovnoběžnost. Navíc proti transformaci tuhého tělesa zahrnuje zkosení.
- **Elastická transformace** – mapuje přímky na křivky.

c) **Doménová transformace**

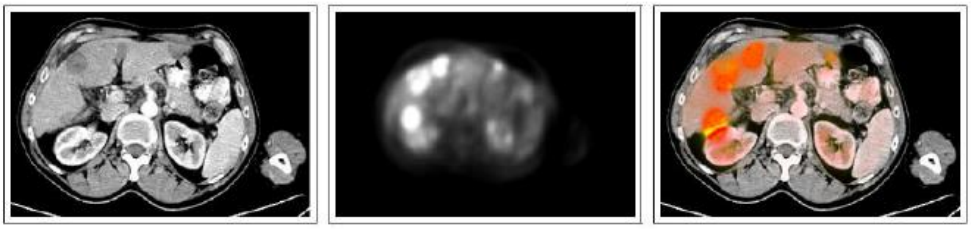
Tato transformace říká, v jaké míře je vhodné danou transformaci aplikovat. Použít však lze i různé typy transformací na různé části obrazu.

- **Globální transformace** – daný typ transformace je aplikován na obraz jako celek.
- **Lokální transformace** – transformuje se pouze vybraná část obrazu, je také složitější než transformace globální.

d) **Modalita**

Modalitu lze vysvětlit jako způsob pořízení obrazu. Registrace obrazu se podle modality dělí na:

- **Monomodální** – oba obrazy jsou pořízeny zařízením stejného typu.
- **Multimodální** – každý z obrazů je pořízen na jiném zařízení, mají tedy různou modalitu.



Obrázek 2.1. Multimodální registrace (snímek z CT, snímek z PET, výsledek). Převzato z [1].

Kriteriální funkce

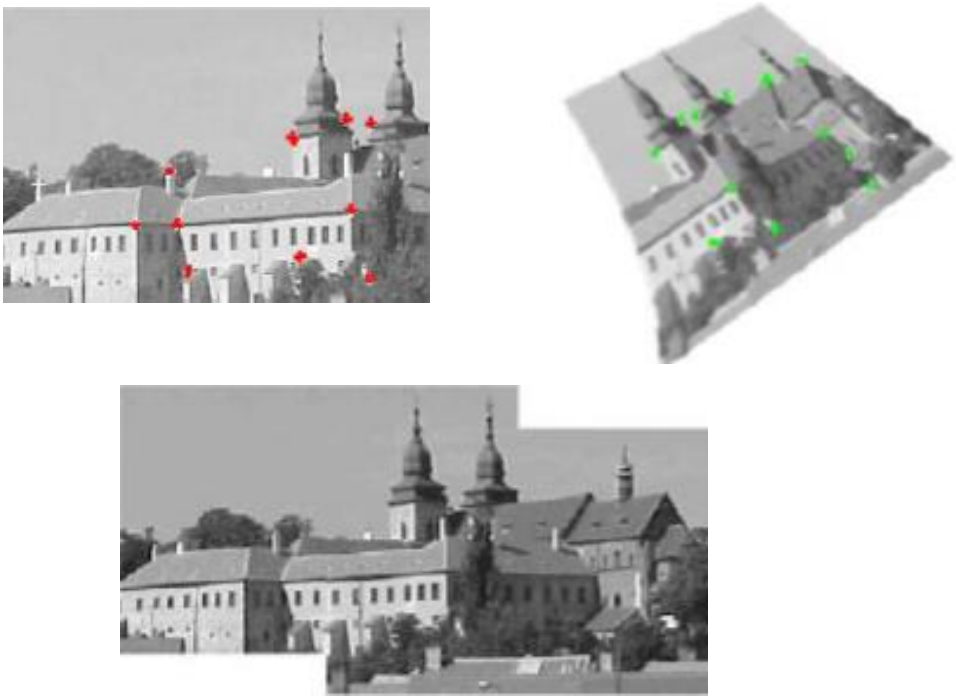
Další velice důležitou částí při registraci obrazu je míra podobnosti. Míra podobnosti slouží k výpočtu rozdílů mezi obrazy. Lze použít globální kriteriální funkce a porovnávat celé obrazy nebo lokální kriteriální funkce a porovnávat části obrazů samostatně [1]. Lze ovšem také použít na různé části obrazu různé kriteriální funkce. Další rozdělení jsou například na metody založené na intenzitě v obrazech a metody založené na informacích v obrazech.

a) **Metody založené na intenzitě v obrazech** (Intensity-based)

Vychází z předpokladu, že dvojice bodů v jednotlivých porovnávaných obrazech mají podobnou intenzitu. Nejčastěji se tedy používá při porovnávání monomodálních obrazů. Plochy jednotlivých obrazů jsou reprezentovány vektory, tím vznikne vektorový prostor, který nese intenzity jednotlivých pixelů. V podstatě jde tedy o nalezení co nejpodobnějších vektorových polí. Využívá mnoho matematických metod, například Euklidovskou vzdálenost, Korelační koeficient či Sumu kvadrátů [1].

b) **Metody založené na informacích v obrazech** (Feature-based)

Předpokládají, že konkrétnímu bodu o jisté intenzitě v obraze odpovídá bod v obraze druhém o intenzitě odlišné [1]. Základním prvkem je zde sdružený histogram, který zachycuje počet pixelů a jejich jednotlivé intenzity v porovnávaných obrazech. Lze provádět i multimodální registraci. Pracuje s algoritmy pro detekci bodů zájmu v obraze [12]. Takovými algoritmy jsou Scale-invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), které budou popsány později. Dále například Maximally Stable Extremal Regions (MSER), který se využívá při hledání spojitostí mezi obrazovými elementy dvou obrazů pořízených z různých úhlů.



Obrázek 2.2. Ukázka Feature-based registrace obrazu. Převzato z [12]

2.2.2 Segmentace

Tyto metody spočívají v rozdělení obrazové scény na samostatné části. Tedy nalezení objektů, které nás v obraze zajímají a jejich odlišení od pozadí. Segmentace rozčlení obraz do částí (segmentů), které souvisí s předměty či oblastmi reálného světa [14]. Mezi důležité metody segmentace patří:

Prahování

Metoda založená na hodnocení jasu každého pixelu. Určený práh (jasová konstanta) se využije k oddělení objektů od pozadí [3]. Stanovení úrovně prahu je pro výsledek rozhodující. Tato metoda je vhodná zejména pro obrazy, kde mají objekty dostatečně odlišnou úroveň jasu od pozadí. Díky jednoduchosti výpočtu je prahování nejrychlejší segmentační metodou a lze ji provádět i v reálném čase.

Obrázek 2.3. Prahování vhodným, nízkým a vysokým prahem. Převzato z [3].

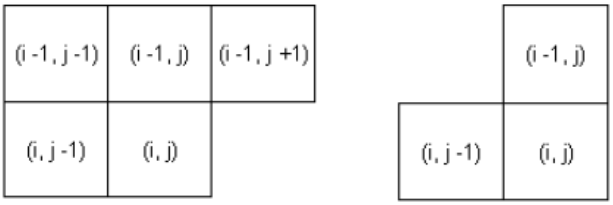
Prosté prahování

Prahování lze chápat jako transformaci vstupního obrazu g na výstupní binární obraz f s prahem T [14]. Výsledkem je binární obraz, kde obrazové elementy náležící objektu (jas větší než práh) mají hodnotu 1 a pixely náležící k pozadí hodnotu 0.

$$f(i,j) = \begin{cases} 1 \; pro \; g(i,j) \geq T \\ 0 \; pro \; g(i,j) < T \end{cases} \tag{2.4}$$

Barvení

Tato metoda prochází obraz po řádcích a každému nenulovému pixelu $f(i,j)$ přiřadí hodnotu podle hodnoty jeho sousedních pixelů (pokud sousední elementy existují). Poloha sousedních elementů je dána maskou [3].



Obrázek 2.5. Různé masky pro barvení [3].

(i – 1, j – 1)	(i, j – 1)	(i + 1, j – 1)
(i – 1, j)	(i, j)	(i + 1, j)
(i – 1, j + 1)	(i, j + 1)	(i + 1, j + 1)

Obrázek 2.4. Maska barvení pro osmiokolí [2].

Barvení probíhá [2] ve dvou průchodech obrazem, kdy v prvním průchodu jsou obarveny všechny nenulové elementy obrazu podle hodnoty jeho sousedních elementů (dány maskou). Pokud všechny sousední elementy mají nulovou hodnotu (jsou součástí pozadí), pak se přiřadí obrazovému elementu nová dosud nepřiřazená barva. Pokud je ovšem sousedních elementů nenulových více a jestliže nebyly hodnoty sousedních elementů různé (tzv. kolize barev), přiřadí se obarvovanému elementu hodnota jakéhokoliv ze sousedních nenulových elementů a zaznamená se ekvivalentní dvojice do tabulky ekvivalencí. Po tomto průchodu jsou všechny obrazové elementy oblastí obarveny, v některých oblastech ovšem mohlo dojít ke kolizím barev (oblast je obarvena více barvami). Proto se v druhém průchodu obraz projde znovu po řádcích. Podle tabulky o ekvivalenci barev se přebarví obrazové elementy dané oblasti. Poté je každá oblast obarvena jedinou, v jiné oblasti se nevyskytující barvou.

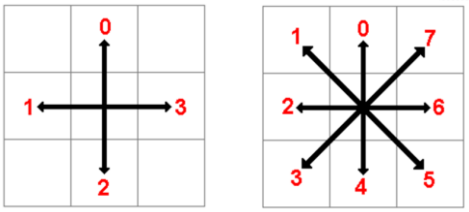
Detekce hran

Vychází z předpokladu, že hranice objektů tvoří hrany. Hledá v obraze oblasti s prudkými změnami jasu (hrany objektů) aplikací některého z hranových operátorů: Laplaceův operátor, Sobelův operátor, operátor Prewittové atd.

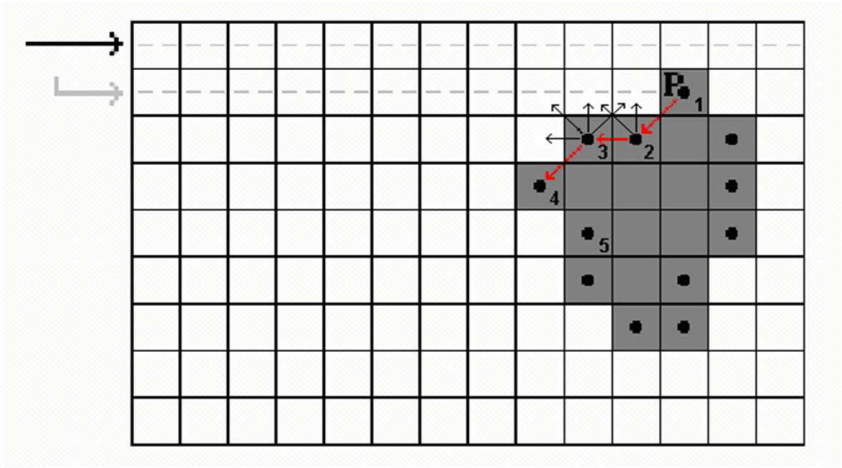
Teoreticky lze proces detekce hran rozdělit na několik částí. První z nich je filtrování, kdy vhodným nastavením filtru lze částečně odstranit šum vzniklý při vzorkování obrazu či rozmazání. Dalším krokem je pak diferenciace, která zvýrazní oblasti v obraze s významnými změnami intenzit jasu. Třetí a poslední částí je samotná detekce, při které jsou detekovány a lokalizovány body, kde je změna intenzity nejvýznamnější [15].

Sledování hranice

Postup aplikovaný na obrazy, které obsahují především informaci o hranicích (například po použití metody Detekce hran). Tvar hranice však není znám jen například barva objektu. Hledání hranice probíhá procházením obrazu po řádcích a postupným „obkroužením“ objektu [14]. Pro každý element objektu je prohledáno jeho okolí (čtyřokolí či osmiokolí). K popisu hranic objektů se používají Freemanovy řetězcové kódy, ty jsou invariantní vůči posunutí, avšak invariantní vůči rotaci nejsou [2]. U příliš zašuměných obrazů nebo také u komplikovanějších objektů tato metoda může selhávat.



Obrázek 2.7. Ukázka čtyřokolí a osmiokolí, pořadí směrů určuje další prohledávání [14].



Obrázek 2.6. Ilustrace algoritmu sledování hranice s využitím osmiokolí [14].

Zápis hranice ve Freemanově kódu pro tento obrázek bude následující: 323545607001.

2.2.3 Part-based modely

Tyto modely jsou založeny na myšlence nalezení fragmentů lokálních objektů nebo podobných částí, které se v obrazech objevují [6]. Jsou to objektové modely založené na takovýchto částech a jejich vzájemných vztazích v prostoru. Pojí se s širokou škálou detekčních algoritmů, které používají různé části obrazu samostatně s cílem zjistit, zda detekovaný objekt existuje a případně kde leží. Například při detekci obličeje se použijí jako detektory menší části, jako jsou oči, nos nebo ústa.

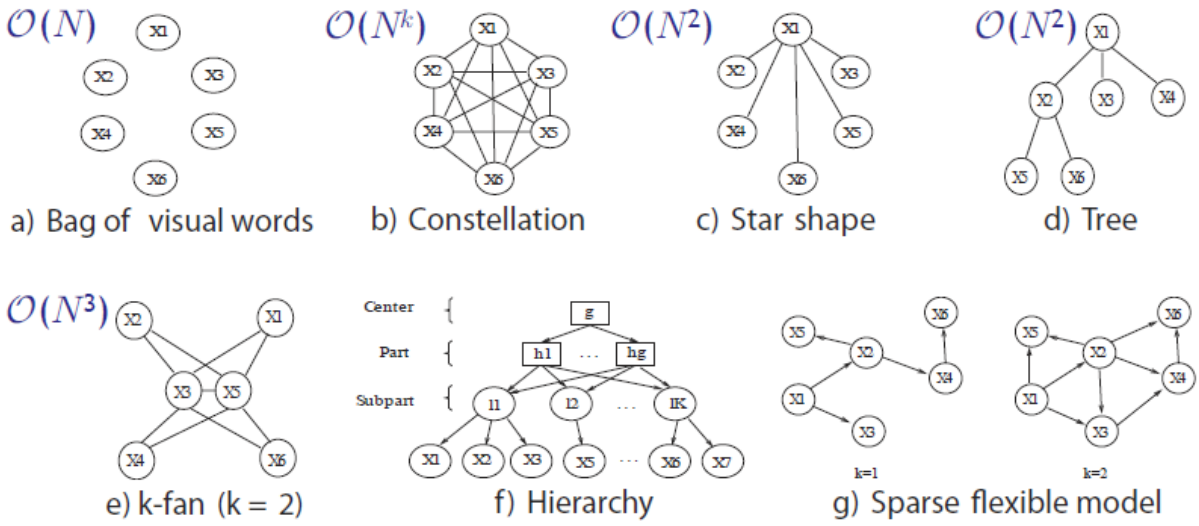
Ačkoli objekty samotné mohou výrazně napovědět, do které kategorie patří, nalezení lokálních částí a jejich vztah v prostoru často dávají důležitá vodítka ke správnému začlenění do kategorie. Toto je demonstrováno na obrázku 2.8, kde jsou pro rozpoznání objektu vozidla použity lokální části (světla, pneumatiky atd.).

Obrázek 2.8. Ukázka nalezení lokálních fragmentů [6].

Učící algoritmus by měl být sám o sobě schopen vybrat, které lokální objekty (části) co reprezentují a také je podle podobnosti seskupit do často reprezentovaných skupin. Řešením problému výběru by mohlo být prohledávání obrovského vyhledávacího prostoru. Avšak detekce invariantních lokálních vzorků poskytuje efektivní alternativu, která se v praxi osvědčila.

Jakmile jsou části definovány, nastává otázka, jak reprezentovat jejich vzájemný vztah v prostoru. Toto rozhodnutí závisí na vzájemně nezávislých předpokladech, které řešíme ohledně relativního umístění částí, a přímo ovlivňuje počet parametrů potřebných pro úplné specifikování výsledného modelu.

Nejjednodušším modelem je Bag of Words model. Tento model jako jediný neuchovává žádnou informaci o geometrických vztazích mezi vizuálními slovy (lokálními částmi). Opačným extrémem je pak úplně propojený model známý jako Constellation Model, který je vyjádřen propojením všech párů. Kompromisem dvou výše popsaných extrémů může být Star Model, kde jsou všechny části spojené pouze k centrální části, a na umístění všech ostatních je nezávislý. Výhodou tohoto modelu je efektivita výpočtu, jehož složitost je $O(N^2)$. Myšlenka Star Modelu může být snadno zobecněna do Tree Modelu, kde každé umístění závisí pouze na umístění rodiče. Za překlenutí rozdílu mezi úplně propojeným Constellation Modelem a centrálně spojeným Star Modelem, lze považovat k-fan model. Ten sestává z úplně propojené skupiny k referenčních částí, ke které se pojí druhá skupina, která je propojená pouze s touto referenční skupinou. Složitost k-fan modelu je $O(N^{k+1})$, na obrázku níže pro $k = 2$. Na podobném principu je založen i Hierarchický Model, který obsahuje vrstvu, kde jsou části objektů. Každá z těchto částí je spojena s lokálními vzorky vespod (obrázek 10. (f)). Posledním zmíněným modelem je Sparse Flexible Model, kde umístění každé lokální části závisí na jejím umístění ke k nejbližším sousedům, což umožňuje flexibilní uspořádání a deformovatelné objekty [6].



Obrázek 2.9. Přehled výše popsaných part-based modelů s jejich složitostmi [6].

Bag of Words model

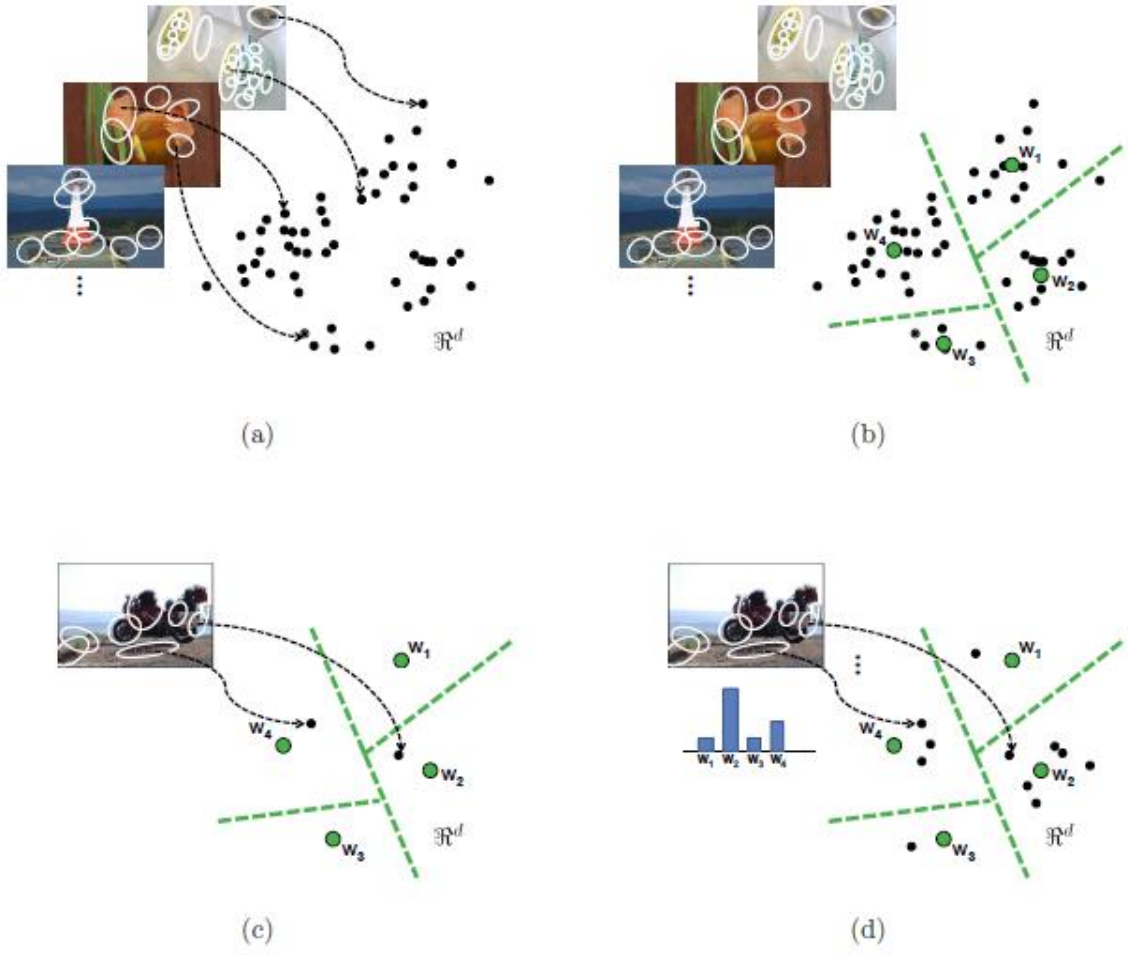
Jak již samotný název napovídá, princip této metody je převzat z analýzy textu. Při té jde o vytvoření dokumentu, který obsahuje neseřazená klíčová slova. V oboru počítačového vidění je to podobné. Každý objekt je reprezentován jako skupina („pytel“ tedy „bag“) vizuálních slov. Jinak řečeno, každý obraz je určen frekvencí těchto slov. Tato frekvence vizuálních slov je zachycena do histogramu daného obrazu [5].

Bag of words definice

Formálněji lze Bag of Words model definovat takto [7]:
Uvažujme tréninkovou skupinu *D* (dataset) obsahující *n* fotografií. Potom tedy: $D = d_1, d_2, \dots, d_n$, kde *d* jsou extrahované vizuální vzorky. Na základě těchto vizuálních vzorků vytvoří algoritmus (například *K*-means) množinu vizuálních slov *W* o předem dané velikosti, reprezentovanou jako $W = w_1, w_2, \dots w_v$, kde *v* je číslo shluku. Následně lze data sumarizovat do tabulky $V\ x\ N$ počtů výskytů, kde v jednotlivých položkách tabulky je uloženo $N_{ij} = n(w_i, d_j)$, kde $n(w_i, d_j)$ značí jak často se vizuální slovo w_i vyskytuje v obraze d_j.

Bag of words slovník

Prvním krokem pro vytvoření modelu je sestavení slovníku [5][6]. Toho docílíme extrakcí rysů (vzorků) z většího množství tréninkových obrazů konkrétního objektu. Po tomto je každý obraz zastoupen několika částmi. Tyto části jsou reprezentovány ve formě vektorů, kterým se říká tzv. deskriptory. Deskriptor by měl být schopen nést informaci o intenzitě, natočení, měřítku atd. Pro detekci a extrakci rysů z obrazů existují různé algoritmy. Jeden z nejznámějších je Scale-invariant feature transform (SIFT).



Obrázek 2.10. Schéma popisující vytvoření a použití slovníku Bag of Words. Převzato z [6].

Obrázek 2.10 popisuje způsob vytvoření a použití slovníku Bag of Words ve čtyřech krocích [6]. V prvním kroku a) je použito velké množství tréninkových obrazů. Bílé elipsy na obrázku značí lokální regiony vzorků a černé tečky značí body v některém z regionů (určené například algoritmem SIFT). b) Dále jsou získané vzorky shlukovány tak, aby pokryly prostor diskrétního (a předem známého) počtu vizuálních slov (například algoritmem *K*-means). Vizuální slova jsou středy těchto shluků, na obrázku jsou znázorněny jako zelené tečky. Zelené čárkované čáry vyznačují Voroného diagram (konkrétně Voroného buňky) na základě vybraných vizuálních slov. c) Při použití na nový obrázek je pro každý jeho vzorek identifikováno nejbližší vizuální slovo. Tímto se přemapuje obrázek ze sady vícerozměrných deskriptorů na list čísel vizuálních slov. d) Ke shrnutí vizuálních slov lze použít histogram. V něm je zachyceno, kolikrát se každé z vizuálních slov vyskytuje v obraze.

Constellation Model

Reprezentuje objekty odhadováním výskytu a tvaru jejich částí. Tudíž části objektů lze charakterizovat rozdílností výskytů nebo rozdílností umístění v objektu. Tento model je velice flexibilní a lze ho aplikovat i na objekty, které jsou popsány pouze svou texturou.

Naučený objekt je reprezentován P částmi a parametry θ, úlohou je potom rozhodnout zda nový testovaný obraz obsahuje instance naučených objektů. Proto se extrahuje N lokálních vzorků s X umístěními, S měřítky a výskyty A. Constellation Model vyhledá přiřazení h vzorků částím podle Bayesianova [6] rozhodnutí R.

$$R = \frac{p(Object|X,S,A)}{p(No\ object|X,S,A)} \approx \frac{p(X,S,A|\theta)p(Object)}{p\left(X,S,A\middle|\theta_{bg}\right)p(No\ object)} \tag{2.5}$$

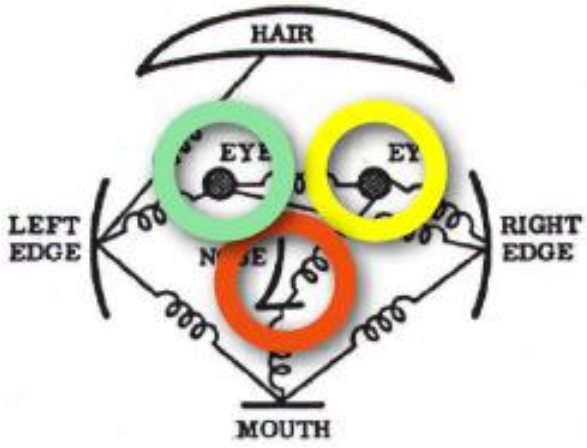
Potom lze pravděpodobnost určit následovně:

$$p(X,S,A|\theta) = \sum_{h\,\epsilon\,\mathrm{H}} p(X,S,A,h|\theta) = \sum_{h\,\epsilon\,\mathrm{H}} p(A|X,S,h,\theta)p(X|S,h,\theta)p(S|h,\theta)p(h|\theta) \tag{2.6}$$

Takto reprezentujeme pravděpodobnost jako výsledek oddělených podmínek výskytu (A), tvaru (X), relativního měřítka (S) a ostatních vlivů.

Míra klasifikace je vypočítána marginalizací přes všechna možná přiřazení vzorků částím $|\mathcal{H}| \subseteq O(N^P)$. Díky tomuto je možné reprezentovat celou kategorii relativně malým počtem částí. Exponenciální složitost tohoto výpočtu představuje hlavní omezení, neboť limituje tento přístup na relativně malý počet částí.

Constallation Model byl navrhnut s cílem učení se slabým dohledem. Což znamená, že stačí znát pouze labely obrázků (cílová kategorie, či pozadí). Cílem učící metody je nalézt maximální odhad pravděpodobnosti pro parametry modelu.

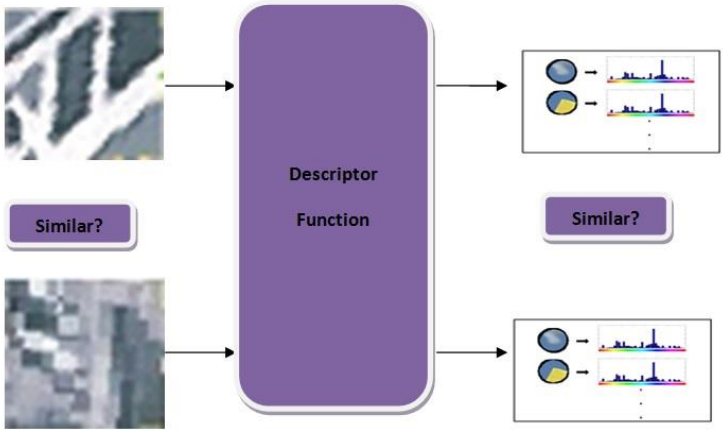


Obrázek 2.11. Reprezentace obličeje a nalezení jeho částí.

2.2.4 Deskriptory

V počítačovém vidění znamená pojem deskriptor popis vizuálních vzorků obsahu obrazu [9]. Jsou nedílnou součástí algoritmů, které využívají tzv. interest point detection tedy se zabývají detekováním bodů zájmu v obraze. Popisují základní vlastnosti jako tvar, barvu, texturu apod. Deskriptory jsou prvním krokem k nalezení spojení mezi pixely v digitálním obraze a lidským pozorováním obrázku (či skupiny obrázků).

Například při určování podobnosti dvou obrazů. Můžeme vzít část z obou porovnávaných obrazů a tuto část porovnávat pixel po pixelu měřením Euklidovské vzdálenosti. Takové porovnávání by ovšem bylo velice citlivé na změnu šumu, otočení či změnu jasu. Právě k tomuto se hodí deskriptory. Deskriptory lze onu část obrazu popsat způsobem, který je nezávislý na takovýchto změnách v obraze (šum, otočení atd.) [8].



Obrázek 2.12. Použití deskriptorů pro porovnání dvou částí obrazů [8].

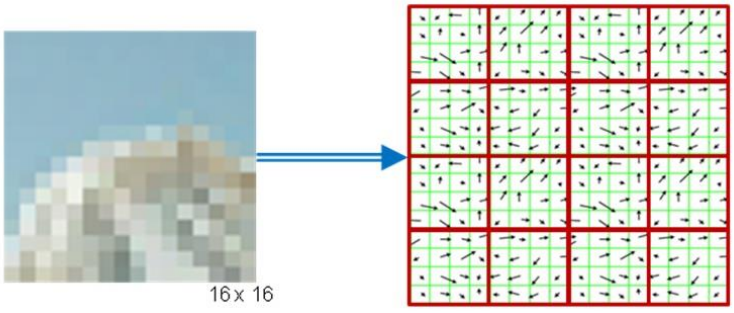
HOG deskriptory (Histogram of Oriented Gradients)

Tato technika je založená na počítání výskytu orientovaných gradientů [8]. Myšlenkou HOG deskriptorů je, že objekt a jeho tvar v obraze lze popsat rozložením intenzity gradientů nebo směrů hran. Nejznámější algoritmy spadající do této rodiny jsou *SIFT*, *SURF* a *GLOH*.

a) **SIFT (Scale-invariant feature transform)**

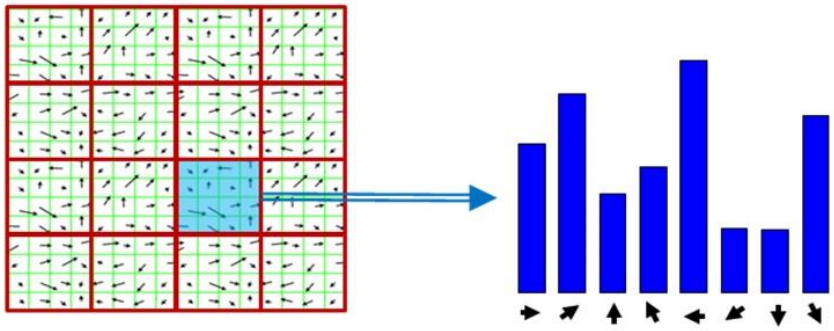
Je algoritmus k detekování a popisu lokálních vzorků (příznaků) v obraze. Představil ho v roce 1999 David Lowe a zahrnuje, jak detekování klíčových bodů, tak i deskriptory [10]. Je invariantní vůči rotaci, měřítku a částečně i změnám jasu.

Prvním krokem tohoto algoritmu je tedy detekce klíčových částí obrazu. Následně vypočítá gradienty pro každý pixel z klíčové části a rozdělí tuto oblast do menších čtverců.



Obrázek 2.13. SIFT výpočet gradientů a rozdělení klíčové oblasti. Převzato z [8].

Pro každý ze čtverců vypočítá histogram směrů gradientů (do 8 směrů).



Obrázek 2.14. SIFT – výpočet histogramu. Převzato z [8].

Výsledné histogramy konkatenuje, z čehož vnikne v tomto případě 128 (16 čtverců * 8 směrů) rozměrný vektor vzorků.

Obrázek 2.15. SIFT – konkatenace histogramů gradientů. Převzato z [8].

Takže pro porovnání dvou obrazů stačí vypočítat jejich deskriptory a určit podobnost změřením podobnosti deskriptorů, což znamená vypočítat vzdálenost mezi deskriptory.

b) **SURF (Speeded Up Robust Features)**

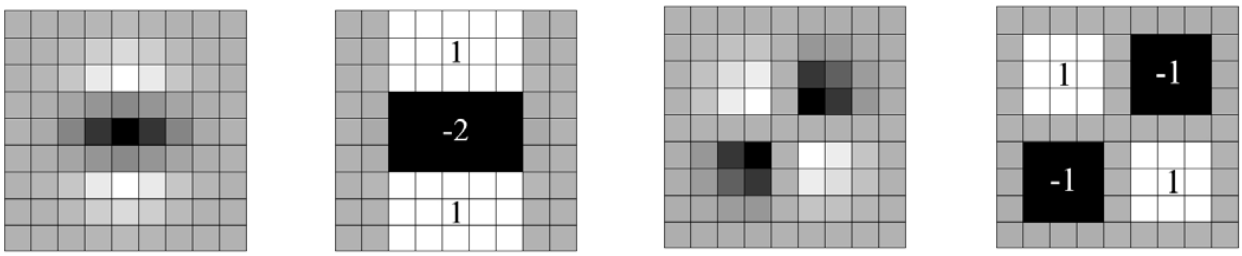
Jedná se o novější obdobu metody SIFT. Poprvé ji představil Herbert Bay na mezinárodní konferenci o Počítačovém vidění v Austrálii v roce 2006. Při navrhování tohoto algoritmu byl kladen důraz na výrazné snížení výpočetní náročnosti při zachování výkonu. Jeho výhodou je tedy hlavně rychlost, která by měla být dostačující pro využití v real-time aplikacích.

Algoritmus SURF je založen na teorii víceúrovňového prostoru. Detekce vzorků (příznaků) je zde založena na výpočtu determinantu Hessovy matice, a to díky jejímu výkonu a přesnosti. V obrázku I bereme bod $\mathrm{x} = (\mathrm{x}, \mathrm{y})$, Hessovu matici lze pak definovat následovně [16]

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{2.7}$$

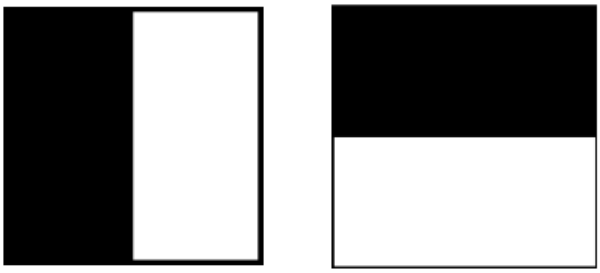
kde $H(x,\sigma)$ je Hessova matice v bodě x a měřítku σ, $L_{xx}(x,\sigma)$ je pak konvoluce druhé derivace Gaussovy funkce s originálním obrazem I v bodě x. Stejně tak i $L_{xy}(x,\sigma)$ a $L_{yy}(x,\sigma)$.

Deskriptory algoritmu SURF jsou založeny na podobném principu jako u SIFT (Scale Invariant Feature Transform), které jsou velmi výkonné. Prvním krokem je rozhodnutí o orientaci založené na informacích z kruhové oblasti kolem bodu zájmu, ve které jsou zaznamenávány odezvy na Haarovy vlnky. Druhým krokem je vytvoření čtvercové oblasti natočené ve vybraném směru, z této oblasti je pak extrahován SURF deskriptor.



Obrázek 2.16. Druhá derivace Gaussovy funkce ve směru y, vedle ní je její aproximace. Následuje druhá derivace Gaussovy funkce ve směru a xy, vedle opět její aproximace. Šedé oblasti jsou rovny nule [16].

Kvůli invarianci vůči rotaci i měřítku, je nutno každému bodu přiřadit orientaci, která je vypočítána z jeho okolí. Pro výpočet se používají Haarovy vlnky jako konvoluční masky ve směru os x a y [16].



Obrázek 2.17. Haarovy vlnky ve směru x a ve směru y. [16]

2.3 Klasifikace

Klasifikací je v této práci myšleno shlukování obrazů podle objektů. Tohoto lze dosáhnout například vytvořením slovníku vizuálních slov (Bag of Words) popsaného v předchozích kapitolách. A následným vypočítáním vektoru vizuálních slov konkrétní fotografie právě vůči tomuto slovníku. Shlukováním těchto vektorů (například algoritmem *k*-means) lze dosáhnout seskupení stejných objektů. Jelikož metoda shlukování (clustering) patří do tzv. unsupervised learning, tedy do úloh učení bez učitele. Učení bez učitele se používá, pokud není k dispozici informace od učitele, tzn. trénovací množina. Lze říci, že tyto metody pracují s myšlenkou, že data patřící jedné třídě jsou si navzájem podobnější než data z různých tříd.

2.3.1 K-Means

Algoritmus *k*-means je nejznámějším a patří k nejpoužívanějším algoritmům shlukování, kdy je shlukováno do *k* shluků a patří do metod nehierarchického shlukování. *K*-means ukládá *k* centroidů, které definují shluk. Bod je považován za součást shluku, pokud jeho vzdálenost ke středu tohoto shluku je menší, než ke kterémukoli jinému středu.

Středy se určují postupně opakováním jednoduchých kroků. Body se přiřadí do shluků na základě aktuálního středu. Následně vypočítá nové centroidy, na základě těchto přiřazených bodů. Algoritmus končí, pokud se středy shluků přestanou měnit.

Formálnější definice tohoto algoritmu říká: Mějme vstupní množinu dat $X = \{x_1, \dots, x_n\}$ a číslo k udávající počet vektorů μ_j. Vektory μ_j, kde $j = 1, \dots, k$, se na začátku inicializují na náhodně zvolenou hodnotu nebo použitím vhodně zvolené heuristiky (například využívající apriorní znalosti o úloze). Poté jsou iterativně opakovány následující dva kroky:

1. **Klasifikace:**
   Veškerá data z X, tedy x_i, kde $i = 1, \dots, n$ se klasifikují do tříd určených vektory μ_j, kde $j = 1, \dots, k$ podle minima euklidovské vzdálenosti. Přiřazení do tříd lze zapsat následujícím vztahem:

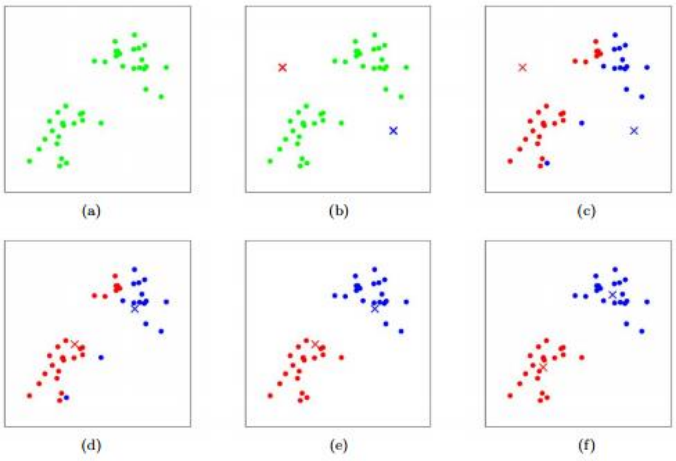
$$y_i = argmin \left\| x_i - \mu_j \right\| \qquad \text{pro } j = 1, \dots, k \tag{2.8}$$

2. **Učení:**
   Vypočítají se nové hodnoty vektorů μ_j jako střední hodnoty dat x_i, které byly klasifikovány do tříd určené příslušným vektorem. Nová hodnota vektoru μ_j je určena vztahem:

$$\mu_j = \frac{1}{n_j} \sum_{i \, \epsilon \, \{i : y_i = j\}} x_i, \tag{2.9}$$

   kde n_j je počet vzorů x_i klasifikovaných v kroku číslo jedna do třídy určené vektorem μ_j.

Tyto dva kroky se opakují do té doby, dokud se alespoň jeden vektor x_i klasifikuje do jiné třídy, než byl klasifikován v předcházejícím kroku. To znamená, že algoritmus skončí, jakmile se vektory x_i ustálí a nemění se [17][18].



Obrázek 2.18. Ukázka K-means algoritmu [18].

Obrázek znázorňuje průběh algoritmu *k*-means, v části a) je vidět vstupní množina bodů. Prvním krokem je inicializace středů na náhodně zvolené hodnoty, znázorněno v b). Na c) až f) jsou vidět dvě iterace algoritmu. Nejdříve jsou přiřazeny body ze vstupní množiny do nejbližšího shluku (znázorněno červenou a modrou barvou). Poté jsou středy shluků posunuty do středu nově přiřazených shluků [18].

2.4 Výběr

V této kapitole budou popsány postupy a metody pro určení nejlepší reprezentativní fotografie. Výběr té nejlepší reprezentativní fotografie je obtížná práce. Jelikož výběr té zdánlivě nejlepší fotografie může být velice subjektivní, tedy každý člověk to může vidět jinak než ostatní. Počítačové vidění ovšem touto subjektivitou netrpí. Stačí určit aspekty, podle kterých má reprezentativní fotografii vybrat.

Jak již bylo řečeno v úvodu, nejčastěji fotografované objekty budou s větší pravděpodobností na reprezentativní fotografii. Tudíž reprezentativní fotografie bude velice podobná ostatním fotografiím ve shluku a zároveň velice rozdílná od náhodných fotografií mimo shluk.

2.4.1 Sub-Clustering

Reprezentační fotografie budou vybírány z již připraveného shluku fotografií zachycujících stejný objekt. To však ještě není dostatečně přesný shluk pro výběr té nejsprávnější fotografie. Proto je třeba detekovat nejvíce podobné fotografie, analyzovat je a až poté vybrat tu nejvhodnější. Před určením nejpodobnějších fotografií a před samotným výběrem lze provést tzv. Sub-Clustering tedy vytvoření podskupin (shluků) [11].

Problémem při hledání nejpodobnějších fotografií by mohlo být obrovské množství dvojic obrazů, které je nutné prozkoumat. Pokud máme například *N* fotografií, je nutné prověřit $\binom{N}{2}$ různých dvojic. Pro redukci složitosti vytvoříme již zmíněné podskupiny. Následně v každé z těchto podskupin určíme nejvíce podobné obrazy, tzn. fotografie z různých podskupin nebudou porovnávány.

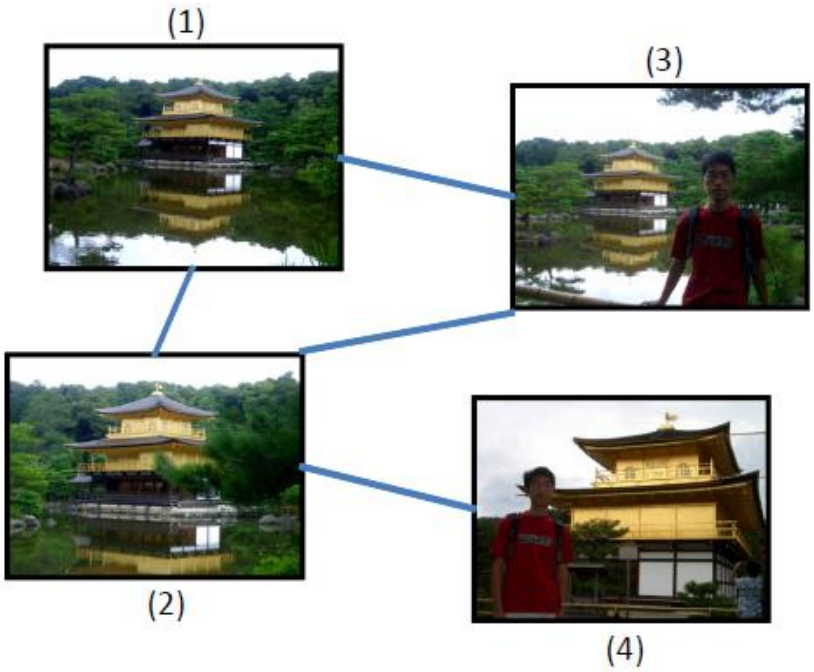
Máme-li N fotografií, které jsou rozděleny do M podskupin $\{C_1, C_2, \dots, C_M\}$, a například $N = 10, M = 2, |C_1| = 4$ a $|C_2| = 6$. Musíme prověřit $\binom{4}{2} + \binom{6}{2} = 21$ dvojic fotografií. Kdežto bez použití tohoto přístupu by bylo nutné prověřit $\binom{10}{2} = 45$ dvojic fotografií, což je více než dvojnásobek.

Realizování procesu seskupení do podskupin lze implementovat například na základě barevných (RGB) histogramů.

2.4.2 Výběr reprezentativní fotografie

Nyní předpokládáme, že v podskupině *C* jsou jen nejpodobnější fotografie. Může tedy začít samotný výběr reprezentativní fotografie.

Vztah mezi fotografiemi v *C* lze reprezentovat jako neorientovaný a nevážený graf $G = (V, E)$, kde $V = \{v_1, v_2, \dots, v_n\}$ je množina, ve které každý vrchol (fotografie) v_i je nejméně jednou považován za nejbližší (nejpodobnější) jinému. Hrana e_{ij}je v E, pokud v_ia $v_j$jsou považovány za téměř shodné [11]. Po sestavení tohoto grafu lze určit nejdůležitější vrcholy. Pro zjištění nejdůležitějších vrcholů musíme porovnat jejich centralitu. K tomuto existuje několik způsobů. Mezi hlavní patří centralita měřená stupněm uzlu, blízkostí polohy ve středu nebo centralita měřená středovou mezipolohou.



Obrázek 2.19. Znázornění vztahů mezi nejvíce shodnými fotografiemi grafem. Převzato z [11].

Jako reprezentativní bude tedy vybrána fotografie, která má v grafu největší hodnotu centrality. V případě obrázku výše je vybrána fotografie číslo 2. Tato fotografie je určená jako nejpodobnější vůči ostatním obrazům ve skupině, při znázornění grafem vztahů má tedy největší hodnotu centrality.

3 Návrh

Zadáním práce je navrhnout algoritmus, který z množiny fotografií vytvoří skupiny zachycující ten samý objekt. Následně ze skupiny vybere dobrou reprezentativní fotografii. Obsahem této práce je tedy vyřešit dva základní problémy, které zadání popisuje. Nejdříve je nutné vytvořit skupiny fotografií se stejnými objekty a následně z těchto fotografií vybrat tu správnou reprezentativní. Toto bude popsáno v následující kapitole.

3.1 Dekompozice problému

Prvním úkolem algoritmu je tedy projít zadanou množinu fotografií a na základě podobnosti vytvořit skupiny fotografií zachycující stejný objekt. Pro toto bude použit model počítačového vidění Bag of Words. Tento model funguje na principu slovníku. Projde se tedy zadaná množina fotografií a algoritmem SIFT se extrahují vizuální vzorky (příznaky), ze kterých algoritmus *k*-means vytvoří shluky. Středy těchto shluků jsou potom slovníkem vizuálních slov. Vytvoření a použití slovníku je popsáno v kapitole 2.2.3 a znázorněno na obrázku 2.10. Jak již bylo zmíněno, pro extrakci vizuálních vzorků bude použit algoritmus SIFT, jeho podrobný popis je uveden v kapitole 2.2.4, proto ho zde nebudu znovu uvádět.

Shlukování fotografií proběhne vypočítáním vektoru slov pro každou z fotografií vůči vytvořenému slovníku. Z těchto vektorů vizuálních slov algoritmus *k*-means vyrobí shluky. Výsledkem je množina shluků fotografií.

Hlavním důvodem pro použití modelu Bag of Words je jeho jednoduchost a pochopitelnost. Myšlenka, se kterou tento model pracuje, mi přijde snadno představitelná, vysvětlitelná i relativně dobře implementovatelná. Algoritmus SIFT jsem zvolil kvůli tomu, že se jedná o jeden z nejznámějších algoritmů i pro jeho kvalitu. Jeho implementace je také zahrnuta ve volně použitelné knihovně OpenCV, což je veliká výhoda.

V tomto okamžiku algoritmus vytvořil shluky fotografií zachycující stejný objekt. Dalším krokem je tedy výběr reprezentativní fotografie z jednoho ze shluků. Určení reprezentativní fotografie proběhne na základě porovnání vektorů vizuálních slov obrazů. Vektory vizuálních slov se určí pro všechny fotografie a provede se porovnávání dvojic fotografií, z čehož se sestaví graf vztahů mezi obrazy.

Pro každou z podskupin poté začne sestavování grafu vztahů. Pro určení nejdůležitějšího vrcholu v grafu a tedy i nejlepší reprezentativní fotografie se zjistí centralita jednotlivých vrcholů. Podle mého názoru se na toto hodí centralita měřená stupněm uzlu. Kdy stupeň uzlu (vrcholu) udává počet přímých vazeb k dalším uzlům. Čím větší stupeň tím větší počet přímých vazeb na další vrcholy. To znamená, že fotografie, která má nejvíce vztahů (je podobná nejvíce ostatním fotografiím) bude považována za reprezentativní. Například na obrázku 2.19 bude jako reprezentativní vybrána fotografie č. 2. Po určení centrality ve všech podskupinách budou prohlášeny za reprezentativní ty fotografie, které dosáhnou největší hodnoty centrality. Tímto bude mít každá z podskupin vybranou reprezentativní fotografii.

4 Implementace

Implementace začíná u open source knihovny OpenCV. Ta je použitá ve verzi 2.4.10, která byla v začátku implementace nejaktuálnější. Jelikož se jedná o práci s větším množstvím dat a náročností výpočtů, je k zajištění rychlosti a efektivnosti algoritmu použita knihovna OpenMP, která je součástí OpenCV a zařídí paralelizaci kritických částí. Práce je logicky členěna do dvou hlavních souborů, tak aby byly na sobě nezávislé. Pro seskupení fotografií to jsou *createVocabulary.cpp, clusterPhotos.cpp*. Ve druhém z nich je i výběr reprezentativní fotografie.

4.1 Vytvoření slovníku

Celý proces začíná vytvořením slovníku ze zadané množiny fotografií. Slovník vytvoří algoritmus popsaný v *createVocabulary.cpp*, který má jediný parametr a to cestu k adresáři ukončenou lomítkem.

Program začne procházet adresář zadaný parametrem. Zde ho zajímají pouze soubory, vše ostatní ignoruje (adresáře, skryté soubory atd). Informace o názvu jednotlivých souborů ukládá do vektoru řetězců (název souboru je uložen i s cestou).

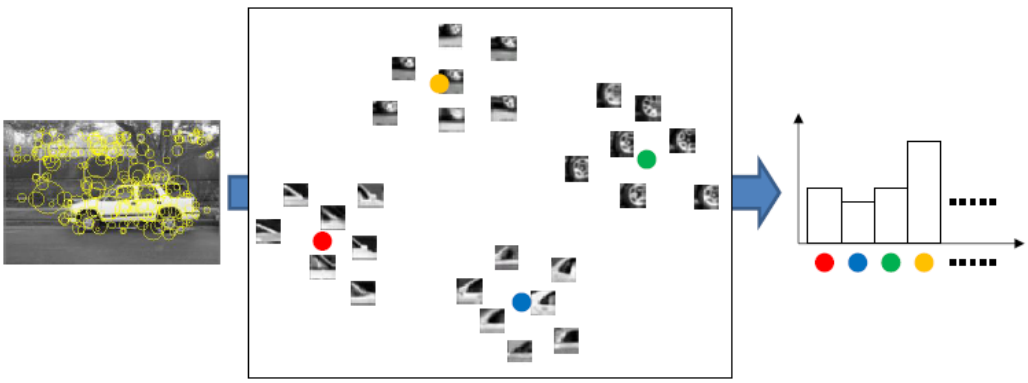
Jakmile jsou načteny všechny soubory, přistoupí se k detekci klíčových bodů a extrakci deskriptorů. Vytvořený vektor, který obsahuje názvy souborů, se začne procházet. Ačkoli otevřít obrázek a detekovat by jistě šlo již v průchodu adresářem, v důsledku zvýšení rychlosti se detekce a extrakce deskriptorů provádí až v cyklu procházející vektor. V tento okamžik je již známý počet obrazů ke zpracování, což umožňuje použití cyklu `for()`, který lze pomocí knihovny OpenMP snadno paralelizovat.

V paralelizovaném `for()` cyklu se nejdříve přečte název obrazu (je i s cestou) a hned poté se obraz načte do matice. Ještě před detekováním proběhne kontrola, zda se načetla nějaká data (zda je obraz v matici načtený). Pokud se něco špatně načetlo, obraz se přeskočí a pokračuje se dalším. K detekování klíčových bodů je použita třída `SiftFeatureDetector()`. Jak už název třídy napovídá, k detekci využívá algoritmu SIFT. Pro samotné detekování je z této třídy (přesněji z rodičovské třídy `FeatureDetector()`) zavolána metoda `detect()`, která vrátí vektor detekovaných klíčových bodů z obrazu. Nyní je nutné extrahovat deskriptory, k čemuž je využita třída `SiftDescriptorExtractor()`. Metodou `compute()`, kterou třída dědí z nadřazené třídy `DescriptorExtractor()`, je pro klíčové body, v předchozím kroku detekované, extrahovaná matice deskriptorů. Ta je uložena k ostatním a pokračuje se s dalším obrázkem.

Obrázek 4.1. Ukázka detekovaných klíčových bodů algoritmy SIFT a SURF.

Po zpracování všech obrazů, se do souboru *usedFiles.txt* uloží vektor obsahující informace o názvu souborů. Pokud se extrahovaly nějaké deskriptory, pak se i ty uloží, a to do souboru *trainedDescriptors.yml*. Po uložení se vloží do třídy `BOWKMeansTrainer()`, která vytváří slovník. Zavoláním metody `cluster()` z této třídy, dojde ke shlukování a vytvoření slovníku vizuálních slov. Ten je poté také uložen a to do *vocabulary.yml*.

Pokud soubor s informacemi o již přečtených souborech existuje při spuštění tohoto programu a existují i soubory ukládající deskriptory a slovník, pak se všechny načtou. Obrazy, jejichž název se načetl ze souboru použitých souborů, se znovu neprocházejí. Pokud se nepodařilo extrahovat žádné deskriptory, což znamená, že slovník již všechny z obrazů obsahuje nebo prostě žádné obrazy nejsou, pak program končí. Extrahované deskriptory se přidají k těm načteným ze souboru a extrahovalo-li se jich více než *K*, kde *K* je počet shluků vytvořených při shlukování slovníku algoritmem *k*-means. Potom se z těchto nových deskriptorů vytvoří nový slovník, který se následně přidá do již uloženého slovníku. Díky tomuto není potřeba trénovat celý slovník pořád dokola.



Obrázek 4.2. Vizualizace vytvoření Bag of Words slovníku. Detekce klíčových bodů, shlukování a slovník vizuálních slov [19].

4.2 Shlukování

Po vytvoření slovníku je dalším krokem shlukování fotografií. K tomu je zapotřebí slovník vytvořený výše popsaným programem. Proces shlukování popisuje *clusterPhotos.cpp*. Jediným parametrem tohoto programu je již zmíněný slovník.

Po spuštění se tedy jako první načte slovník uvedený parametrem, také se ze souboru *usedFiles.txt* načtou názvy obrazů. Vytvoří se instance třídy `BOWImgDescriptorExtractor()`, do které se následně vloží slovník, v konstruktoru se třídě nastaví extraktor a matcher. Extraktorem je už dříve použitá třída `SiftDescriptorExtraktor()` a třída `DescriptorMatcher()` vytvoří matcher typu FlannBased. Také je nutné vytvořit detektor klíčových bodů, k čemuž je použita třída `SiftFeatureDetector()`.

Opět v paralelizovaném cyklu `for()` se začnou procházet názvy obrazů. Obraz je načten do matice, ze které se vytvořeným detektorem detekují klíčové body. Poté jsou metodou `compute()`, třídy `BOWImgDescriptorExtraktor()` vypočítány deskriptory z Bag of Words slovníku. Tato metoda nejprve spočítá deskriptory obrázku a detekovaných klíčových bodů, následně pro každý z deskriptorů nalezne nejbližší vizuální slova z BoW slovníku. Nakonec podle těchto nejbližších vizuálních slov určí BoW deskriptory. Ty se uloží do matice, kde každý jeden řádek odpovídá jednomu obrázku (matice bude mít stejný počet řádků jako je obrazů).

Nyní je potřeba vytvořit z těchto dat shluky. K vytváření shluků se použije algoritmus *k*-means. Tento algoritmus je součástí OpenCV, takže stačí zavolat metodu `kmeans()` nad právě sestavenou maticí BoW deskriptorů. Tato metoda vrátí matici labelů, na jejíž každém řádku je číslo shluku, do kterého daný řádek (obrázek) patří. Také vrátí matici středů vytvořených shluků, kde každý řádek odpovídá právě jednomu shluku.

Problémem ovšem je, jak správně určit vhodný počet shluků, tak aby obrazy nebyly příliš pomíchané. Jedním ze způsobů může být vypočítat rozptyl shluků a jejich vzdálenosti. Pro každý z bodů ve shluku je tedy vypočítána Euklidovská vzdálenost od středu shluku, který vrátil *k*-means. Z tohoto se určí průměrná vzdálenost ve shluku a následně i průměrná vzdálenost ve všech shlucích. Pokud je ta větší než experimentálně určená přesnost, pak jsou shluky příliš rozptýlené a je potřeba snížit jejich počet. Celý proces se tedy zopakuje s počtem tříd zvětšeným o jedničku.

Jakmile se vytvoří optimální počet shluků, projdou se názvy fotografií a načtou se do pole indexovaného labely shluků. Kde na každém indexu je pole řetězců názvů všech obrazů ze shluku. To se následně projde a do vytvořené složky *Output/* se vytvoří adresáře jednotlivých shluků. Sem se uloží všechny fotografie patřící do tohoto shluku.

4.3 Reprezentativní fotografie

Jakmile je shlukování u konce přichází na řadu výběr co nejlepší reprezentativní fotografie. Tento proces je součástí dříve zmíněného *clusterPhotos.cpp*. Celý tento proces výběru reprezentativní fotografie vychází z teorie, že vhodná reprezentativní fotografie je nejvíce podobná s ostatními obrazy ve shluku. Jelikož je v tuto chvíli již provedeno shlukování a ve shluku se nachází podobné obrazy, dal by se proces výběru reprezentativního obrázku realizovat i náhodným výběrem. Pro výběr té co možná nejlepší fotografie, je ovšem zapotřebí obrazy ve shluku navzájem porovnat.

Porovnání probíhá v cyklu procházejícím vytvořené shluky. V každém ze shluků se projdou fotografie, které do shluku patří. Vektor vizuálních slov každé z náležících fotografií je postupně porovnáván se všemi ostatními ve shluku. Pokud je podobnost větší než experimentálně určená hodnota, pak se fotografii přičte počítadlo s kolika obrazy je fotografie podobná. Toto se dá přirovnat k sestavování grafu vztahů. Ačkoli se nevytváří přímo vztahy mezi fotografiemi, je tento postup dostačující, protože není třeba znát přesně, se kterou další fotografií je aktuální obraz ve vztahu. Důležité je zjistit počet obrazů, se kterými je fotografie ve vztahu podobnosti. Zároveň se hledá maximum, s kolika obrazy je každá z fotografií podobná. Ta fotografie, která má největší počet k sobě podobných fotografií, je zapamatována do pole indexovaného číslem shluku.

Určené reprezentativní fotografie se postupně uloží do předem vytvořeného adresáře *Output/*, kde se vytvoří adresář pro vybrané reprezentativní fotografie z názvem *representatives/*. Zde se fotografie uloží a pojmenují se názvem shluku, do kterého patří. Zároveň se i všechny ihned zobrazí, aby uživatel viděl, které jsou vybrané jako reprezentativní.

5 Dosažené výsledky

V této části bude popsán postup testování algoritmu popsaného v předchozí kapitole. Navržený a realizovaný program byl testován na několika množinách vstupních fotografií. Každá z množin obsahovala zhruba 200 až 300 fotografií. Přičemž v každém z datasetů bylo zastoupeno několik tříd fotografií, které obsahovaly podobné objekty. Třídy byly při jednotlivých testech různě promíchány. Všechny obrazy ze vstupních množin měly přibližně stejnou velikost.

Testovacích tříd bylo celkem pět a zahrnovaly fotografie automobilů, cyklistických kol, Eiffelovy věže, blesků na obloze, budov a města obecně. Testování probíhalo kombinací těchto skupin a různým zastoupením počtu jejich fotografií. Předmětem testování bylo především zjištění úspěšnosti správného shlukování.

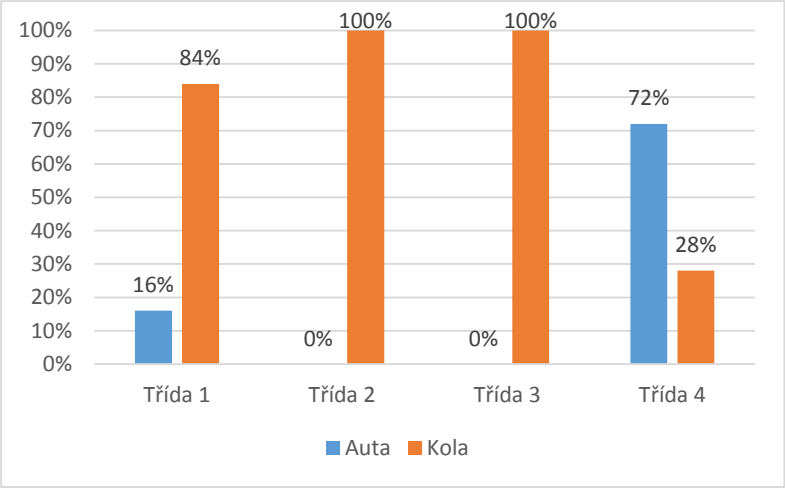
Obrázek 5.1. Ukázka několika fotografií z testovacích tříd.

5.1 Testování

První testovací množina byla sestavena ze 101 fotografií kol a asi 85 fotografií automobilů. Vytvoření slovníku trvalo přibližně 18 minut, musím ale podotknout, že testování probíhalo na osobním notebooku, který není zrovna nejnovější. Program následně rozdělil fotografie do čtyřech shluků. Rozdělení je znázorněno v tabulce níže. Procentuální vyjádření rozdělení jsou zobrazeny v grafu na obrázku pod tabulkou.

	Třída 1	Třída 2	Třída 3	Třída 4
Auta	9	0	0	76
Kola	48	10	14	29

Tabulka 5.1. Rozdělení fotografií v jednotlivých shlucích.



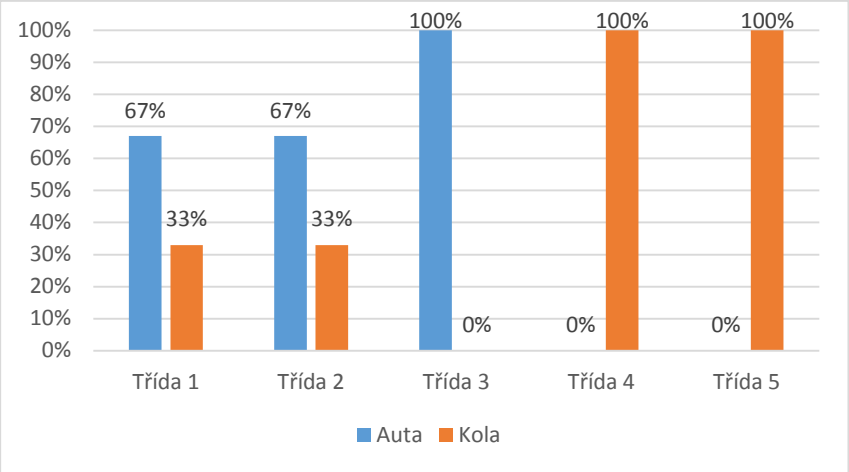
Obrázek 5.2. Graf rozdělení fotografií.

V každé z tříd byla jako reprezentativní vybrána fotografie kola a to i navzdory tomu, že ve třídě číslo 2 je výrazná převaha obrázků automobilů. Podle mého názoru, je toto způsobeno tím, že ačkoli je v tomto shluku těchto fotografií méně, jsou si navzájem více podobné. V dalším testu proto byly počty obrazů ve vstupní množině vyrovnány.

V následujícím experimentu program vytvořil o jeden shluk navíc než v předchozím. Konkrétní počty zastoupení fotografií v jednotlivých shlucích jsou zobrazeny v tabulce. Zde je vidět, že v jednom ze shluků je pouze jediná fotografie. Toto je způsobeno určitou nepřesností při shlukování.

	Třída 1	Třída 2	Třída 3	Třída 4	Třída 5
Auta	59	25	1	0	0
Kola	28	12	0	13	32

Tabulka 5.2. Rozdělení fotografií při stejném počtu fotografií.



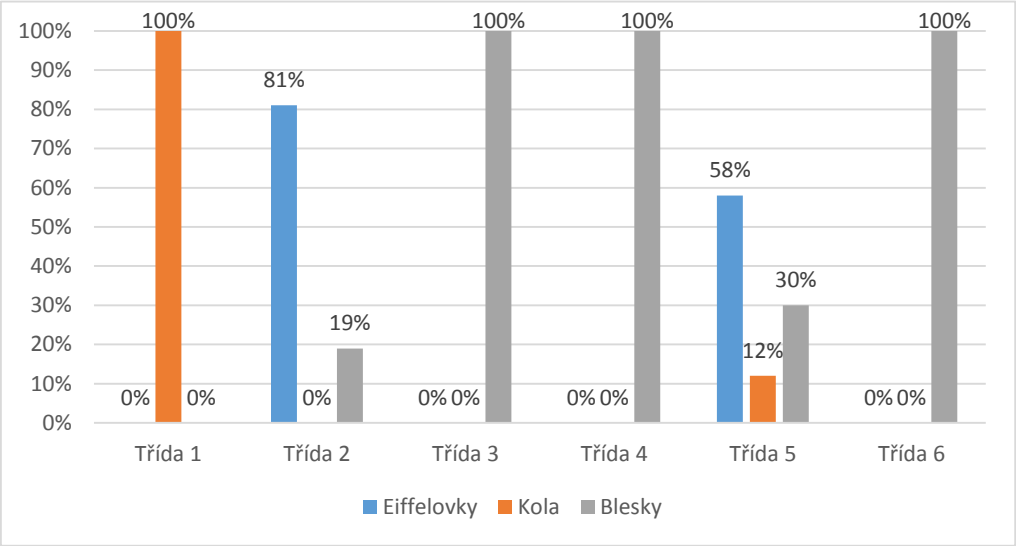
Obrázek 5.3. Graf rozdělení fotografií při druhém testu.

Tentokrát nebyl výběr reprezentativní fotografie tak jednostranný. Pouze pro třídu číslo jedna je nepřesně vybráno kolo, u ostatních tříd je vybráno podle očekávání. Z grafů těchto dvou testů lze vidět, že cyklistická kola jsou rozpoznávána lépe než auta. K tomuto bych ještě dodal, že všechny obrazy, kde byla pouze kola na bílém pozadí (podobně jako v ukázce na začátku kapitoly) zastávají v grafech oněch 100%.

V dalším testu byla vstupní množina fotografií složena z 50 fotografií kol, 50 blesků osvětlujících oblohu a 100 fotografií Eiffelovy věže. Sestavení slovníku zabralo opět zhruba 20 minut. Program z těchto 200 fotografií tentokrát vytvořil šest shluků.

	Třída 1	Třída 2	Třída 3	Třída 4	Třída 5	Třída 6
Eiffelovky	0	30	0	0	70	0
Kola	35	0	0	0	15	0
Blesky	0	7	6	1	34	2

Tabulka 5.3. Třetí test.



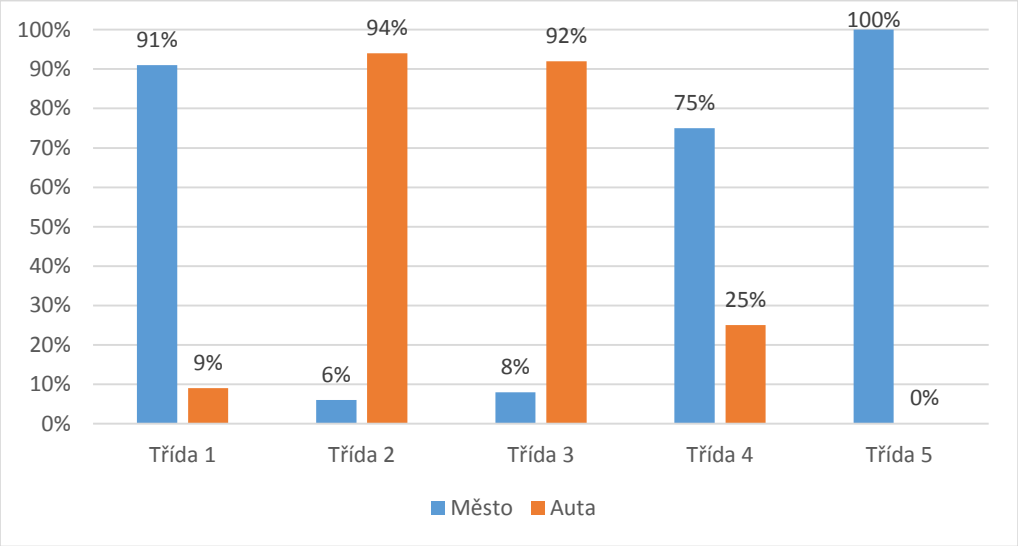
Obrázek 5.4. Procentuální vyjádření rozdělení fotografií při třetím testu.

V tomto testu se většina fotografií rozdělila do čtyřech shluků, opět se vytvořily i skupiny, kde jsou pouze dvě a jedna fotografie. Z grafu je patrné, že fotografie blesků a Eiffelovy věže jsou si v určitých směrech asi podobné. Toto je způsobeno i tím, že některé blesky jsou zachyceny nad městem, tudíž je výsledek mírně zkreslen budovami. Pro každou třídu byly jako reprezentativní vybrány fotografie ze skupiny s největším zastoupením (jak ukazuje graf).

Další z experimentů zahrnoval 100 fotografií města dohromady se 100 fotografiemi automobilů. Jelikož počet obrazů je stejný jako v předchozím testu, i doba sestavení slovníku zabrala stejný časový úsek. Z této vstupní množiny čítající 200 fotografií vytvořil program pět tříd.

	Třída 1	Třída 2	Třída 3	Třída 4	Třída 5
Město	30	2	2	64	2
Auta	3	32	24	21	0

Tabulka 5.4. Rozdělení fotografií do jednotlivých tříd při čtvrtém testu.



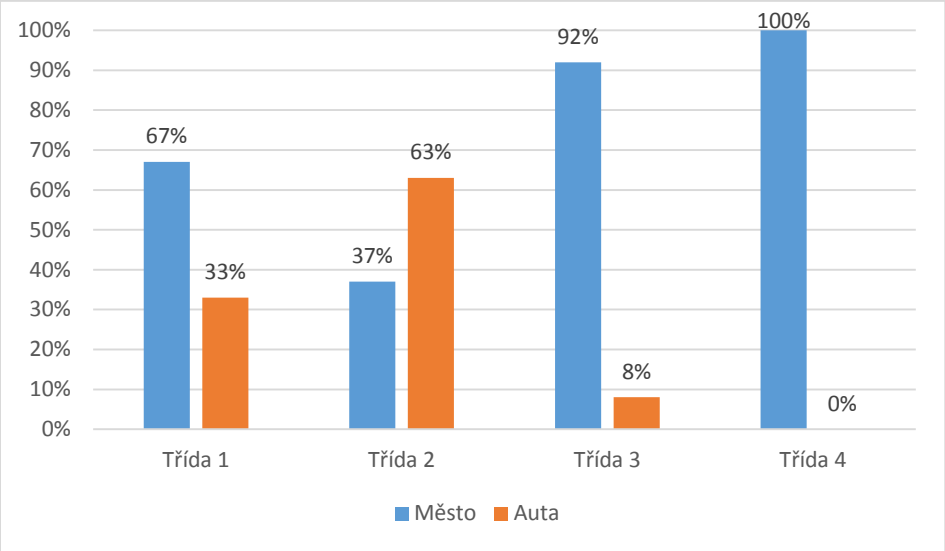
Obrázek 5.5. Graf zobrazující obsazení výsledných tříd v procentech.

Z tabulky i z grafu vyplývá, že úspěšnost správného rozdělení je v tomto testu poměrně vysoká. Což je nejspíše dáno poměrně rozdílnými skupinami fotografií, ze kterých byla sestavena vstupní množina. Jako reprezentativní byly pro každou z výsledných tříd správně vybrány fotografie většinového zastoupení.

Poslední test zkoumá, jaká bude změna, pokud se změní poměr počtu fotografií. Oproti předchozímu testu bylo použito 80 fotografií aut a 200 fotek měst, tedy počet fotografií měst se vůči vozům zvětšil více než dvojnásobně. Vytvoření slovníku tentokrát zabralo 40 minut.

	Třída 1	Třída 2	Třída 3	Třída 4
Město	2	39	158	1
Auta	1	66	13	0

Tabulka 5.5. Tabulka posledního testu.



Obrázek 5.6. Graf posledního testu.

Na rozdíl od předchozího testu byly vytvořeny jen čtyři výsledné třídy. Podobně jako v předchozím testu zde vidíme poměrně úspěšné oddělení fotografií měst. Pro první dvě třídy byly reprezentativní fotografie vybrány z procentuálně menšího zastoupení fotografií.

6 Závěr

Tato práce patří do oboru počítačového vidění a jejím cílem bylo prostudovat algoritmy zpracování obrazu a podobnosti. Následně navrhnout a implementovat algoritmus pro seskupení obrazů zachycující stejný objekt a pro každou ze skupin vybrat vhodnou reprezentativní fotografii. Navržený algoritmus byl implementován v jazyce C++ s použitím zmiňovaných knihoven OpenCV a OpenMP. Volba těchto open source knihoven implementaci výrazně zjednodušila i zefektivnila.

Metod pro zpracování obrazu i určování podobností v obrazech je velké množství a jsou využívány ve spoustě vědních oborů, ale i v každodenním životě (např. fotoaparáty s detekcí úsměvu atd.). V úvodní části práce jsou rozepsány základní, ale i pokročilejší přístupy a metody počítačového vidění.

Pro implementaci byl zvolen model Bag of Words, jehož síla je v jeho jednoduchosti a zároveň osvědčenosti [6]. Seskupení obrazů podle podobnosti bylo provedeno na základě detekovaných významných bodů. Pro detekci těchto bodů a extrakci deskriptorů z obrazů byl použit algoritmus SIFT [8]. Ten byl zvolen pro jeho vlastnosti (invarianci vůči měřítku, rotaci atd.) a přesnost. Slovník je vytvořen metodou shlukování extrahovaných deskriptorů. Ke klasifikaci obrazů je použita učící metoda *k*-means. Ta se řadí mezi metody učení bez učitele, kdy není známa tréninková množina. Pro každý z obrazů je ze slovníku vypočítán vektor vizuálních slov. Zmíněná metoda *k*-means poté z těchto vektorů vytvoří shluky [18]. Tímto je proces seskupení fotografií zachycující stejný objekt u konce. Z výsledných tříd fotografií je poté vybrána reprezentativní fotografie. Za reprezentativní je prohlášena ta fotografie, která má k sobě ve skupině nejvíce podobných ostatních fotografií [11]. Pro uložení je vytvořen adresář *Output/*, do kterého se vytvoří adresáře jednotlivých tříd. Také je zde vytvořen adresář *representatives/*, kde jsou uloženy vybrané reprezentativní fotografie.

Zrealizovaný program byl otestován na několika množinách fotografií. Každá z těchto vstupních množin sestávala z několika tříd stejných objektů. Obrázky použité pro testování byly staženy ze serveru Flickr. Cílem prováděných testů bylo zjistit úspěšnost seskupení stejných objektů do výsledných skupin. Z uskutečněných experimentů lze vyčíst, že program vstupní množinu fotografií rozdělil ve většině případů správně. Také je nutné podotknout, že toto rozdělení velice závisí na fotografiích jako takových. Například v prvním testu se ve 4. třídě nacházelo 72% fotografií automobilů a 28% bicyklů. V těchto 28% se nacházely fotografie s pozadím podobným pozadí u automobilů, tzn. fotografie auta na ulici a kola na ulici je klasifikována do stejné třídy. Stejně tak jsou tímto ovlivněny i ostatní testy.

Podobným problémem trpí i výběr reprezentativní fotografie. Jak je vidět u prvního testu, kde pro 4. třídu, ve které mají převahu fotografie automobilů, byla vybrána fotografie kola. Při přímém porovnání fotografií jsou si totiž kola navzájem více podobná. I přes tyto nepřesnosti dávaly testy uspokojivé výsledky.

Literatura

[1] ANTOŠOVSKÝ, Jaroslav. *Registrace obrazu*. Plzeň, 2013. Dostupné z: https://otik.uk.zcu.cz/bitstream/handle/11025/10427/BP_Jaroslav_Antosovsky_A10B0829P_final.pdf. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra kybernetiky. Vedoucí práce Ing. Tomáš Ryba.

[2] ŠŤASTNÝ, Petr. *Moderní metody identifikace objektů*. Brno, 2010. Dostupné z: http://is.muni.cz/th/255824/fi_b/Metody_BP.pdf. Bakalářská práce. MASARYKOVA UNIVERZITA, FAKULTA INFORMATIKY. Vedoucí práce prof. RNDr. Jiří Hřebíček, CSc.

[3] ŠŤASTNÝ, Jiří. *Netradiční metody a algoritmy pro rozpoznávání objektů technologické scény*. Brno, 2006. ISBN 80-214-3117-2. Dostupné z: http://www.vutium.vutbr.cz/tituly/pdf/ukazka/80-214-3117-2.pdf. Habilitační práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta strojního inženýrství.

[4] PRINCE, Simon J. *Computer vision: models, learning, and inference* [online]. New York: Cambridge University Press, 2012, xi, 580 s. [cit. 2015-01-23]. ISBN 978-1-107-01179-3. Dostupné z: http://www.computervisionmodels.com/.

[5] Bag of Words Models for visual categorization. In: *Gil's CV blog* [online]. 2014 [cit. 2015-01-23]. Dostupné z https:/gilscvblog.wordpress.com/2013/08/23/bag-of-words-models-for-visual-categorization/.

[6] GRAUMAN, Kristen a Bastian LEIBE. *Visual Object Recognition* [online]. [cit. 2015-03-29]. Dostupné z: http://cs.gmu.edu/~kosecka/cs482/grauman-recognition-draft-27-01-11.pdf

[7] TSAI, Chih-Fong. Bag-of-Words Representation in Image Annotation: A Review. *ISRN Artificial Intelligence* [online]. 2012, vol. 2012, s. 1-19 [cit. 2015-01-23]. DOI: 10.5402/2012/376804. Dostupné z: http://www.hindawi.com/journals/isrn.artificial.intelligence/2012/376804/

[8] A Short introduction to descriptors. *Gil's CV blog* [online]. 2013 [cit. 2015-01-23]. Dostupné z: https://gilscvblog.wordpress.com/2013/08/18/a-short-introduction-to-descriptors/

[9] Visual descriptors. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2015-01-23]. Dostupné z: http://en.wikipedia.org/wiki/Visual_descriptors

[10] Scale-invariant feature transform. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2015 [cit. 2015-01-23]. Dostupné z: http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[11] CHU, Wei-Ta a Chia-Hung LIN. Automatic Selection of Representative Photo and Smart Thumbnailing Using Near-Duplicate Detection. *National Chung Cheng University: Computer Science* [online]. [cit. 2015-01-23]. Dostupné z: http://www.cs.ccu.edu.tw/~wtchu/papers/2008MM-chu.pdf

[12] ČADÍK, Martin. *Image Registration Methods.* (předáška) [online]. Brno [cit. 2015-03-17]. Dostupné z: https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_10_registrace_obraze_2014.pdf

[13] KARAS, Pavel. *Studium metod registrace obrazu* [online]. Brno, 2009 [cit. 2015-03-21]. Dostupné z: http://is.muni.cz/th/106808/prif_m/dp.pdf. Diplomová práce. MASARYKOVA UNIVERZITA, PŘÍRODOVĚDECKÁ FAKULTA.

[14] KALOVÁ, Ilona. *Segmentace a detekce geometrických primitiv*. (přednáška) [online]. Brno [cit. 2015-03-21]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/lectures-pdf/09_Segmentace_obrazu.pdf

[15] ŠPANĚL, Michal a Vítězslav BERAN. *Obrazové segmentační techniky* [online]. Brno, 2005 [cit. 2015-03-22]. Dostupné z: http://www.fit.vutbr.cz/~spanel/segmentace/

[16] BAY, Herbert, Andreas ESS, Tinne TUYTELAARS a Luc VAN GOOL. *Speeded-Up Robust Features* [online]. 2008 [cit. 2015-05-17]. Dostupné z: ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf

[17] ŠOCHMAN, Jan. *Shlukování k -means* [online]. 2005 [cit. 2015-05-17]. Dostupné z: http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/kmeans/kmeans.pdf

[18] PIECH, Chris. *K Means* [online]. 2013 [cit. 2015-05-17]. Dostupné z: http://stanford.edu/~cpiech/cs221/handouts/kmeans.html

[19] SHIRAHAMA, Kimiaki. *Bag of Visual Words and Support Vector Machine* [online]. [cit. 2015-05-17]. Dostupné z: http://www.pr.informatik.uni-siegen.de/sites/www.pr.informatik.uni-siegen.de/files/courses/Winter14-15/MMR/p10_gor1.pdf

Seznam příloh

Příloha 1 – Manuál k použití programu

Pro přeložení je nejdříve nutné zavolat cmake project/ a vygenerovaným Makefilem aplikaci přeložit. Aplikace je psaná v C++11.

Spuštění

`./createVocabulary <cesta_k_adresaři_fotografií>` => vytvoření slovníku

`./clusterPhotos` => seskupení a výběr reprezentativní fotografie

Příloha 2 – Obsah přiloženého CD

Název adresáře	**Obsah adresáře**
Text	Bakalářská práce v elektronické podobě
Program	Adresář programu a zdrojové texty
Libs	Použité knihovny