VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

Ústav počítačové grafiky a multimédií
Department of Computer Graphics and Multimedia

Vyhledávání podobných CT dat podle obsahu

Content-Based Searching for Similar CT Scans

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — Pavel Hošek
AUTHOR

VEDOUCÍ PRÁCE — Ing. Michal Španěl, Ph.d.
SUPERVISOR

Brno, 2018

Zadání bakalářské práce/18489/2017/xhosek11

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Hošek Pavel**
Obor: Informační technologie
Téma: **Vyhledávání podobných CT dat podle obsahu**
Content-Based Searching for Similar CT Scans
Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte dostupné materiály na téma zpracování obrazových dat a metody detekce význačných bodů. Seznamte se s problematikou zpracování medicínských CT dat a jejich sesouhlasení a porovnání.
2. Vyberte vhodné metody a navrhněte nástroj pro vyhledávání v databázi naskenovaných CT dat, kdy dotazem jsou vzorová data a případná ruční anotace.
3. Experimentujte s vaší implementací a případně navrhněte vlastní modifikace metod.
4. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
5. Vytvořte stručný plakát prezentující vaši diplomovou práci, její cíle a výsledky.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění prvních dvou bodů zadaní.

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

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

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

Vedoucí: **Španěl Michal, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

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

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Práce se zabývá návrhem a implementací nástroje pro vyhledávaní CT dat podle podobnosti. Zabývá se také porovnáním jednotlivých metod pro vyhledávání CT dat. Využívá knihovny třetích stran pro extrakci různých deskriptorů a vyhledávání v nich. Práce prezentuje výsledky implementovaného vyhledávacího nástroje a výsledky porovnání jednotlivých metod a navrhuje možná vylepšení.

Abstract

The work deals with the design and implementation of tool for Content-Based Searching for similar CT scans. This work also deals with comparision of individual methods for searching for similar CT scans. Third-party libraries are used to extract individual descriptors. Furthermore, this work presents results of the implemented searching tool and also presents results of the comparision of individual methods for Content-Based Searching for similar CT scans and suggests possible improvements.

Klíčová slova

Vyhledávání, podobnost, CT snímky, CT série, obrazové vlastnosti, deskriptor, databáze, Java.

Keywords

Searching, similarity, CT scans, CT series, image features, descriptor, database, Java.

Citace

HOŠEK, Pavel. *Vyhledávání podobných CT dat podle obsahu*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Španěl Michal.

Vyhledávání podobných CT dat 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.

……
Pavel Hošek
17.května 2018

Poděkování

Rád bych poděkovali vedoucímu práce Ing. Michalovi Španělovi,Ph.D. za odborné vedení a rady při psaní této práce. Rád bych také poděkoval Ing. Václavu Hoškovi za konzultace a cenné rady v oblasti implementace.

Obsah

1 Úvod

Digitální obrázky jsou v dnešní době důležitým nástrojem pro uchování obrazových dat. Se stále zvyšujícím se počtem obrázků, rostou také databáze, které je uchovávají. Problém nastává při manuálním hledání v obrovských databázích, které obsahují vysoký počet obrázků. Člověk, který bude chtít najít určité obrázky v databázi která obsahuje 50 obrázků, pravděpodobně nebude mít problém najít to co hledá v rozumném čase. Pokud bude ale databáze obsahovat tisíce, desetitisíce či statisíce obrázků, stane se takové manuální hledaní velmi časově náročné, a ne příliš efektivní. Tato práce se výhradně zaměřuje na obrázky z lékařského prostředí, konkrétně obrázky získané metodou výpočetní tomografie. Výpočetní tomografie neboli CT je radiologická vyšetřovací metoda, která pomocí rentgenového záření umožňuje neinvazivní zobrazení vnitřních orgánů a tkání člověka či zvířat s vysokou rozlišovací schopností a ve 3D projekci[1].Lékař si může chtít vyhledat podobné snímky jako má pacient, kterého právě léčí, aby zjistil jak byli předchozí pacienti s podobnými příznaky léčeni a jaké to mělo výsledky.

Jedna z možností, jak hledat obrázky v takové databázi je popsat každý snímek textem a poté hledat snímky na základě textových informací. Nevýhody této metody jsou velká časová náročnost pro popsání každého snímku a také že ne každý snímek lze popsat slovy. Další z možností, jak hledat obrázky je podle jejich dat neboli obsahu, na což se zaměříme ve zbytku této práce. Vyhledávání podle obsahu znamená, že systém extrahuje některé z obrazových vlastností (tzv. features) obrázku a na základě nich poté vyhledá obrázky podobné. Při hledání podle dat v databázi obrázků nebudeme hledat přesně nýbrž bude nás zajímat obrázek či obrázky které jsou nejpodobnější hledanému obrázku.

V této práci budeme zkoumat a experimentovat s různými vlastnostmi obrázků, které jdou z nich automaticky extrahovat a lze je také použít při hledání podle dat. Jako příklady vlastností obrázků lze uvést hodnotu jednotlivých pixelů či histogram barev vyskytující se v obrázku atd. Hlavním přínosem a cílem této práce je najít vhodné metody a vytvořit nástroj pro vyhledávání v databázi naskenovaných CT dat, kde dotazem jsou vzorová data .Pro implementaci nástroje jsem využil open source knihovnu Lire. Lire je Java knihovna která implementuje extrakci různých vlastností obrázku a také metody pro vyhledávaní.

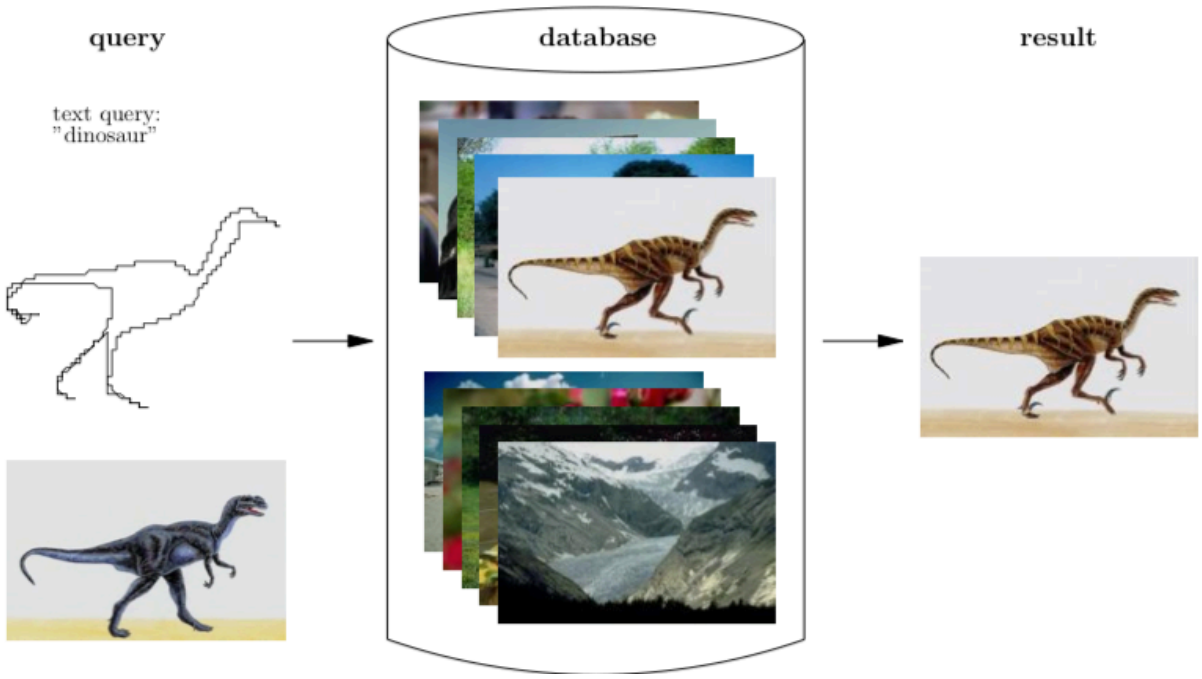
Práce je členěna do několika kapitol. První čtyři shrnují teorii potřebnou k pochopení obsahu práce. Kapitola číslo 5 se zabývá metrikami pro porovnání jednotlivých metod. Kapitola číslo 6 se zabývá návrhem řešení cílů této práce. Samotná implementace je pak popsána v kapitole číslo 7. V kapitole číslo 8 je uvedeno zhodnocení nástroje pro vyhledávání a zhodnocení kvality vyhledávaní jednotlivých metod. V kapitole číslo 9 je uveden závěr.

2 Základní principy pro vyhledávání obrázků

Při vyhledávání v databázi obrázků mohou být použity následující typy dotazů.

- **Text jako dotaz:** Uživatel zadá textový popis obrázku, který hledá
- **Náčrt jako dotaz:** Uživatel dodá náčrt obrázku, který hledá, vyrobí ho typicky pomocí kreslícího nástroje
- **Obrázek jako dotaz:** Uživatel použije jako dotaz obrázek, a hledá jemu podobný.

V této práci se zaměříme na přístup, kdy bude obrázek použit jako dotaz. Vyhledávání obrázků podle obsahu (anglicky content-based image retrival-CBIR), je soubor technik, které hledají v obrázcích pomocí automaticky odvozených vlastností (features), jako je barva, struktura či tvar. Abychom získali odvozené vlastnosti obrázků musíme je nejprve extrahovat. Extrakcí vlastností vznikne deskriptor či deskriptory, které daný obrázek popisují. Datový typ deskriptoru se liší v závislosti na extrahované vlastnosti obrázku. Například pro histogram barev v obrázku, kde se bude vyskytovat 10 různých barev, bude deskriptor reprezentován jako vektor deseti čísel. Každé číslo bude vyjadřovat počet pixelů dané barvy v obrázku. Pro zjištění míry podobnosti obrázků se pak deskriptory jednotlivých obrázků porovnají např. (Jensen-Shannon metodou, Euklidovou metodou atd..). Poté při vyhledání podobných obrázků, se buď určí hranice míry podobnosti nebo se vybere určitý počet obrázků, které mají od porovnávaného obrázku nejmenší velikost.



Obrázek 2.1: Content based image retrival(CBIR) [1]

Definice vyhledávání, kdy dotaz do databáze je obrázek by mohla vypadat následovně:
Nechť Q je vstupní obrázek, R je výstupní množina obrázků, n je počet vyhledávaných obrázků, D je množina deskriptorů, A je množina všech obrázků, tedy $R \subset A$ a d je desktiptor pro vstupní obrázek tedy $d \in D$. Potom pro každé $x \in D$ se spočítá vzdálenost vůči d a vybere se n obrázků s nejmenšími vzdálenostmi, které tvoří množinu R.

3 Vlastnosti obrázků pro Content based image retrival

Hlavní myšlenka CBIR techniky je nespoléhat se na textový popis obrázku. Místo toho používá soubor vlastností, které lze získat z obrázku a umožní uživateli najít obrázky visuálně podobné hledanému obrázku.

Pojem podobné zde může znamenat různé věci, takový lékař, který pracuje s radiologickými obrázky může mít jiná kritéria na podobnost než například vývojář webu, který chce najít obrázek pro svůj web. Z čehož plyne to že nelze splnit různá kritéria na podobnost obrázků pouze jednou metodou. Existují různé způsoby, jak popsat obrázek, což je zajištěno různými vlastnostmi, které lze z obrázku extrahovat. Většinou jde rozdělit vlastnosti do skupin, podle toho, jakou vlastnost obrázku obsahují, a to na texturní vlastnosti, vlastnosti barvy a vlastnosti tvaru. V této kapitole budou popsány různé vlastnosti, které slouží pro hledání obrázků podle obsahu.

3.1 Základní vlastnosti

Nejvíce přímočarý přístup pro hledání podle obrázků je porovnat obrázky přímo. Což znamená každý pixel obrázku se porovná s korespondujícím pixel jiného obrázku, který je upraven na stejnou velikost. Pro mnoho případů tento způsob není dobře proveditelný, jelikož nelze určit který pixel odpovídá kterému. Pokud například budeme uvažovat obrázek krajiny, kde bude jelen a ten samí obrázek, kde jelen nebude na stejné pozici, ale bude posunut v krajině, bude tento způsob fungovat velmi špatně. Hledání tímto způsobem bude fungovat pouze tehdy, pokud budou jednotlivé elementy obrázku mít stejnou velikost a budou na stejných místech v obrázku.

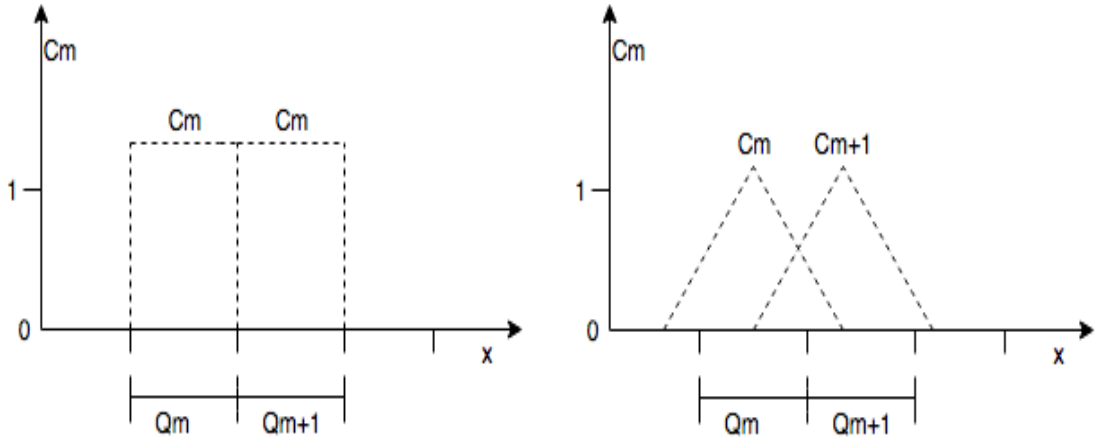
Existují různá vylepšení této metody. Filtry a transformace jsou použity pro upřednostnění jisté vlastnosti obrázku. Například aplikace Prewitt filtrů pro zvýraznění hran.

3.2 Histogram barev (Color Histogram)

Histogram barev reprezentuje rozložení barev v obrázku. Pro vytvoření histogramu je nejprve třeba rozdělit prostor barev do tříd. Například hojně používaný 24 bit RGB model bude mít 2^{24} skupin. Každá skupina vyjadřuje odstín, který se vyskytuje v modelu. Histogram potom obsahuje tolik sloupců, na kolik tříd je rozdělen barevný model. Z čehož vyplývá neefektivnost při ukládání. Proto je systém většinou kvantován. U tohoto přístupu je ale nutné najít rovnováhu mezi požadavkem na paměť a ztrátou dat.

Pro radiologické obrázky, které jsou většinou černobílé je situace daleko lepší, jelikož histogram bude obsahovat pouze 256 sloupců. Každý sloupec bude odpovídat stupni šedi v obrázku. Nevýhoda histogramů je, že jsou velmi citlivé na malé iluminační změny a kvantizační chyby [3]. Velmi podobné barvy budou přiřazeny do jiných tříd, a tím pádem bude histogram odlišný. Nicméně fuzzy přístup, řeší tento problém přiřazením každého pixelu do barevné třídy s určitým stupněm příslušnosti. Cílem fuzzy histogramů je odstranit nespojitost tradičního histogramu.

Obrázek 3.2.1: Ukázka histogramu v programu Adobe Photoshop.



a) Příklad funkce přiřazení do třídy v klasickém histogramu.

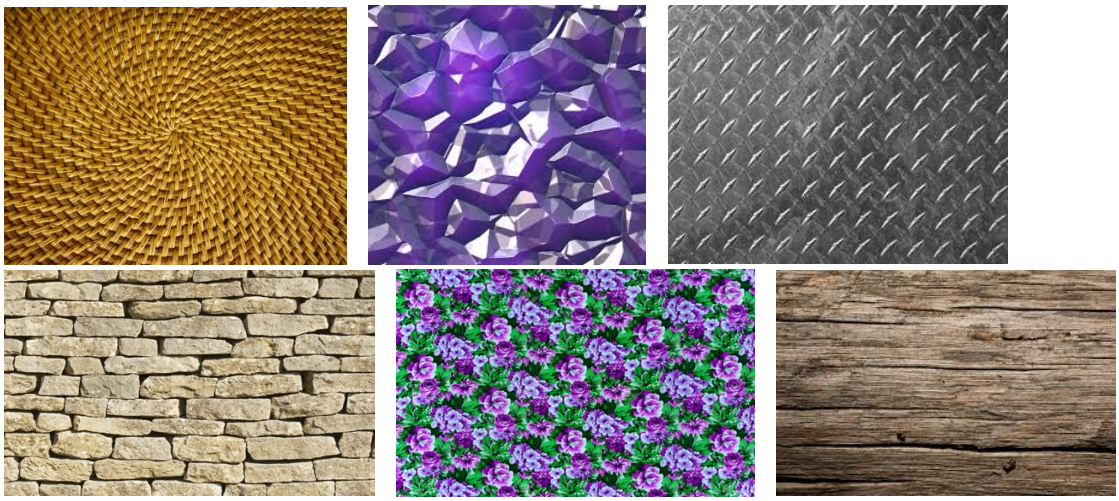
b) Příklad funkce přiřazení do skupiny v fuzzy histogramu.

3.3 Texturní vlastnosti

Textura je patří mezi nejdůležitější atributy při rozpoznávání věcí. Pro lidské vnímaní, textura většinou znamená specifickou opakující se strukturu povrchů, tvořenou opakováním určitého prvku nebo několika prvků v různých relativních prostorových polohách. Obrázkové textury jsou definovány, jako obrazy přírodních texturovaných povrchů a uměle vytvořených vizuálních vzorků, které v určitých mezích přibližují tyto přirozené objekty.

Je téměř nemožné popsat textury slovy, ačkoli existují definice, které zahrnují různé neformální strukturní rysy, jako je jemnost – hrubost, hladkost, zrnitost, směrovost, drsnost, pravidelnost – náhodnost a tak dále. Tyto pojmy, které definují prostorové uspořádání složek struktury, pomáhají rozlišit jednotlivé typy textur, např. jemný nebo drsný povrch fasády. Je obtížné používat klasifikaci, kterou používá člověk, jako základ pro formální definice obrazových textur. Jelikož neexistují žádné zřejmé způsoby, jak tyto rysy vnímané lidským zrakem zakomponovat do výpočetních modelů popisujících textury.

Nicméně po několika desetiletích výzkumu a vývoji analýzy textur, byly nalezeny různé výpočetní charakteristiky a vlastnosti, pro indexování a získávání textur. Vlastnosti textur popisuje lokální uspořádání obrazových signálů v prostorové doméně nebo v doméně Fourierovy či jiné spektrální transformace.



Obrázek 3.3.1: Ukázky textur[1].

[1] https://freestocktextures.com, https://cz.depositphotos.com

3.4 Tamura vlastnosti

V [4] autoři definovali 6 vlastností textur (hrubost, kontrast, směr, vertikální-podobnost, pravidelnost a drsnost). Dělali experimenty a porovnávali, jak tyto vlastnosti souvisí s lidským vnímáním. Tři z nich, autoři označily za velmi důležité (hrubost, kontrast a směr). Tyto tři vlastnosti velmi souvisí s lidským vnímáním obrazu.

Hrubost

Lze chápat jako velikost jednotlivých prvků textury. Je označována za nejdůležitější vlastnost textury. Berou se v potaz vždy největší prvky textury. Výpočet hodnoty hrubosti bude následující:

1) Spočítá se průměr každého bodu se svým okolím, kde velikost okolí mohou být mocniny 2

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{I(i,j)}{2^{2k}}$$

I(i,j) udává stupeň šedi v pixelu na souřadnicích (i,j), (x,y) jsou souřadnice počítaného bodu, k je koeficient.

2) Pro každý bod (n_0,n_1) se spočítá rozdíl mezi nepřekrývajícím se okolím na protějších stranách bodu a to v horizontálním a také vertikálním směru

$E_k^h(n_{0,}n_1) = |A_k(n_0 + 2^{k-1}, n_1)\text{-}\ A_k(n_0 - 2^{k-1}, n_1)|$ pro horizontální směr

a

$E_k^v(n_{0,}n_1) = |A_k(n_0, n_1 + 2^{k-1})\text{-}\ A_k(n_0, n_1 - 2^{k-1})|$ pro vertikální směr

3) V každém bodě (n_0,n_1) se vybere největší E_k.

$S(n_{0,}n_1) = argmax\ E_k^d(n_{0,}n_1)$

4) Posledním krokem je udělat průměr 2^S, čímž vypočítáme koeficient hrubosti obrázku

$$F_h = \frac{1}{N_0 N_1} \sum_{n0=1}^{N_0} \sum_{n1=1}^{N_1} 2^{S(n0,n1)} \qquad (3.1)$$

Kontrast

Ve vizuálním vnímání je kontrast určen rozdílem barvy a jasu objektů v rámci stejného zorného pole[4].Může být ovlivňován následujícími vlastnostmi.

- dynamický rozsah stupně šedi
- polarizace distribuce černé a bílé na histogramu zobrazujícím stupně šedi
- ostrost hran
- doba opakování vzorů v obrázku

Kontrast obrázku se vypočítá jako:

$$F_{con} = \frac{\sigma}{\alpha 4^{z}}, \alpha 4 = \frac{\mu^{4}}{\sigma^{4}} \tag{3.2}$$

kde $\mu 4 = \frac{1}{N_0 N_1} \sum_{n0=1}^{N_0} \sum_{n1=1}^{N_1} (X(n_0, n_1) - \mu 4)^4$ je čtvrtým centrální moment se střední hodnotou μ, σ^2je rozptyl hodnot stupně šedi v obrázku, (n_0,n_1) jsou souřadnice bodu a z bylo určeno experimentálně na $\frac{1}{4}$.

Směr

Je globální vlastnost v určitém regionu. Vyjadřuje orientaci v textuře a měří celkový směr.
Pro výpočet hodnoty, která udává směr, horizontální a vertikální derivace Δ_H a Δ_v jsou vypočítány konvolucí obrázku X(n_0, n_1) s následujícími operátory.

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

A pro každý bod (n_0, n_1) je spočítán směrný úhel α

$$\alpha = tan^{-1} \frac{\Delta_v (n_0, n_1)}{\Delta_H (n_0, n_1)} \tag{3.3}$$

Histogram H (α) kvantifikovaných směrových hodnot α je vytvořen počítáním pixelů na hranách s odpovídajícími směrovými úhly, které mají případně větší pevnost hrany, než je předem definovaná prahová hodnota. Histogram je poměrně uniformní pro snímky bez silné orientace. Hodnota směru se pak vypočítá jako suma druhých centrálních momentů okolo každého vrcholu.

Zbylé tři vlastnosti jsou velmi souvztažné s výše uvedenými třemi vlastnostmi a nepřispívají příliš k efektivnosti popisu textury. Ve většině případů se pro CBIR používají pouze první tři výše definované vlastnosti.

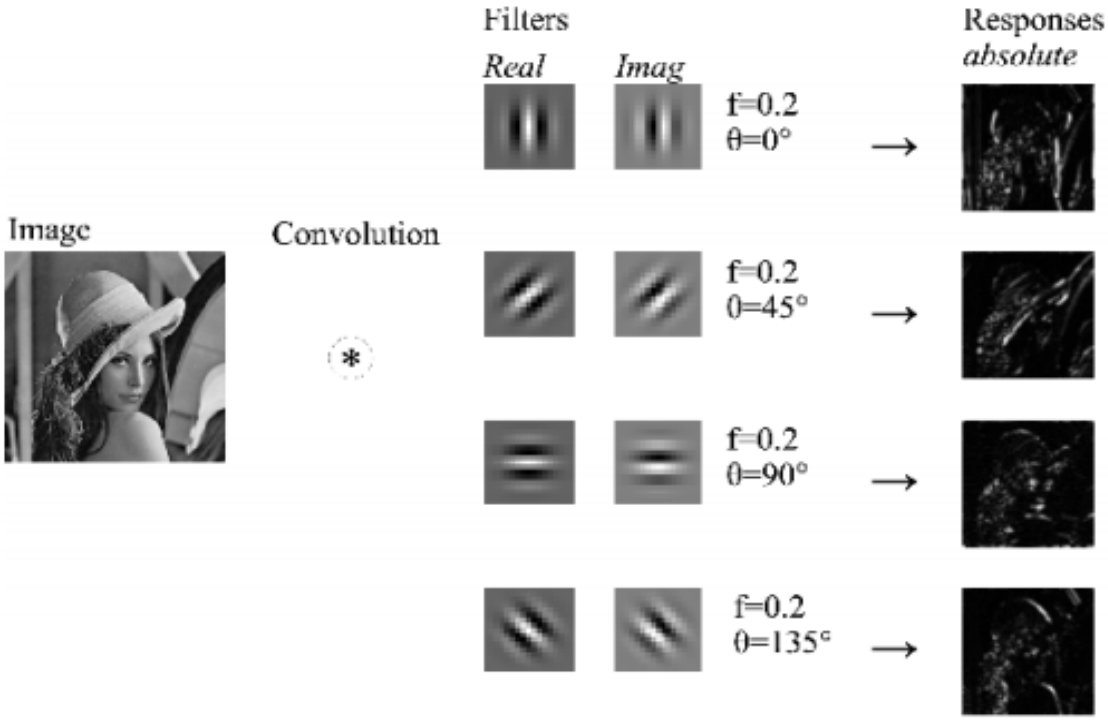
3.5 Gáborovy příznaky

Jsou to vlastnosti získané aplikací Gáborových filtrů. Gáborovi filtry jsou orientačně citlivé filtry, používané pro analýzu hran a textur. V podstatě analyzují, zda existuje nějaká specifická frekvence v obraze v daných směrech v lokalizované oblasti. Gáborovi filtry získaly v posledních letech značnou pozornost. Mohou napodobovat určité vlastnosti buněk ve zrakovém kortexu některých savců. Metody založené na Gáborových filtrech, byly úspěšně použity v mnoha aplikacích založených na strojovém vidění či zpracování obrazu. Jsou vůbec jedny z nejlepších metod pro rozpoznávání obličeje, rozpoznání duhovky či porovnávání otisků prstů. 2D Gáborův filtr lze definovat jako komplexní rovinnou vlnu násobenou Gaussovou obálkou.

$$\psi(x,y) = \frac{f^2}{\pi\gamma\eta} e^{\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\eta^2}y'^2\right)} e^{j2\pi f x'} \tag{3.4}$$

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta \tag{5}$$

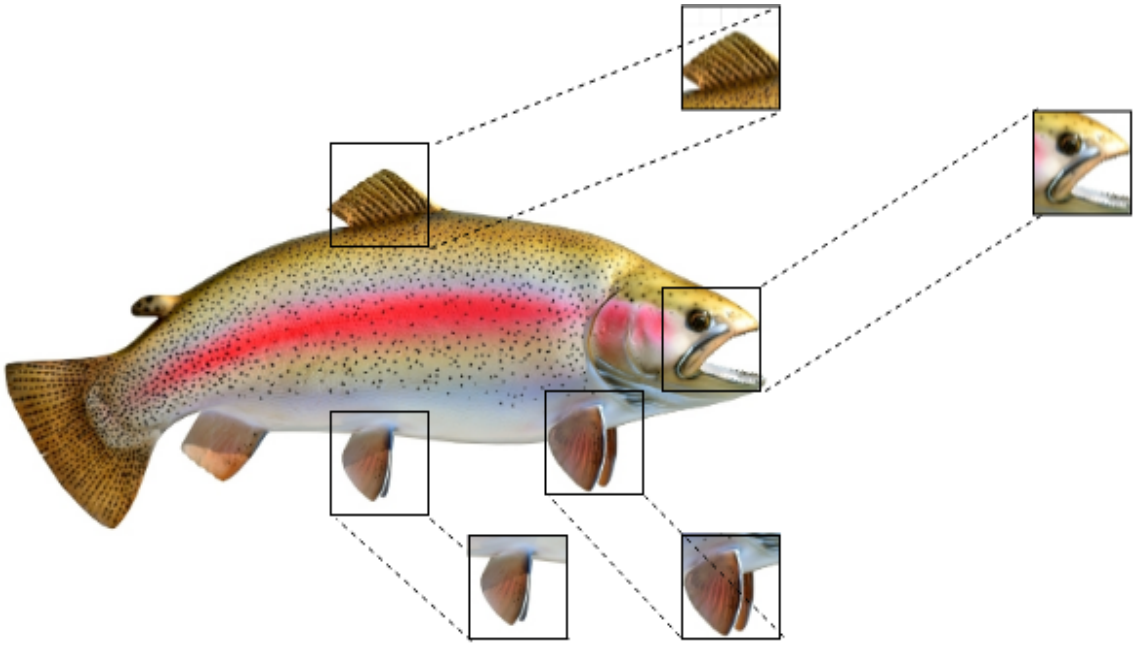
Kde f je frekvence filtru, θ je rotační úhel, γ je ostrost (šířka pásma) podél Gaussovi hlavní osy, a η je ostrost podél vedlejší osy (kolmá k vlně).



Obrázek 3.5.1 – Ukázka extrakce features za použití Gáborových filtrů[5].

3.6 Lokální vlastnosti

Lokální vlastnosti jsou vlastnosti, které jsou extrahovány z malých čtvercových obrázků v rámci jednoho obrázku. Lokální deskriptor nepopisuje celý obrázek, ale pouze malou část obrázku (určitou podmnožinu pixelů). Je známo že přístupy používající lokální vlastnosti obrázku mohou přinést velmi dobré výsledky při klasifikaci, např.(SIFT,SURF…). Nicméně počet lokálních vlastností extrahovaných z každého obrázku může být obrovský, obzvláště pokud uvažujeme dostatečně velkou databázi s obrázky. Počet extrahovaných vlastností v rámci jednoho celého obrázku se většinou pohybuje mezi 100 až 1000.



Obrázek 3.6.1: Extrahování lokálních vlastností.

Bag of visual words model

Bag of visual words dále (BoVW) model patří mezi nejefektivnější metody, jak zobrazit obrázek jako soubor lokálních vlastností. Vychází z modelu bag of words dále (BoW), který je běžně používaný při klasifikaci textu. V BoW modelu je každý dokument reprezentován jako neuspořádaná množina unikátních slov, které se vyskytují v dokumentu. Dokument je pak reprezentován jako histogram unikátních slov. Tyto histogramy jsou poté použity při klasifikaci dokumentu.

Podobný princip je také v BoVW modelu. Obrázek je reprezentován jako neuspořádaná množina unikátních lokálních vlastností. Tyto lokální vlastnosti jsou pak typicky reprezentované lokálními deskriptory,

kde počet těchto deskriptorů ve větší databázi obrázků může být obrovský. Deskriptory jsou pak shlukovány (např. metodou k-means) do skupin. Tyto skupiny jsou pak pojmenované jako visuální slova. Poté můžeme reprezentovat obrázek jako histogram těchto slov. Princip je ukázán na obrázku

Obrázek 3.6.2: Bag of visual words.[2]

3.7 SIFT vlastnosti

Scale Invariant Feature Transform (SIFT) je metoda pro detekci a extrakci lokálních deskriptorů, které jsou invariantní vůči osvětlení, šumu obrazu, rotaci a změně velikosti[6].Tato metoda byla navržena Davidem Lowem, v roce 2004 v Univerzitě British Columbia v Kanadě. Jedna z hlavních předností tohoto přístupu spočívá v tom, že dokáže správně rozpoznávat objekty v různých obrázcích. Objekty mohou být jinak natočené, mít jiné měřítko, jinak osvětlené, a přitom by měli být stále rozpoznány. I díky těmto vlastnostem se tato metoda používá v mnoha odvětvích, jako je například rozpoznávání objektů, robotické mapování a navigace, spojování obrázků (panoramata).

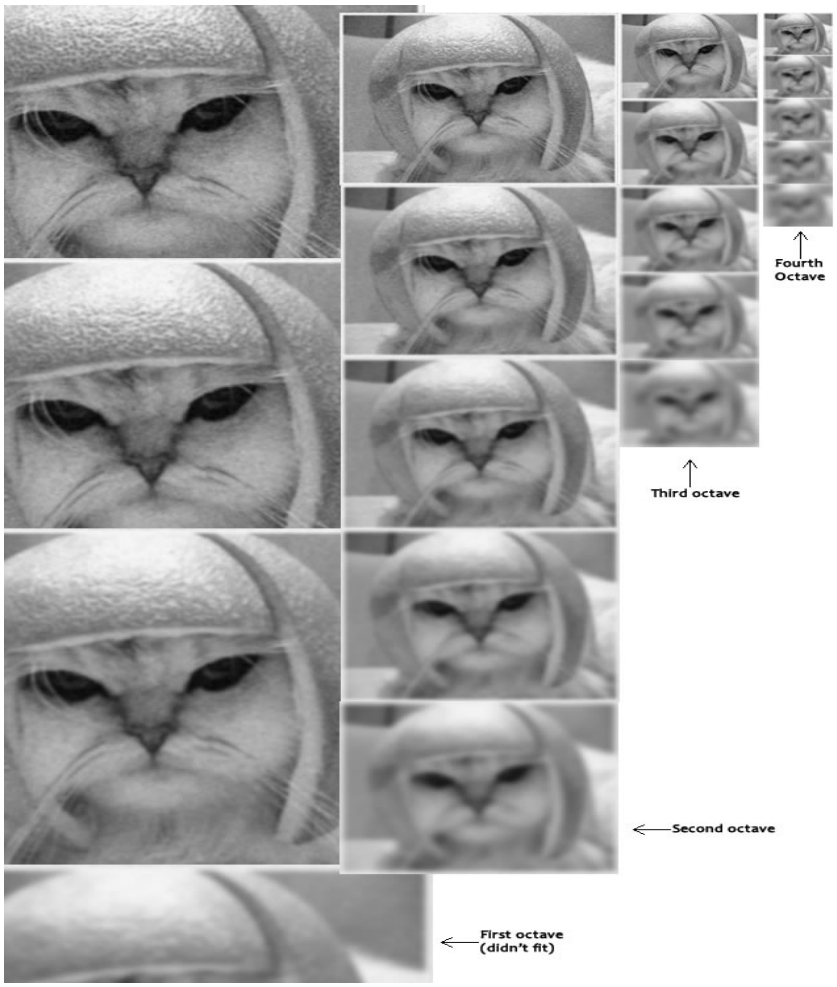
[2] : http://crcv.ucf.edu/courses/CAP5415/Fall2012/Lecture-17-BagOfWords.pdf

SIFT metoda ze skládá ze čtyř hlavních kroků:

- detekce extrému v scale-space
- určení polohy významných bodů (keypoints)
- přiřazení orientace významným bodům
- vytvoření deskriptorů pro významné body

Detekce extrému v scale-space

Cílem tohoto kroku je najít potencionálně významné body (kandidáty), jenž zůstávají stabilní i při změně měřítka obrazu. Nejprve je potřeba si vytvořit takzvané scale-space. Scale-space se člení do oktáv, které se skládají z vrstev stejného rozměru. Základem následující oktávy, je poslední vrstva oktávy před ní, která má poloviční rozměr. Jedna oktáva se dá definovat, jako sbírka obrázků získaná postupným vyhlazováním vstupního obrázku, což je analogický proces postupnému snižování rozlišení obrázku. Vyhlazení obrázku je realizováno konvolucí obrázku s Gaussovými filtry, vznikne tzv. Gaussovo rozostření obrázku. Vytvoření scale-space je znázorněno na obrázku 3.7.1.



Obrázek 3.7.1: Scale-space[3]

[3] http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/

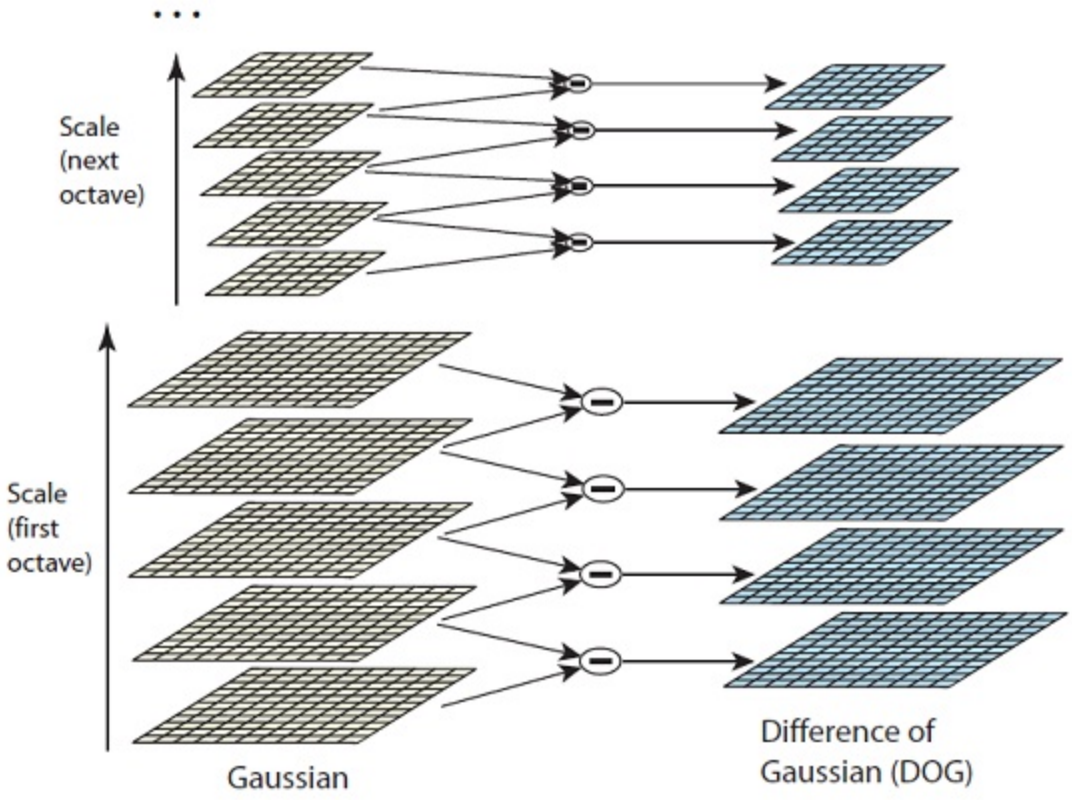
Dalším krokem je odečtení sousedních obrázků $L(x, y, k\sigma)$, v rámci oktávy, ve scale-space. Odečtení sousedního obrázku ve scale-space lze definovat následovně:

$$D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j, \sigma),$$

kde $L(x, y, k\sigma)$ je konvoluce původního obrázku $I(x, y)$, s Gaussovým rozostření $G(x, y, k\sigma)$,s měřítkem $k\sigma$.

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$$

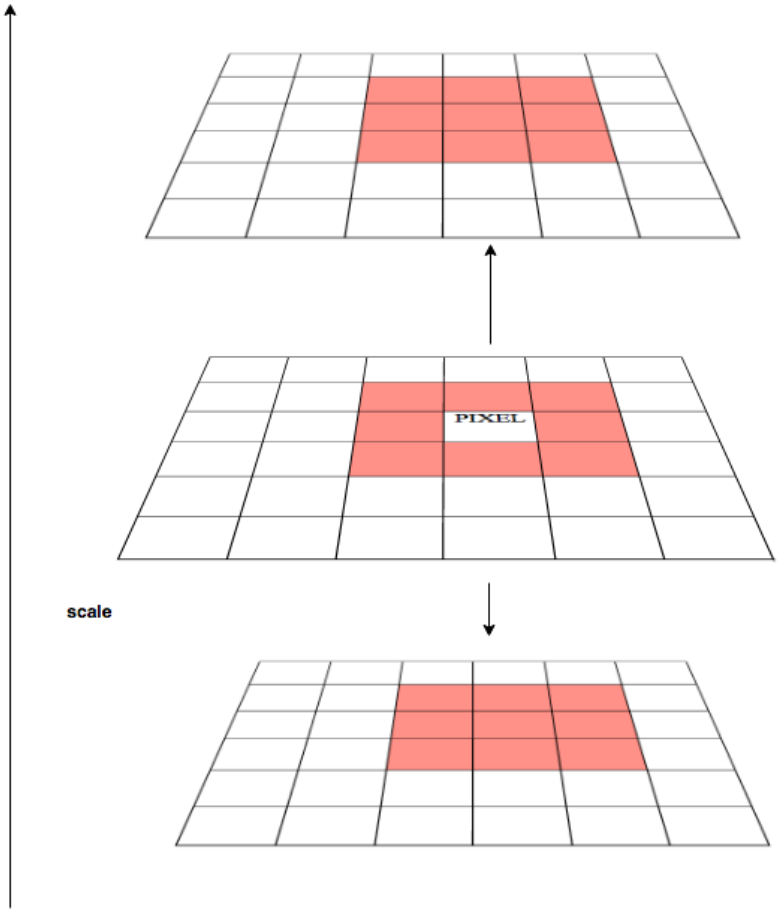
Odečítaní sousedních obrázků ve scale-space je znázorněno na obrázku. 3.7.2.



Obrázek 3.7.2: Princip odčítaní sousedních obrázků[4]

Odečtením sousedních obrázků dostáváme tedy D (x, y, σ). V obrázku D (x, y, σ) se hledají lokální maxima a minima. Hledá se porovnáváním každého bodu obrázku s 8 sousedními pixely a 9 pixely sousedních obrázků viz (Obrázek 3.7.3). Pokud takový bod má nejmenší, či největší hodnotu v rámci svého okolí, stává se kandidátem na významný bod.

[4] http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/



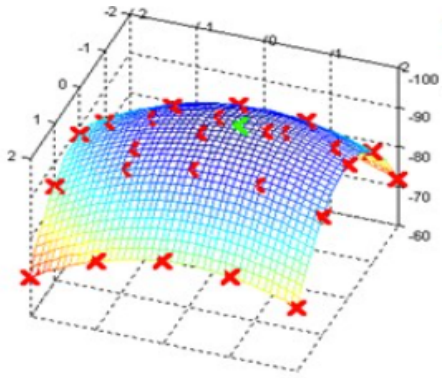
Obrázek 3.7.3: Hledání minima či maxima v rozdílových obrázcích.

Určení polohy významných bodů

Nalezení kandidáti v předchozím kroku byli určení pomocí přibližného maxima a minima. Přibližného, protože maxima či minima téměř nikdy neleží přesně na pixelu. Leží někde mezi pixely. Nelze ale přistupovat k datům mezi pixely. Takže se musí matematicky lokalizovat umístění v subpixelové oblasti. Pomocí dostupných pixelových dat se tedy generují subpixelové hodnoty. To se provádí rozšířením Taylorova rozvoje obrazu D(x,y,σ), kolem zkoumaného bodu x. Matematicky to lze zapsat jako:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} \tag{3.5}$$

Přesná pozice kandidáta v subpixelové oblasti je určena posunutím $\hat{x}$, které se spočítá jako extrém Taylorova rozvoje. Ten je spočten pomocí derivace funkce D (x) podle x a následným položením rovno nule. Pokud je posunutí $\hat{x}$ vetší než 0.5 v jakékoliv dimenzi, extrém se nachází blíže k sousednímu bodu. Pokud se tak stane, tak je původní bod nahrazen bodem sousedním a celý výpočet se opakuje. Pokud $\hat{x}$ nepřesáhne hodnotu 0.5, je výsledné posunutí přičteno k souřadnicím původního bodu. Po přičtení posunutí, je vypočítána pozice kandidáta v subpixelové oblasti.



Obrázek 3.7.4: Ukázka subpixelové oblasti červené body značí jednotlivé pixely v obrázku a zelený bod přesnou hodnotu maxima[5].

Odstranění bodů s nízkým kontrastem

Je vypočítána hodnota Taylorova rozvoje D(x) v bodě xˆ.

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x} \tag{3.6}$$

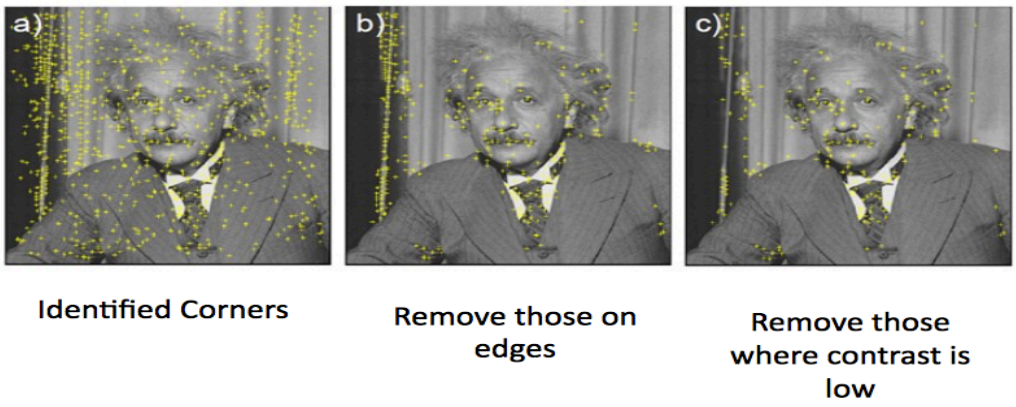
Pokud je hodnota |D (xˆ) | menší než 0,03 je kandidát z množiny významných bodů odebrán.

Odstranění bodů v blízkosti hran

Rozdíl Gaussových funkcí D (x, y, σ) má velkou odezvu v blízkosti hran obrazu. Tyto body je nutné eliminovat, jelikož jsou spatně lokalizovatelné a tím pádem nestabilní. Body v blízkosti hran vykazují velkou hlavní křivost podél hran a malou hlavní křivost v pravoúhlém směru vůči hraně. Hlavní křivost v okolí bodu lze vyjádřit pomocí vlastních čísel Hessianovy matice:

$$H = \begin{matrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{matrix}$$

Pokud je poměr velké hlavní křivosti podél hran a v pravoúhlém směru v daném rozmezí, je bod zanechán v množině významných bodů.



Obrázek 3.7.5: Ukázka odstranění bodů na hranách a s nízkým kontrastem[6]

[5] http://slideplayer.com/slide/1420241
[6] http://slideplayer.com/slide/4066021

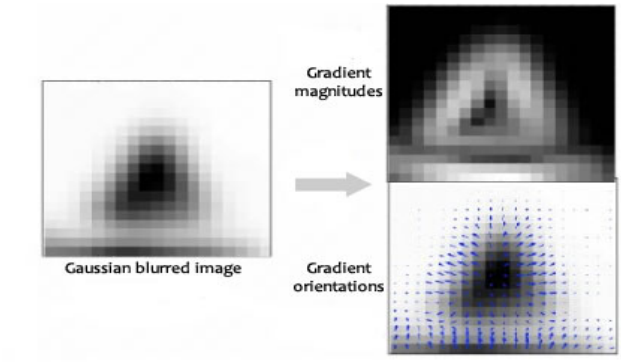
Přiřazení orientace významným bodům

V tomto kroku, každému významnému bodu bude přiřazena jedna či více dominantních orientací. Což zajistí invarianci vůči rotaci. Přiřazení orientace k významnému bodu probíhá následovně. Je nalezen obraz L (x, y, σ), který je měřítkově nejbližší k významnému bodu, což zajistí že všechny výpočty jsou prováděny invariantě vůči meřítku. Pro každý bod obrazu L (x, y, σ), který má měřítko σ, je spočtena velikost gradientu m (x, y) a orientace gradientu θ (x, y).

$$m(x,y) = \sqrt{(\mathrm{L}(\mathrm{x}+1,\mathrm{y}) - L(x-1,y)) + (L(x,y+1) - L(x,y-1))^2} \tag{3.7}$$

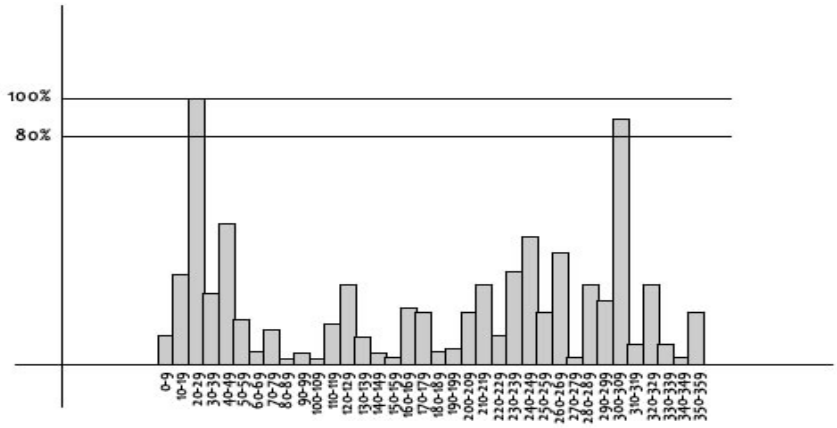
$$\theta(x,y) = tan^{-1}\left(\frac{(L(x,y+1) - L(x,y-1))}{(\mathrm{L}(\mathrm{x}+1,\mathrm{y}) - L(x-1,y))}\right) \tag{3.8}$$

Poté je zkonstruován histogram orientací, který je tvořen z orientací gradientů v okolí významného bodu. Histogram obsahuje 36 sloupců, kde každý sloupec značí 10 stupňový rozsah. Každý vzorek, který je přidán do histogramu, je vážen velikostí gradientu a koeficienty Gaussova kruhového okna. Gaussovo okno má střed ve významném bodu a její měřítko (σ) odpovídá 1.5násobku měřítka významného bodu. Vrcholy v histogramu odpovídají dominantním orientacím gradientů v okolí bodu. Dominantní orientace významného bodu, je určena nejvyšším sloupcem v histogramu. Pokud se v histogramu vyskytnou sloupce, které dosahují alespoň 80 % hodnoty nejvyššího sloupce, je vytvořen nový významný bod. Tento významný bod má stejnou lokaci a meřítko jako původní významný bod, ale jinou orientaci. Tím pádem mohou vzniknout významné body, které mají přiřazeno více orientací.



Obrázek 3.7.6: Ilustrace velikostí gradientů a jejich orientace v Gaussově rozostřeném obrázku[7]

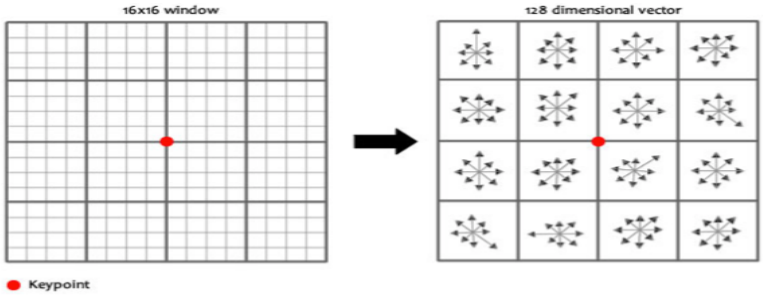
[7] http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/



Obrázek 3.7.7: Ukázka histogramu orientací[8].

Vytvoření deskriptorů pro významné body

V minulém kroku metody byla významným bodům přiřazená dominantní orientace. Dominantní orientace byla dána konstrukcí histogramu orientací v okolí významných bodů. Cíl tohoto kroku je vytvořit deskriptor, který bude nenáročný na výpočet. Deskriptor by mněl popsat okolí významných bodů, tak aby byl nezávislý na geometrických a jasových transformacích obrazu. Deskriptor se tedy vytvoří z velikostí gradientů a jejich orientací v okolí významného bodu. Nejprve je nalezen obraz L (x, y, σ), který je měřítkově nejbližší k významnému bodu, aby byla opět zachována nezávislost vůči měřítku. Šestnáct čtvercových oken o velikosti 4*4 v okolí bodu, bude použito pro výpočet. Pro každé okno bude sestaven histogram orientací, který bude mít 8 sloupců. Velikost gradientu každého bodu je vážena Gaussovou funkcí. Což má za důsledek to, že vzdálenější gradienty od středu deskriptoru budou mít menší vliv na celý deskriptor. Histogramy jsou poté natočeny vůči určené dominantní orientaci významného bodu. Tímto krokem je zachována nezávislost na rotaci. Výsledný deskriptor se tedy skládá z jednotlivých histogramů orientací. Těchto histogramů je celkem 16 a každý obsahuje 8 čísel. Těchto 128 čísel je poté normalizováno a vzniká 128 dimenzionální SIFT vektor.



Obrázek 3.7.8 : Výsledný deskriptor metody SIFT[9]

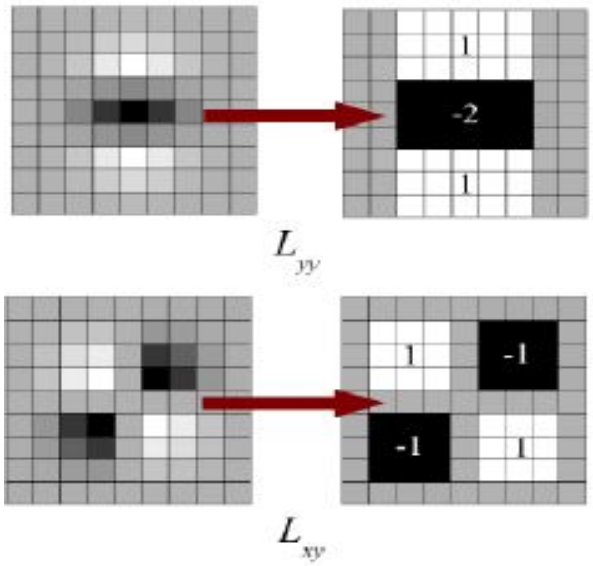
[8] http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/
[9] http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/

3.8 SURF vlastnosti

Metoda SURF (Speeded-Up Robust Feature), byla představena roku 2006, třemi lidmi, Herbertem Bayem ,Tinne Tuytelaarsem a Luc Van Goolem. Jak název napovídá, autoři se inspirovali metodou SIFT kterou chtěli zrychlit. Standartní verze SURF je podle autorů rychlejší a více robustní vůči různým transformacím.

Detekce významných bodů

SIFT metoda detekuje významné body pomocí rozdílů Gaussových funkcí. Poté je ale potřeba některé nestabilní body odstranit z množiny významných bodů. Což je realizováno pomocí Hessianu. V metodě SURF jsou tyto dva kroky sloučeny a významné body jsou detekovány přímo, použitím determinantu Hessianovy matice. Metoda pro výpočet prvků hesianovy matice využívá filtry o velikosti 9x9. Tyto filtry aproximují druhou derivaci v diskrétní formě. Prvky jsou tedy vypočítané jako konvoluce obrazu s filtry. Výhoda tohoto přístupu je, že konvoluce s filtrem se snadno počítá s pomocí integrálních obrázků. Lze ji také počítat paralelně pro různá měřítka.

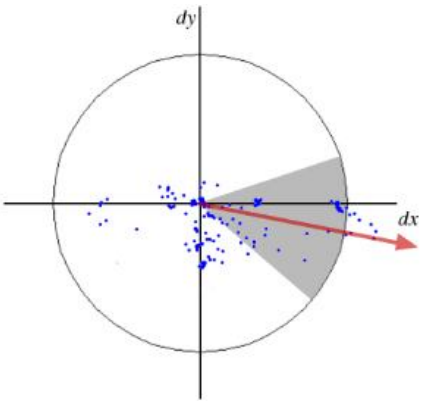


Obrázek 3.8.1: Ukázka některých filtrů[10]

Přiřazení orientace

Pro přiřazení orientace využívá vlnkové odezvy v horizontálním a vertikálním směru v kruhovém okolí. Toto okolí má poloměr 6s, kde s značí měřítko vrstvy ve scale-space, ve kterém se zkoumaný bod nachází. Tyto odezvy jsou stejně jako u SIFT, váženy koeficienty Gaussova kruhového okna s parametrem σ=2,5s. Poté jsou zakresleny do souřadnicového systému. Výsledná dominantní orientace bodu je poté určena, jako součet všech odezev v rámci vyseče o velikosti 60 stupňů.

[10] http://opencv-python-tutroals.readthedocs.io



Obrázek 3.8.1: Určení dominantní orientace SURF metodou[11]

Sestavení deskriptorů

Pro sestavení deskriptorů používá SURF vlnkové odezvy v horizontálním a vertikálním směru. Pro výpočet je využito čtvercové okolí s hranou délky 20s (s je meřítko ve scale-space). Okolí je potom rozděleno na 4x4 podoblasti. V každé podoblasti jsou vypočteny horizontální a vertikální vlnkové odezvy a výsledný deskriptor je sestrojen jako:

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{3.9}$$

Výsledný deskriptor se skládá z takto získaných vektorů ze všech 16 podoblastí. Což ve výsledku tvoří 64 dimenzionální deskriptor (16 podoblastí, 4 prvkový vektor).

3.9 FCTH a CEDD vlastnosti

FCTH a CEDD jsou globální vlastnosti, které kombinují v jednom histogramu barvu a texturní informace. Metoda FCTH vychází z kombinace tří fuzzy kroků, kde velikost jeho deskriptoru je limitována na 72 bytů na jeden obrázek. Metoda CEDD je blokový postup, který využívá rozkladu obrazu na textury a na barvy. Velikost deskriptoru CEDD je limitována na 54 bytů. Hlavní předností metody CEDD je velmi malá výpočetní náročnost. Na tuto vlastnost je kladen velmi velký důraz, pokud se hledá v obrovských databázích. Obě metody ve srovnávání vůči ostatním metodám, vykazují velmi dobré výsledky také z hlediska přesnosti vyhledávání. Detailní informace ohledně těchto metod mohou být nalezeny v[8] .

[11] http://opencv-python-tutroals.readthedocs.io

4 Porovnávání vlastností

V předchozí kapitole bylo uvedeno několik různých vlastností obrázků. Pro zjištění podobnosti dvou obrázků, je třeba tyto obrázky reprezentovat pomocí některých jejich vlastností a tyto vlastnosti potom vzájemně porovnat. V této kapitole budou představeny různé metriky pro porovnávaní vlastností obrázků, které se hojně používají v CBIR systémech.

4.1 Metriky pro porovnávání histogramů

V CBIR systémech je porovnávaní histogramů velmi častou operací. Ať už porovnávání vlastností, které jsou reprezentovány jako histogramy (histogram barev, texturové histogramy, histogramy Gáborových vlastností), nebo porovnání histogramů v systémech založených na BoVW modelu.

Minkowského vzdálenosti.

Mezi nejznámější podobnostní funkce patří Minkowského vzdálenosti. Je to soubor podobnostních funkcí definovaných jako:

$$d_p(H,K) = \left(\sum_i |h_i - k_i|^p\right)^p \tag{4.1}$$

H a K značí jednotlivé histogramy.Často používaná a nejznámější je Euklidova vzdálenost s parametrem p=2. Používané jsou také Manhattanská vzdálenost s parametrem p=1 a maximální vzdálenost s parametrem p který je roven nekonečnu.

Prusečík histogramů

Tato metoda byla navržena v 0 a byla vyvinuta přímo pro porovnávání histogramů. Zanedbává vlastnosti, které se vyskytují pouze v jednom histogramu. Je definovaná jako:

$$d(H,K) = \sum_{m=1}^{M} \min(H_m K_m) \tag{4.2}$$

X^2 vzdálenost

Používá se často v BovW modelu lze ji spočítat jako:

$$d(H,K) = \sum_{m}^{M} \frac{(H_m - K_m)^2}{(H_m + K_m)} \tag{4.3}$$

Jensen Shannon rozdílnost

Používá se pro měření rozdílu dvou pravděpodobnostních rozdělení. Je definována jako:

$$D(H,K) = \sum_{m=1}^{M} H_m \log \frac{2H_m}{H_m + K_m)} \quad K_m \log \frac{2K_m}{H_m + K_m)} \tag{4.4}$$

[2]

Earth Mover's vzdálenost

Výpočet EMD je založen na řešení Transportačního problému (tzv. Monge-Kantorovich problém).Pro porovnávání histogramů je výpočet následující:

$$d(H,K) = \frac{\sum_{ij} d_{i,j} g_{i,j}}{\sum_{ij} g_{i,j}} \tag{4.5}$$

5 Metriky Vyhledávání

Pro ohodnocení CBIR systému je potřeba mít metriky, které dokáží vyjádřit kvalitu vyhledávání. Tyto metriky vyjadřují, jak "dobře" umí systém vyhledávat. Kvalita či správnost vyhledávání, závisí na požadavcích, které jsou od systému očekávané. Vyhledávání v obrázcích může být občas velmi subjektivní. Jeden člověk může považovat dvojici obrázků za velmi podobné, přičemž jiný člověk již nemusí. Ohodnocení systému proto závisí téměř vždy na lidském vnímaní obrazu. V této kapitole budou představeny různé metriky pro hodnocení vyhledávání.

Preciznost

$$preciznost = \frac{počet\ nalezených\ relevantích\ obrázků}{celkový\ počet\ nalezených\ obrázků}$$

Preciznost (anglicky precision) udává poměr mezi počtem nalezených relevantních obrázků a celkovým počtem nalezených obrázků. Nejkvalitnější systém z pohledu preciznosti bude takový systém kdy jeho preciznost vyjde 1. Což bude znamenat že všechny nalezené obrázky budou zároveň relevantní. Naopak nejméně kvalitní bude, pokud preciznost vyjde 0.

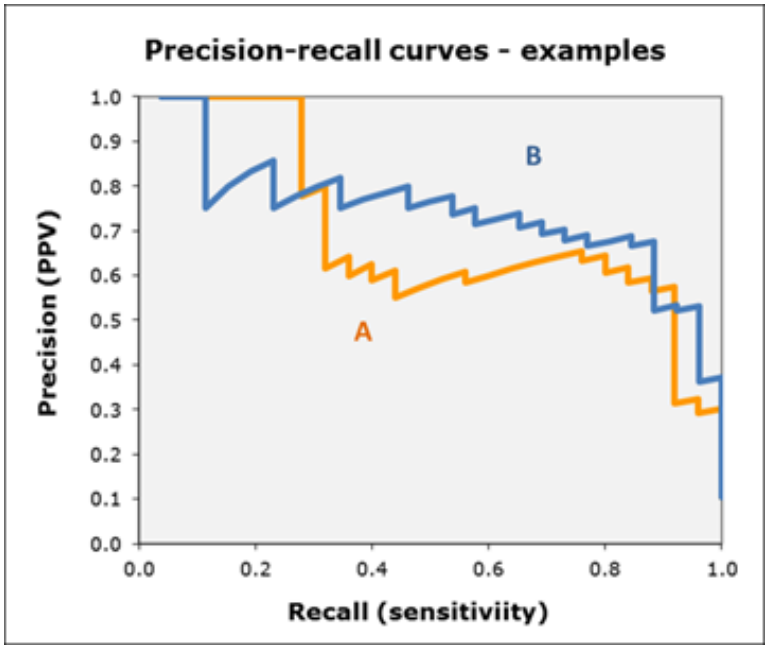
Citlivost

$$citlivost = \frac{počet\ nalezených\ relevantích\ obrázků}{celkový\ počet\ relevantních\ obrázků}$$

Citlivost (anglicky recall) je určena poměrem mezi počtem nalezených relevantních obrázků a celkovým počtem relevantních obrázků. Jako nejkvalitnější systém, bude považován systém, jehož citlivost vyjde 1.Nejméně kvalitní systém bude, pokud citlivost vyjde 0.Pokud vyjde citlivost 1, tak všechny relevantní obrázky z množiny obrázků, ve kterých se hledalo, byly nalezeny.

Precision-recall křivka

Hodnoty citlivosti a preciznosti jsou většinou reprezentované v grafu $R \rightarrow P(R)$. Tento graf znázorňuje závislost preciznosti na citlivosti.



Obrázek 5.1: Ukázka graf preciznosti a citlivosti[12]

Průměrná preciznost

Nejčastější způsob, jak reprezentovat graf preciznosti a citlivosti jako jedno číslo, je průměrná preciznost. Lze ho definovat jakou obsah plochy pod křivkou v grafu. Vypočítá se tedy jako:

$$AP = \int_0^1 p(r)dr \tag{5.1}$$

Lze také vypočítat průměr průměrné preciznosti (anglicky mean average presicion) pro Q dotazů jako:

$$AP = \frac{1}{|Q|}\sum_{q\epsilon Q} AP(q) \tag{5.2}$$

Četnost chyb (Error rate)

Error rate budeme chápat jako četnost výskytu chyb ve všech hledání. Jako správné vyhledávání, se bude chápat vyhledávaní, kde z určitého počtu všech nejpodobnějších obrázků, budou všechny relevantní. Jinak bude vyhledání označeno za chybné.

$$ER = \frac{1}{|Q|}\sum_{q\epsilon Q}\begin{cases}0 & pokud\ obrázek\ bude\ relevantní \\ 1 & jinak\end{cases} \tag{5.3}$$

Tato metrika je zajímavá v případě, pokud databáze s obrázky bude obsahovat obrázky, přiřazené do tříd (např. množina obrázků CT snímek ramena, CT snímek ramena).

[12] https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

6 Návrh řešení

Cíle této jsou vytvořit nástroj pro vyhledávání v databázi naskenovaných CT dat, a nalezení nejlepších metod pro vyhledávání CT dat. Jako vzorová data budou sloužit obrázky ve formátu DICOM. Což je formát pro ukládání medicínských dat, pořízených snímacími metodami jako jsou CT, MRI či ultrazvuk. Výstupní data z CT či MRI jsou desítky až stovky obrázků, které tvoří jeden dataset. Například výstupem CT vyšetření hlavy pacienta není jeden obrázek, ale množina obrázků (skenuje se po vrstvách). Proto je nutné porovnávat a vyhledávat celé datasety a ne jednotlivé obrázky.

6.1 Požadavky na nástroj pro vyhledávání

- **Vyhledávání podobných obrázků**

  Je hlavní požadavek pro výslednou aplikaci. Jako data budou sloužit adresáře, které obsahují sérii snímků z CT. Tedy co jeden adresář to jedna série snímků. Jak již bylo uvedeno v úvodu, s jednou sérií se bude zacházet jako s celkem. Vždy se budou porovnávat celé série. Navrhovaný nástroj by mněl být schopen vyhledávat ve velkém množství CT snímků, které jsou v uloženy v jednotlivých sériích.

  Použití tohoto nástroje by mnělo být zejména v lékařství. Kdy lékař má k dispozici velké množství CT snímků a potřebuje si najít určité CT série. Tyto série si bude hledat na základě podobnosti se vzorovou sérií CT dat. Funkcionalita vyhledání podobných snímků mu zjednoduší její procházení a výběr.

- **Vkládání, úprava a mazaní CT dat**

  Vkládání nových sérií bude probíhat tak, že do kořenového adresáře se vloží podadresář s CT daty. Mazání bude probíhat podobně, kdy se vyhledá konkrétní adresář a ten se odstraní z kořenového adresáře. Všechny CT série budou pojmenovány. Název se určí podle dat, které jsou obsahem adresáře. Typicky tedy názvy částí těla.

- **Podpora formátu DICOM**

  Jelikož většina snímků z CT, jsou ve formátu DICOM, je třeba umět načítat obrázky v tomto formátu.

- **Přehledné grafické uživatelské rozhraní**

  Aby si uživatel, který bude chtít vyhledat určitý CT dataset , nemusel zadávat všechny potřebné parametry složitě. Například cesta ke vzorovému CT datasetu by nemněla být psána, ale zadána kliknutím. Také aby výsledky z vyhledávání, byly zobrazeny přehledně a v rozumné formě.

- **Rychlé a přesné vyhledávání**

  Tyto vlastnosti budou velkou mírou určeny použitou knihovnou Lire, která je používaná v praxi i pro milionové databáze s obrázky, viz kapitola Implementace 7. Přesnost vyhledávaní bude určena hlavně správným výběrem metody pro vyhledávání v CT datech.

6.2 Požadavky pro nalezení nejlepších metod

- **Kvalitní a dostatečné množství CT dat**

  Tento požadavek bude klíčovým požadavkem pro nalezení nejlepší metody. Bez kvalitních CT dat, by nalezení nejlepších metod pro vyhledávání v CT datech nebylo možné.
- **Výběr kvalitních metrik pro porovnání jednotlivých metod**

  Jelikož jednotlivé metody fungují na rozdílných principech a vrací jiné výsledky je třeba mít dostatečně kvalitní metriky pro porovnání těchto výsledků. Metrikami pro CBIR systémy se zabývá kapitola 5.
- **Uložení výsledků experimentů do databáze**

  Pro aplikaci metrik na jednotlivé metody CBIR systému, je potřeba mít výsledky hledání uložené přehledně na jednom místě. Nejlépe v databázi, kde se budou ukládat argumenty, čas spuštění, výsledky hledání atd.
- **Funkční nástroj pro CBIR vyhledávání**

  Pro experimentální nalezení nejlepších metod je potřeba mít nástroj, kde tyto metody budou implementované. Nejprve je třeba vytvořit nástroj a až poté přijde na řadu nalezení nejlepších metod.

6.3 Návrh vyhledávání

Stěžejní funkcí navrhovaného nástroje je vyhledávání CT snímků podle podobnosti.

Princip CBIR vyhledávání lze ve zkratce shrnout tak, že pro každý obrázek je extrahován zvolenou metodou deskriptor. Extrakce deskriptoru by měla být tzv. offline operací, což znamená že se provádí pouze jednou. Pokud dojde k úpravě databáze s obrázky je nutné provést extrakci deskriptorů znovu. Při vyhledávání se pak porovnávají deskriptory dotazovaného obrázku s deskriptory všech obrázků v databázi.

Extrakce deskriptorů je činnost, která zdaleka trvá nejdelší dobu. Pro rychlost systému je velmi důležité extrahované deskriptory uložit a vyhledávání provádět na již existujícími deskriptory. V Lire knihovně jsou tyto deskriptory ukládány do souborového sytému s vyžitím Lucene knihovny.

Což zaručuje požadovanou vlastnost systému.

6.4 Vyhledávání v CT sériích

Navrhl jsem následující postup. Z CT série, která bude sloužit jako vzorová data, se vybere parametrizovaný počet obrázků. Každý vybraný obrázek bude porovnáván danou metodou, vůči všem obrázkům v databázi. Poté se vybere parametrizovaný počet nejlepších shod a zjistí se, do které CT série obrázky patří. Pak se podle míry shody, stanový procentuální zastoupení příslušnosti do jednotlivých CT sérií. Například pro první obrázek ze vzorové CT série vyjde 35% příslušnost dataset1, 60% příslušnost dataset2 a 5% příslušnost dataset3.Tento postup se provede pro každý vybraný obrázek a pak se spočítá procentuální příslušnost celé vzorové CT série. Spočítá se jako aritmetický průměr procentuálních příslušností, jednotlivých obrázků.

6.5 Použité data

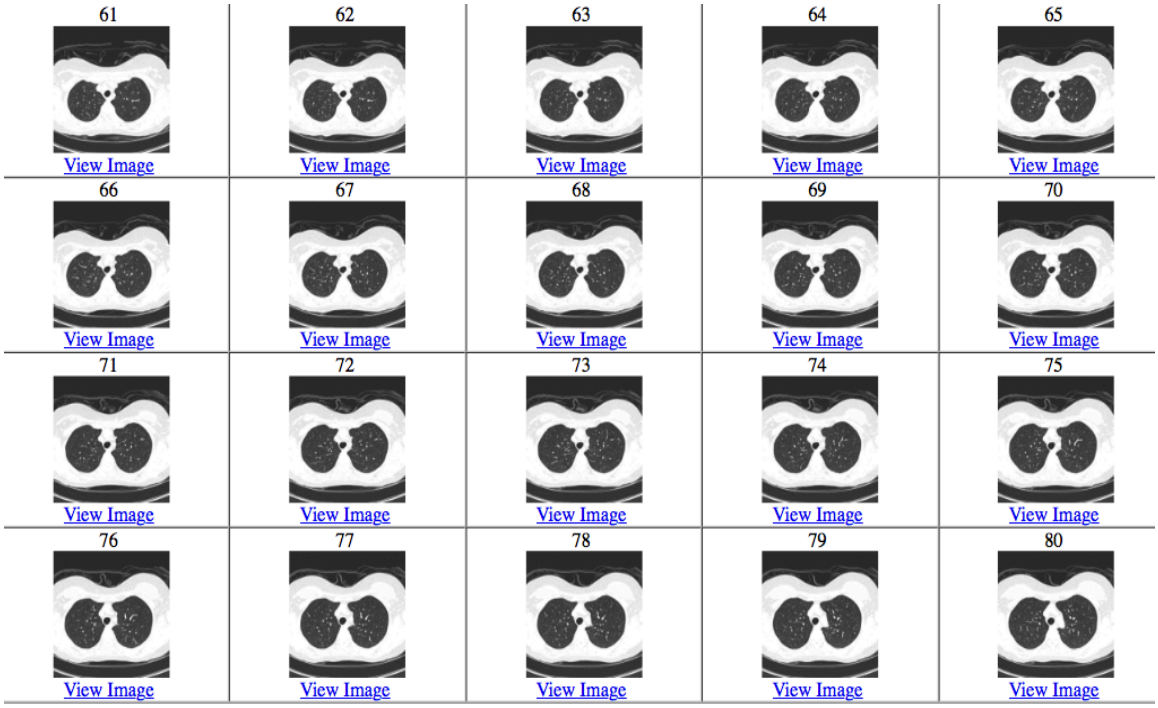
Jako data budou sloužit CT snímky pacientů s rakovinou, které jsou k dispozici v webovém archivu[13].Tyto data jsou ve formátu DICOM a počet obrázků v jednotlivých sérií se pohybuje 100 do 400 obrázků. Velikost jedné série se pohybuje od 100 MB do 200 MB. Třídy jednotlivých obrázků jsou např. snímky močové měchýře, snímky plic atd. Ukázka série CT snímků hrudníku je na obrázku 6.1.

6.6 Vyhodnocení metod

Budou se vyhodnocovat metody: Gabor, Tamura, CEDD, FCTH, SIFT, SURF a korelogram barev.
Z výsledků hledání jednotlivých metod se vytvoří graf preciznosti a citlivosti, vypočítá se průměrná preciznost, průměr průměrné preciznosti, a četnost chyb. Těmito metrikami se zabývá kapitola 5.

Výsledky hledání jednotlivých metod budou uloženy v SQL databázi. Pro aplikaci metrik na jednotlivé metody je třeba určit které obrázky jsou relevantní a které ne. Relevantnost obrázků bude určena třídou, do které patří. Tzn. pokud vyhledaný obrázek bude patřit do stejné třídy jako vzorový bude brán jako relevantní. Tyto třídy budou určeny názvem adresáře CT série. Například “Ramenní kloub“, “Hrudník“.

[13] http://www.cancerimagingarchive.net



Obrázek 6.1: Série CT snímků hrudníku.

7 Implementace

Tato kapitola popisuje implementační detaily řešení, které bylo popsáno v minulé kapitole.
Realizace řešení se skládá z jednoduchého uživatelského rozhraní, řízení volání API vyhledávacího enginu LIRE, zpracovaní a analýza výsledků, uložení výsledků do databáze, konzolové a dávkové volaní.

7.1 Technologie

Jako implementační jazyk jsem zvolil jazyk Java nejméně ve verzi JAVA 8. Jazyk byl dá se říci určen výběrem enginu LIRE, který je implementovaná v jazyce Java. Aplikace má klasickou MVC architekturu. Kde model je reprezentován souborovým systémem pro vstupní lékařská obrázková CT data. Pro uložení výsledných dat z vyhledání jsem zvolil relační SQL databázi. Konkrétně jsem použil MySql databázi. Pro práci s databází jsem nevyužil žádného ORM frameworku.

Operace nad databází se dějí dle standartního JDBC API. Důvodem nevyužití frameworku je, že databáze je zatím využita jen jako evidence vyhledaných dat. Současná aplikace tato data dále nijak neanalyzuje a nezobrazuje. Kontroler je řešen pomocí Java tříd spojující volání z GUI na API vyhledávajícího enginu LIRE. Uživatelské rozhraní je realizováno pomocí knihoven swing, awt. Pro dávkové testování jsem použil linux shell skripty. Pro kompilaci je využit nástroj pro automatizaci sestavování programů Gradle.

7.2 Knihovny a nástroje třetích stran

Nejdůležitější použitá knihovna v této práci je knihovna LIRE. Je to knihovna, která se používá pro CBIR vyhledávání. Implementuje extrakci většiny vlastností obrázků, které jsou momentálně známé. Využívá také další knihovny jako jsou Apache Lucene či OpenCV. LIRE knihovnu používá Dánská Národní policie pro hledání podobných scén, či pro detekování duplikátů. Používá ji taky organizace WIPO pro hledání v milionech obrázcích. Dále byla použita knihovna ImageJ, tato knihovna byla použita pro načtení DICOM obrázků. O práci nad databází se stará JDBC API.

7.3 Důležité třídy

Obsahem této podkapitoly bude přehled základních tříd, které navržený nástroj využívá.

Indexer

Tato třída řeší volání LIRE operací pro indexovaní vstupní množiny CT dat. Nejdůležitější metodou třídy je metoda public static void index (String pathString, Class featureClazz). Metoda rekursivně projde obrázkové soubory z cesty pathString a indexuje je pomocí feature Class.

Searcher

Tato třída řeší volání LIRE operací pro vyhledání kandidátních obrázků. Nejdůležitější metodou třídy je metoda public static void search(String pathString, int pocetHledanych, int pocetZpracovanychHitu, Class featureClazz). Metoda rekursivně projde obrázkové soubory z cesty pathString. Pokud není vstupem soubor, ale adresář je hledána množina obrázků vyskytujících se v adresáři. Pokud je vstupem dataset obrázků, parametr početHledaných určuje velikost množiny hledaných obrázků. Parametr pocetZpracovanychHitu řeší, kolik nejpodobnějších obrázků bude nalezeno a zpracováno. Parametr featureClazz určuje, jakou metodou se bude obrázek vyhledávat.

Gui

V této třídě je implementováno grafické uživatelské rozhraní.

ImageUtil

Tato třída se stará o operace s obrázky. Např. převod DICOM formátu na png. Uložení obrázků na souborový systém.

FileUtil

Servisní třída pro práci se souborovým systémem.

DbStorage

Třída pro práci s databazí.

8 Dosažené výsledky

Tato kapitola shrne výsledky, které byly dosažené při tvorbě nástroje pro vyhledávání v CT datech a výsledky porovnání jednotlivých metod. Informuje také o způsobu vyhodnocení těchto výsledků

8.1 Nástroj pro vyhledávání v CT datech

Nástroj vyhledává celé CT série, na principu, který jsem navrhl v podkapitole 6.4. Implementační detaily jsou popsány v kapitole 7.

8.1.1 Grafické uživatelské rozhraní

Ukázka grafického uživatelského rozhraní nástroje je na obrázku 8.1.

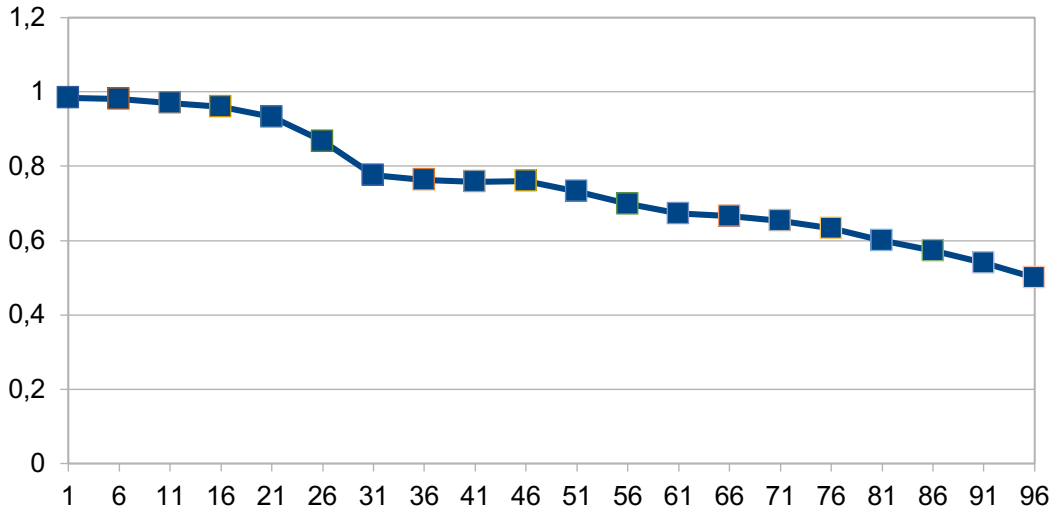
8.1.2 Experimenty pro nalezení optimální konfigurace nástroje

Použitá data
Vyhledávalo se v CT sériích snímků, získané z webového archivu viz podkapitola 6.5 Jednotlivé údaje jsou zobrazeny v tabulce 8.1.

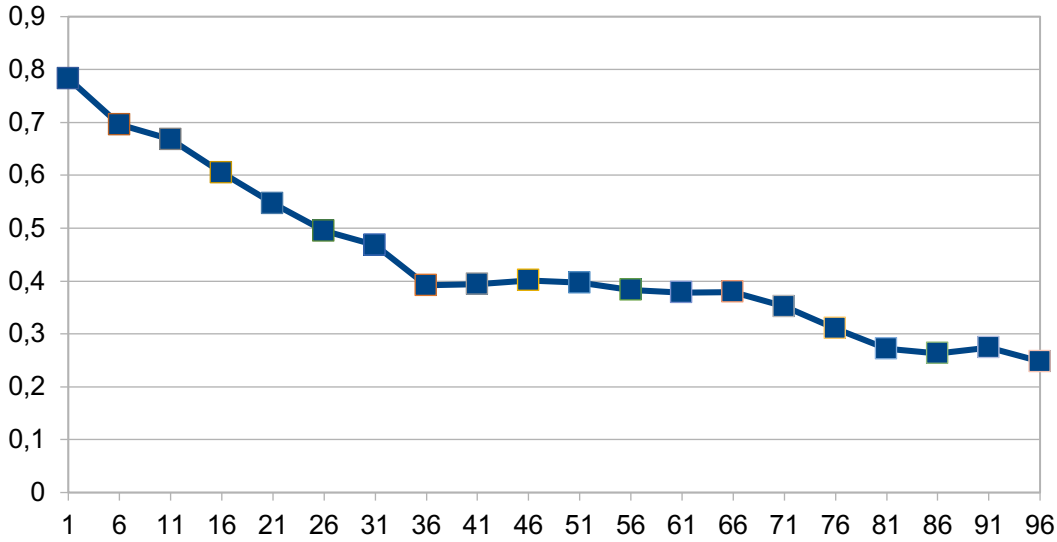
Počet CT sérií	Celkový počet obrázků	Počet tříd
40	6345	10

Tabulka 8.1: Údaje o databázi CT dat.

Pro správné nastavení nástroje bylo nutné rozhodnout, pro jaké defaultní parametry bude hledání v CT sériích realizováno. Princip hledání v CT sériích je popsán v podkapitole 6.4. Pro tento účel proběhlo několik sérií testů vyhledaní. Graf 8.1 a 8.2 zobrazuje závislost pravděpodobnosti nalezení relevantního CT datasetu, na počtu nalezených nejpodobnějších obrázků.

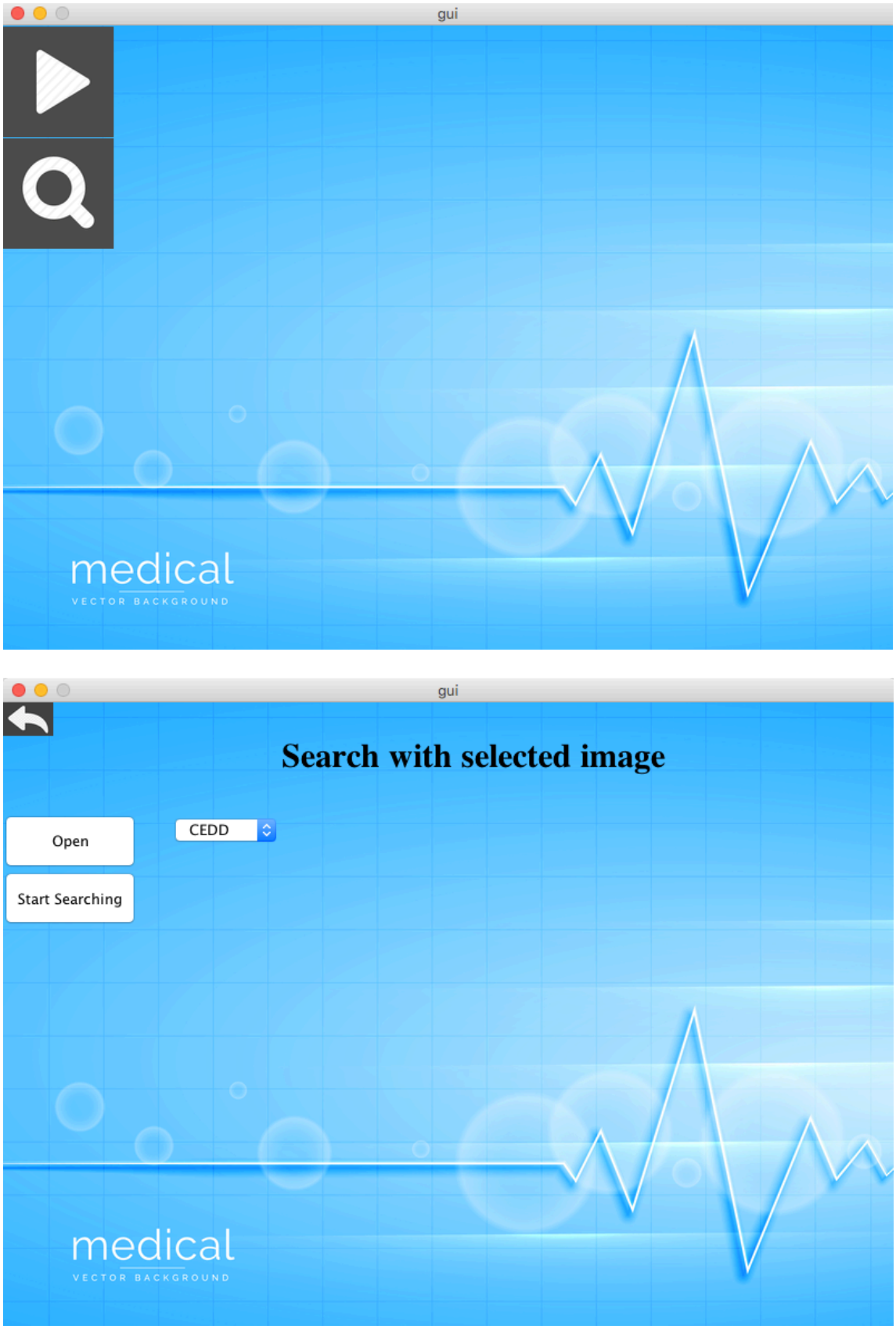


Graf 8.1: Graf závislosti pravděpodobnosti nalezení relevantního CT datasetu, na počtu nalezených nejpodobnějších obrázků, pomocí metody CEDD.



Graf 8.2: Graf závislosti pravděpodobnosti nalezení relevantního CT datasetu, na počtu nalezených nejpodobnějších obrázků, pomocí metody Tamura.

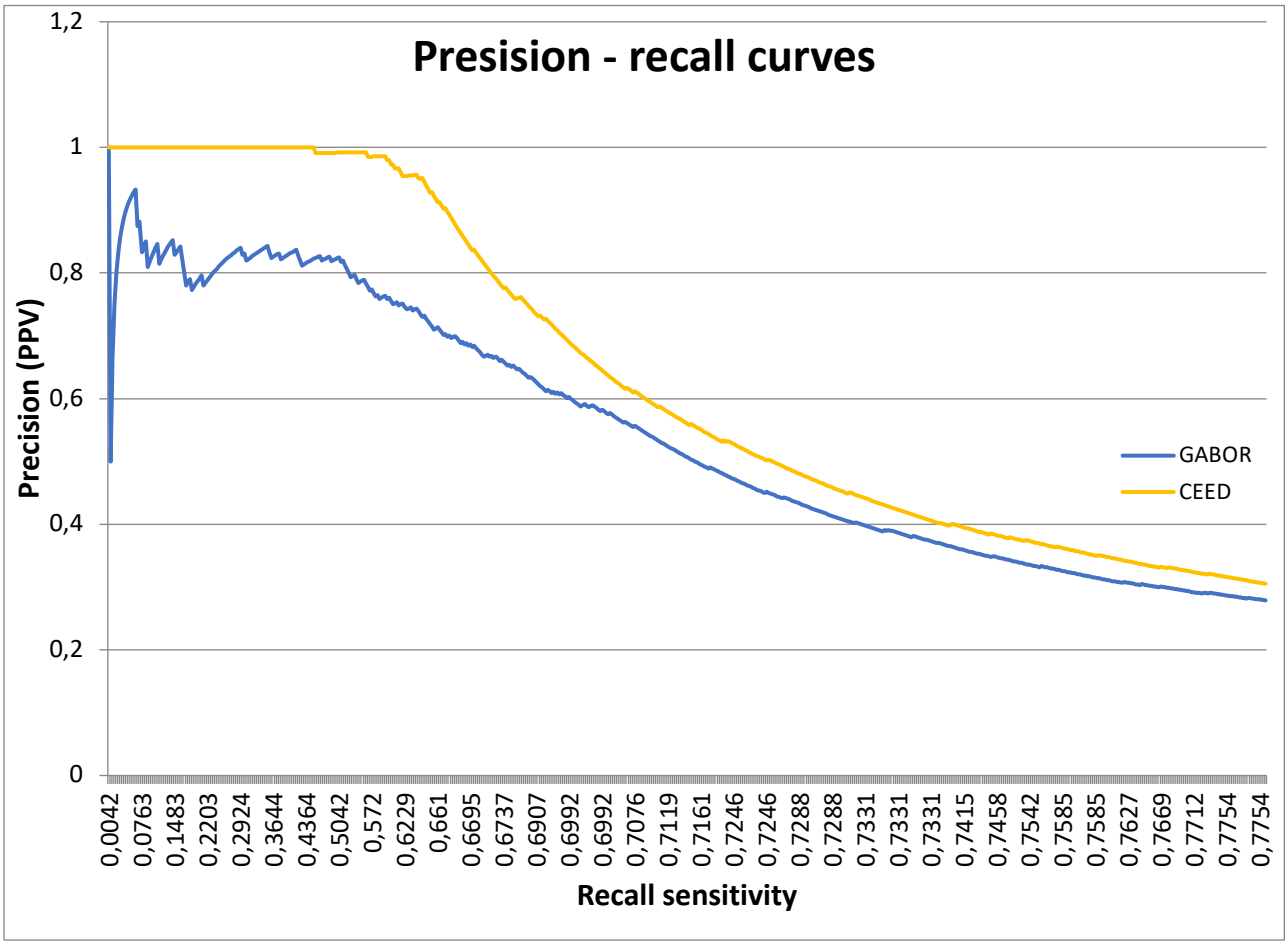
Pro nástroj se jako optimální parametry jeví vyhledání v počtu obrázků, který je větší než třetina celkového počtu zvolené série, se zpracováním 10 nejpodobnějších obrázků. Testovaní proběhlo automatizovaně a přesnost vyhledání správného datasetu, odpovídá přesnosti vyhledání jednotlivých metod.



Obrázky 8.1: Ukázka vytvořeného nástroje pro vyhledávání v CT datech.

8.2 Výsledky vyhodnocení metod

Vyhodnocení metod proběhlo na principu, kterým je vysvětlen v podkapitole 6.6.Vyhledávalo se ve stejných datech, které byly použity u při experimentech pro nalezení optimální konfigurace nástroje. Pro zvýšení přesnosti při ohodnocení metod se počítal *mean average precision* (map*)*. Což znamená, že bylo vypočteno více *precision recall křivek*, pro různé dotazy. *Precision recall* křivky byly počítány pro citlivost v intervalu (0,0.5). Poté z těchto křivek byl vypočten map. Ukázka *precision recall* křivek, které byly získané z vyhledávaní pomocí metod Gábor a CEDD, je na grafu 8.3. Hodnoty *map* pro jednotlivé metody jsou uvedeny v tabulce 8.2.



Graf 8.3: Ukázka *precision recall křivek* metod Gábor a CEDD .

vlastnost	mean average precision (map)
Gábor	63.3
Tamura	56.5
SIFT	42.9
SURF	19.4
CEDD	92.8
FCTH	45.5
korelogram barev	22.4

Tabulka 8.2 : Map v [%] pro jednotlivé vlastnosti (citlivost v intervalu (0,0.5)).

Zhodnocení výsledků a doporučení vhodné metody

Jednoznačně nejlepší výsledky vykazovala metoda CEDD, mohlo to být částečně dáno tím, že tato metoda bere v potaz i barvy v obrázku, které jsou v případě CT dat poměrně neměnné. Dobré výsledky také vykazovaly metody Gábor, Tamura, FCTH a SIFT. Takže 4 z 5 nejlepších metod, jsou metody, které berou v potaz texturu obrázku. Což může znamenat, že textura je poměrně důležitá vlastnost při ohodnocování CT dat.

Naopak nejhorší výsledky vykazovaly metody, SURF a korelogram barev. Korelogram barev nemněl dobré předpoklady pro dobré výsledky, jelikož bere v úvahu pouze barvy obrázku. Autoři metody SURF tvrdily, že metoda má lepší výsledky pro vyhledávání než metoda SIFT. Toto tvrzení nemohu potvrdit, jelikož výsledky pro vyhledávání v CT datech vyšly přesně opačné. Pro vyhledávání v CT datech, tedy lze doporučit metodu CEDD. Tato metoda dosáhla kvalitních výsledků a má velmi malou výpočetní náročnost.

Pro získání výsledků uvedených v tabulce 8.1 bylo provedeno přes 150 tisíc hledání a bylo nalezeno více jak 2.2 milionu obrázků.

9 Závěr

Výsledkem práce je v Javě implementovaný nástroj pro vyhledávaní podobných CT dat. Byl navržen postup pro vyhledávání v celých sérií CT dat, který vykazoval poměrně dobré výsledky. Byly také experimentálně porovnány jednotlivé metody a na základě výsledků, doporučeny nejvhodnější metody pro vyhledávaní v CT datech. Z hlediska dalšího vývoje projektu navrhuji experimentování s větším množstvím metod pro vyhledávání a jejich kombinacemi. Tímto přístupem by bylo možné nalezení dokonalejších metod pro vyhledávání, které by vykazovaly lepší výsledky. Také by bylo dobré testovat metody na daleko větším počtu dat, což by přineslo lepší přesnost z hlediska porovnávání Mohlo by být také navrhnuty nové postupy pro vyhledávání v CT sériích a ty pak porovnány.

Návrh a implementace vyžadovaly nastudování a pochopení metod využívaných pro porovnaní a určení podobnosti CT dat. Bylo také nutné nastudovat a aplikovat metriky vhodné pro porovnání vyhledávacích metod. Jelikož se pracovalo s CT daty, bylo také třeba se seznámit s jejich reprezentací.

Výsledný nástroj může být využit všude tam, kde je uloženo v databázi velké množství CT dat a je třeba v nich vyhledávat podle podobnosti.

10 Literatura

[1] Výpočetní tomografie [online], poslední aktualizace 10. 3. 2018 v 11:57 [cit. 1. 5. 2018], Wikipedie. Dostupné z: https://cs.wikipedia.org/wiki/Výpočetn%C3%AD_tomografie

[2] Thomas Deselaers, Daniel Keysers, Hermann Ney Features for Image Retrieval: An Experimental Comparison. Information Retrieval. 2008.
Dostupné z: http://thomas.deselaers.de/publications/papers/deselaers_diploma03.pdf

[3] O. Kucuktunc, D. Zamalieva, "Fuzzy color histogram-based CBIR system", Proceedings of 1st International Fuzzy Systems Symposium, 2009
Dostupné z: https://pdfs.semanticscholar.org/2cd6/d00212bb02b75a20f4967f86e373c1ff2da8.pdf

[4] [Tamura & Mori+ 78] H. Tamura, S. Mori, T. Yamawaki. Textural Features Corresponding to Visual Perception. IEEE Transaction on Systems, Man, and Cybernetcs, Vol. SMC-8, No. 6, pp. 460–472, June 1978.
Dostupné z:https://www.semanticscholar.org/paper/Textural-Features-Corresponding-to-Visual-Tamura-Mori/597edc3174dc9f26badd67c4e81d0e8a58f9dbb3

[5] J. Ilonen, J.-K. Kamarainen, and H. Kalviainen, “Fast extraction of multi-resolution gabor features,” in 14th Int Conf on Image Analysis and Processing (ICIAP), 2007, pp. 481–486.
Dostupné z: https://www2.it.lut.fi/project/simplegabor/downloads/laitosrap100.pdf

[6] Lowe, DG. 2004. Distinctive image features from scale-invariant features. International Journal of Computer Vision, 60(2): 91–110.
Dostupné z: https://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/lowe_ijcv2004.pdf

[7] M. J. Swain, D. H. Ballard. Color Indexing. International Journal of Computer Vision, Vol. 7, No. 1, pp. 11–32, Nov. 1991

[8] Kumar.P, Compact Deskriptors for Accurate Image Indexing And Retrival: FCTH and CEDD. *International Journal of Engineering Research and Technology* **2012**, *1*,
Dostupné z: https://www.ijert.org/download/1334/compact-descriptors-for-accurate-image-indexing-and-retrieval-fcth-and-cedd.