

UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN POSLEDIPLOMSKIH STUDIJA

DOKTORSKA DISERTACIJA

DIZAJN I IMPLEMENTACIJA TEHNIKE KOMBINOVANOG
PRETRAŽIVANJA TEKSTA I SLIKE U MULTIMODALNIM
ELEKTRONSKIM POSLOVNIM DOKUMENTIMA

Mentor

Prof. Dr Ranko Popović

Kandidat

Mr Dragan Marković, dipl. inž.

BEOGRAD, 2009. godina

ZAHVALNICA

U izradi ovog rada uloženi su napor i vreme za koje je uskraćena moja porodica. Zato se u prvom redu zahvaljujem supruzi *Vesni* i kćerkama *Ivi* i *Ani* za opštu podršku i razmevanje koje su mi pružili u periodu rada na disertaciji.

Posebno se zahvaljujem članovima komisije, mentoru *Prof. Dr Ranku Popoviću*, profesoru Univerziteta Singidunum na inicijativi, punoj podršci, datim sugestijama i ukupnom pozitivnom pristupu i ličnom doprinosu u definisanju forme i sadržaja rada u pripremi, realizaciji i odbrani i *Prof. Dr Milanu Milosavljeviću*, prorektoru Univerziteta Singidunum na razumevanju potrebe, odobravanju teme, stvaranju opštih uslova za realizaciju disertacije i ukupnom angažovanju u radu ispitne komisije. Izuzetno se zahvaljujem *Prof. Dr Branislavu Kovačeviću*, rektoru Beogradskog univerziteta na angažovanju u radu ispitne komisije, iznetim sugestijama i velikom doprinosu da rad dobije formu i kvalitet na zahtevanom nivou.

SADRŽAJ

Apstrakt	5
Abstract	5
1. Uvod	7
1.1 Predmet istraživanja	7
1.2 Pregled vladajućih stavova i shvatanja u području istraživanja	7
1.3 Cilj istraživanja	8
1.4 Metode istraživanja	8
1.5 Očekivani rezultati	9
1.6 Struktura rada	9
2. Problem istraživanja	11
3. Kombinovano pronalaženje teksta i slike – presek stanja	17
3.1 Dokument i upit	17
3.2 Obeležja i pronalaženje	18
3.3 Metodologija pronalaženja	20
3.4 Multimodalni dokumenti	20
3.5 Tehnike indeksiranja	22
3.5.1 Indeksiranje tekstualnog sadržaja	22
3.5.2 Indeksiranje vizuelnog sadržaja	24
3.6 Modeli pronalaženja	25
3.6.2 Modeli pronalaženja slika na osnovu sadržaja	27
3.7 Kombinacija teksta i slika	28
3.7.1 Prednosti i nedostaci modaliteta teksta i slika	29
3.8 Pregled multimodalnih IR sistema	30
3.9 Tehnike kombinovanja	32
3.9.1 Sekvencijalno korišćenje modaliteta teksta i slike	32
3.9.2 Simultano korišćenje oba modaliteta	33
3.10 Tehnike za poboljšanje efektivnosti IR	34
3.10.1 Latentno semantičko indeksiranje	34
3.10.2 Semantičke mreže	36
3.11 Korisnikova povratna informacija o značajnosti	37
3.11.1 Uvod u povratnu informaciju o značajnosti	37
3.11.2 Povratna informacija o značajnosti kod multimodalnog pronalaženja	38
3.12 Grupisanje i klasifikacija dokumenata	38
3.12.1 Uvod u klasifikaciju i grupisanje dokumenata	38

3.12.2 Multimodalna klasifikacija	39
3.12.3 Multimodalno grupisanje	39
3.13 Rezime	40
4. Relevantnost	41
4.1 Osvrt na početak	41
4.2 Principi i teorijske osnove	42
4.3 Informacijski procesi i sistemi za obradu	45
4.4 Definicija relevantnosti	46
4.5 Prevazilaženje problema	47
5. Funkcionisanje mašine za pretraživanje	49
5.1 Procesiranje dokumenta	49
5.2 Ukratko o leksikografiji	51
5.4 Indeksiranje podataka	55
5.5 Obrada upita	57
5.6 Rangiranje relevantnih rezultata	58
5.7 Poređenje svojstava biblioteka za pronalaženje informacija	60
6. Dizajniranje aplikacije	63
6.1 Softverska biblioteka mašine za pretraživanje – Lucene	63
6.1.1 Funkcionalnost biblioteke Lucene	63
6.1.2 Indeksiranje	64
6.1.3 Pretraživanje	66
6.1.4 Analiza	67
6.1.5 Jednačina ponderisanja biblioteke Lucene	69
6.1.6 Parseri dokumenata	69
6.2 Procesiranje tekstualnih dokumenata	70
6.2.1 Razvoj elektronskog zapisa teksta	70
6.3 Formatii poslovnih dokumenata	76
6.3.1 MS Word	76
6.3.2 PDF	78
6.3.3 RTF	80
6.4 Procesiranje slika	80
6.4.1 Boja i atributi boje	80
6.4.2 Spektar vidljivog svetla	82
6.4.3 Kamere i oči	82
6.4.4 Primarne boje	87
6.4.5 Komplementarne boje	88
6.5 Prostor boja	90
6.5.1 RGB prostor boja	90
6.5.2 HSV prostor boja	91
6.6 Slike i stvaranje digitalnih slika	92
6.6.2 Reprezentovanje slike	93
6.7 Obeležja slike	97
6.8 Presek stanja CBIR sistema	98
6.9 Histogrami digitalnih slika	99
6.9.1 Primeri histograma	101
6.9.2 RGB histogram i histogrami po bojama	103
6.9.3 Histogram osvetljenja	105

6.9.4 Globalni histogrami boje	106
6.9.5 Poređenje histograma	106
6.9.6 Lokalni histogrami boja	107
6.9.7 Problemi kod histograma boja	108
6.10 Obeležja Tamura	108
6.11 Obeležje globalne teksture	108
6.12 Gabor obeležje	109
6.13 Histogrami invarijantnog obeležja	109
6.14 Lokalna obeležja slike	109
6.15 MPEG-7	112
6.15.1 Izbor obeležja/deksriptora	114
6.15.2 MPEG-7 deskriptor SCD	118
6.15.3 MPEG-7 deskriptor EHD	120
6.15.4 Povezivanje deskriptora SCD i EHD sa Lucene bibliotekom	122
6.16 Projektni zahtevi	122
7. Implementacija, eksperimenti i rezultati	123
7.1 Implementacija	123
7.2 Eksperimenti i rezultati	134
8. Zaključak	145
9. Literatura	147
10. Skraćenice	157

Apstrakt

Predmet rada je proučavanje pronalaženja informacija u multomodalnim poslovnim dokumentima u desktop okruženju.

Dokument prenosi informaciju koristeći više modaliteta, uključujući tekst, raspored/stil i slike. Na primer, članci u časopisima obično imaju slike koje ilustruju eksperimentalne rezultate a naslovi obično imaju veću veličinu fonta od tela teksta. Indeksiranje i pronalaženje samo pomoću teksta je tradicionalna metoda u sistemima za pronalaženje informacija. Širenjem Interneta i pojavom digitalnih biblioteka nametnula se potreba za razvojem tehnika za inteligentno indeksiranje i pronalaženje multimodalnih dokumenata.

Indeksiranje i pronalaženje su dve važne komponente sistema za pronalaženje informacija. Za datu kolekciju dokumenata, indeksiranje opisuje dokumente koristeći indeksni jezik. Pronalaženje koristi rezultate indeksiranja i nalazi dokumenta koja odgovaraju korisnikovom upitu. Modaliteti teksta i slike koriste različite tehnike indeksiranja i pronalaženja. Jednomodalni sistemi za pronalaženje informacija, koji koriste ili samo tekst ili samo sliku, imaju ograničenja. Multimodalni sistemi imaju za cilj da prevaziđu ograničenja svakog od modaliteta tako što ih kombinuju.

Ključne reči: multimodalno pronalaženje informacija, multimodalni dokumenti, CBIR, tekstualno pretraživanje, MPEG7.

Abstract

This paper deals with exploring information retrieval of multimodal business documents in desktop environment.

A document conveys information using multiple modalities, including text, layout/style and images. For example, journal articles usually have figures to illustrate experimental results, and the title in a journal article usually has a different font size than the body text. Indexing and retrieval using only text is the traditional way of IR (Information Retrieval). With the development of the Internet and Digital Libraries, it becomes increasingly important to develop IR techniques for intelligent indexing and retrieval of multimodal documents.

Indexing and retrieval are two important components of an IR system. Given a collection of documents, indexing describes documents using an index language. Retrieval uses the results of indexing and finds related documents corresponding to a user's query. Text and image modalities use different indexing and retrieval techniques. Single-modality IR, either using text or images, has limitations. Multimodal IR aims to overcome the limitations in each single modality by combining them.

Keywords: multimodal Information Retrieval, multimodal documents, CBIR, text retrieval, MPEG7.

1. Uvod

1.1 Predmet istraživanja

Potreba za boljim aplikacijama za pretraživanje informacija postaje sve izraženija i urgentnija zbog naglog rasta informacija u digitalnom obliku. Novi sadržaji se kreiraju svakodnevno a stari sadržaji se retko brišu. Posledica toga je akumuliranje tokom vremena ogromnih količina informacija tako da nije neobično da pojedinci imaju uskladišten sadržaj od par desetina gigabajta a firme srednje veličine od nekoliko terabajta.

Aplikacije za pretraživanje treba ne samo da pronađu korisne informacije u masovnim kolekcijama, već i da obave pretraživanje u prihvatljivom vremenskom intervalu. Većina korisnika veća naviknuta je da brzo i lako pronađe informaciju zahvaljujući moćnim mašinama za pretraživanje. Međutim, to nije slučaj u tzv. desktop okruženju i okruženju relativno malih intraneta.

Hardver visokih performansi nije sam po sebi rešenje tog problema. Potrebno je stvoriti aplikaciju za pretraživanje koja će pretraživati, predlagati, vizuelizovati i kombinovati rezultate iz više izvora informacija kako bi korisnik što pre i na što lakši način dobio kvalitetnu informaciju.

1.2 Pregled vladajućih stavova i shvatanja u području istraživanja

Pretraživanje informacija se može definisati u užem i širem smislu. U širem smislu, pretraživanje informacija se definiše kao „polje koje se bavi strukturom, analizom, organizacijom, smeštajem, traženjem i pretraživanjem informacija”. U užem smislu, pretraživanje informacija se odnosi na *ad hoc pretraživanje*, što je zadatak „traženja u kolekciji dokumenata relevantnih dokumenata koji sadrže potrebnu informaciju”. Kod *ad hoc pretraživanja* nije na raspolaganju apriorno znanje o relevantnosti dokumenata. Sistem može samo da koristi opisni informacioni zahtev, koji se često naziva *upit*. Tipični primeri *ad hoc* sistema pretraživanja su mašine za pretraživanje na webu, npr. Google (www.google.com).

Indeksiranje i pretraživanje su dve važne komponente sistema za pretraživanje informacija. Za datu kolekciju dokumenata, indeksiranje opisuje dokumente pomoću tzv. Indeksnog jezika. Pretraživanje koristi rezultate indeksiranja i pronalazi dokumente koji odgovaraju korisnikovom upitu.

Dokument isporučuje informaciju koristeći više modaliteta, uključujući tekst, raspored/stil i slike. Model multimodalnog dokumenta se definiše kao skup više objekata koji mogu biti tekstualni i slikovni. Tekstualni objekat predstavlja blok teksta. Slikovni objekat predstavlja sliku/grafiku.

Standardno, jedan modalitet, bilo tekst ili slika, koristi se za pretraživanje sadržaja u multimodalnim dokumentima.

Sistemi multimodalnog pretraživanja informacija koriste i tekstualne i slikovne objekte. Pretraživanje informacija koje koristi ili tekstualni ili slikovni modalitet ima i prednosti i nedostatke. Sistemi multimodalnog pretraživanja informacija se zasnivaju na jednomodalnim sistemima pretraživanja informacija i mnogo su složeniji od njih. Mogu da se koriste kombinovani upiti koji uključuju i tekst i slike. Oba sadržaja, i tekstualni i vizuelni su indeksirani i različite tehnike kombinovanja teksta i slika su razvijene.

Multimodalno pretraživanje informacija je ogromna istraživačka oblast. Blisko je povezana sa sledećim istraživačkim oblastima: pretraživanje teksta, pretraživanje slika, pretraživanje skeniranih dokumenata i klasifikacijom dokumenata. Svaka od ovih istraživačkih oblasti je prilično velika i veoma dinamična, tako da je nemoguće napraviti sveobuhvatni pregled tekućeg stanja.

1.3 Cilj istraživanja

U skladu sa do sada navedenim, **opšti cilj istraživanja** ovog rada je dizajniranje i implementacija tehnike kombinovanog indeksiranja i pretraživanja teksta i slike kod jedne kategorije elektronskih poslovnih dokumenata (tzv. Word i PDF dokumenti) u kancelarijskom poslovanju.

Naučni cilj istraživanja se može opisati kao potreba i želja za definisanjem, implementacijom i evaluacijom tehnike koja kombinuje tekst i sliku (logo) u pretraživanju elektronskih poslovnih dokumenata i sagledavanje mogućnosti njene primene na opštije slučajeve pretraživanja elektronskih dokumenata (repozitorijumi nastavnih materijala, digitalne biblioteke itd.).

Praktični cilj ovog istraživanja je dizajniranje i implementacija samostalne desktop aplikacije za Windows platformu za efikasno pretraživanje elektronskih poslovnih dokumenata u kancelarijskom poslovanju.

1.4 Metode istraživanja

Osnovu i suštinu metoda istraživanja čini sistematičan pristup definisanju i pokušaju rešavanja problema. Primenjena je kombinovana teorijsko-eksperimentalna metoda rada, tj. osnovne naučne metode primerene ovom nivou i tipu naučnog istraživanja:

- Prikupljanje i proučavanje postojeće literature, modela i analiza,
- Metode komparativne analize prikupljenog i proučenog materijala,
- Metode analize i sinteze, zasnovane na stečenom znanju i iskustvima,
- Primena eksperimentalnih i kvantitativnih metoda, po potrebi.

1.5 Očekivani rezultati

Rezultat disertacije je:

- prikaz novog koncepta sistema koji za pretraživanje elektronskih poslovnih dokumenata koristi tekst i sliku.
- dizajniranje i implementacija novog rešenja,
- donošenje opštih zaključaka na osnovu eksperimentalnih rezultata.
- dobijanje jasnije slike o upotrebljivosti predloženog rešenja u praksi.

1.6 Struktura rada

Rad je strukturiran u deset poglavlja.

U prvom poglavlju dat je uvod u problematiku pronalaženja informacija.

Pregled postojećih desktop aplikacija za pretraživanje personalnih informacija dat je u drugom poglavlju.

U trećem poglavlju detaljno se razmatra problematika kombinovanog pretraživanja teksta i slika.

U četvrtom poglavlju dat je kratak osvrt na problematiku relevantnosti pronalaženja informacija i potencijalna rešenja za prevazilaženje problema.

Peto poglavlje se bavi tipičnom arhitekturom aplikacija ovog tipa i izborom tehnologije za implementaciju.

Šesto poglavlje odnosi se na dizajn aplikacije. Dat je pregled funkcionalnih mogućnosti indeksiranja, analiziranja i parsiranja softverske biblioteke Lucene.net.

U sedmom poglavlju prikazana je realizovana desktop aplikacija sa grafičkim korisničkim interfejsom za *ad-hoc* pronalaženje informacija. Navedeni primeri izvršenih eksperimenata na realnim poslovnim dokumentima.

U osmom poglavlju dati su zaključci izvedeni iz analize rezultata izvršenih eksperimenata i navedeni pravci daljih istraživanja.

U devetom poglavlju dat je obiman pregled korišćene stručne i naučne literature, a u desetom spisak korišćenih skraćenica.

2. Problem istraživanja

Kao pripadnici informatičkog društva svakodnevno stvaramo, skladištimo i koristimo digitalne informacije. Količina informacija sa kojom moramo da se izborimo svakim danom je sve veća. Za par decenija, kancelarije su se iz potpuno analognog sveta „preselile” u skoro potpuno „bezpapirno” okruženje u kome se informacije skladište u digitalnom obliku, kao fajlovi dokumenata, e-pošte, slika, veb stranica itd.

Kada pokušavamo da izađemo na kraj sa tom količinom digitalnih informacija kojoj smo izloženi svaki dan, koristimo različite strategije kako se ne bismo izgubili u obilju informacija ili izgubili informacije. Te aktivnosti nazivamo opštim imenom upravljanje personalnim (ličnim) informacijama (engl. *Personal Information Management – PIM*). Prvi korak u upravljanju ličnim informacijama posle kreiranja je njihovo organizovanje. Kada je kasnije neka informacija potrebna, preduzima se proces pronalaženja informacije (engl. *Information retrieval*). Kako količina informacija raste, pronalaženje postaje zametno, posebno ako su dokumenti loše organizovani.

Termin pronalaženje informacija (*information retrieval*) prisutan je u oblasti računarstva i informatike već dugi niz godina. Kao i druga informatička terminologija pravi problem u srpskom jeziku jer prilikom prevođenja ovog termina s engleskog jezika on gubi svoje pravo značenje.

Collins Cobuildov rečnik definiše ovaj pojam kao: informacije koje smo već imali, ali su sakrivene i moramo ih pronaći. U ostalim rečnicima prevodi za ovu reč su: setiti se, dospeti u pamet, naći, doneti, ponovo dobiti, ponovo naći, dobiti podatke iz kompjutera.

Zbog svih tih različitih značenja teško je pronaći odgovarajući sinonim u srpskom jeziku. Najadekvatniji prevod bi bio pribaviti, povratiti, ali ta značenja nailaze na podsmeh. Zato je jedini moguć termin pronalaženje informacija.

Da bi se pospešio proces pronalaženja razvijene su različite softverske aplikacije. Neki sistemi su fokusirani na pronalaženje, dok drugi nude kompletno okruženje za upravljanje. Međutim, zajednički cilj svih sistema je da obezbede krajnjim korisnicima interfejs pomoću koga će obrađivati svoje lične informacije, bez obzira na tip.

Kao pojedinci sa različitim ponašanjem, svi mi imamo svoj jedinstven način postupanja sa svojim ličnim dokumentima. Dok su neki ljudi pedantni i vole da drže stvari na organizovan način,

dugi se sasvim solidno snalaze u svom haosu. Razlika u strategijama organizovanja može da utiče na to koliko će biti uspešno pronalaženje „zagubljenih” dokumenata, a obično je olakšano kod pedantnih korisnika.

Zbog povećanih smeštajnih mogućnosti standardnih personalnih računara, korisnici više nisu motivisani da brišu stare fajlove. Korisnici teže da sačuvaju ogromne količine multimedijalnih podataka, tj. tekstualnih dokumenata, slika, audio i video fajlova, elektronske pošte, prezentacija itd. Praksa „nikada ne brisati fajlove” ima izrazite prednosti pošto osigurava da važni podaci nikada neće biti slučajno uklonjeni. Međutim, kao nepovoljan bočni efekat, personalni računar često postaje prenatrpan. Stoga, lociranje specifičnog fajla unutar sistema fajlova može postati izazov ili čak zastrašujući zadatak.

Kako kompjuterizovani sistemi mogu pomoći korisniku u pronalaženju „zaturenih” dokumenata/informacija podržavajući prelistavanje i unapređujući strategiju pretraživanja?

Nedavno, izazov razvijanja alata za efektivno desktop pretraživanje proučavan je i u akademskom i u komercijalnom okruženju [1], [2], [3], [4], [5], [6], [7]. Mada se ti alati razlikuju po nekim mogućnostima i obliku korisničkog interfejsa, većina prihvata isto intuitivno okruženje koje se ogleda u sledećem:

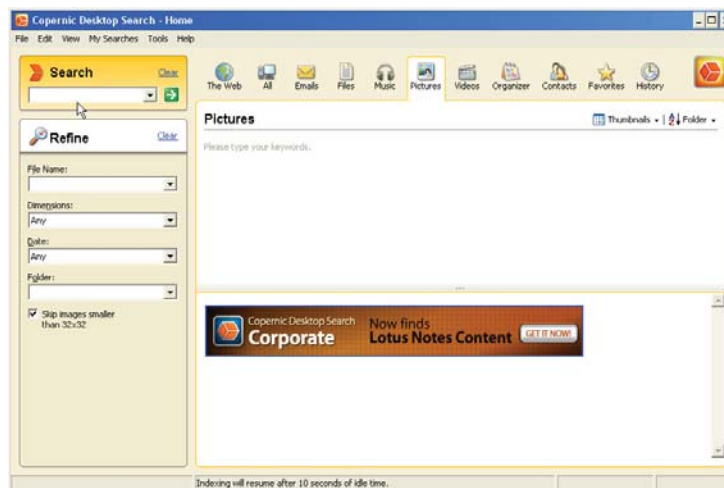
1. definisanje upita kao malog skupa ključnih reči
2. pretraživanje skupa dokumenata koji su povezani na neki način sa ključnim rečima upita
3. prezentiranje korisniku tih dokumenata uređenih prema nekoj primarnoj šemi rangiranja i
4. omogućavanje korisniku da preuredi listu dokumenata prema sekundarnim šemama rangiranja, koje su opšti atributi kao što su datum poslednjeg pristupa, ime fajla, tobožnja relevantnost sadržaja dokumenta prema upitu itd.

Karakteristično je, da iako je poslednje svojstvo lako implementirati, mnogi komercijalni alati za desktop pretraživanje pretežno koriste lakoničnije oblike korisničkog interfejsa, koji korisniku vraćaju jednu (verovatnu strukturiranu) rangiranu listu rezultata pretraživanja [5], [7]. Ovo podržava ono što mi verujemo da bi moglo biti inherentno svojstvo pretraživanja lične sfere podataka – tipična pretpostavka korisnika desktop pretraživanja je da oni znaju kako korektno da postavite upit sistemu za željenim fajlom i da oni očekuju da će sistem taj fajl locirati uspešno.

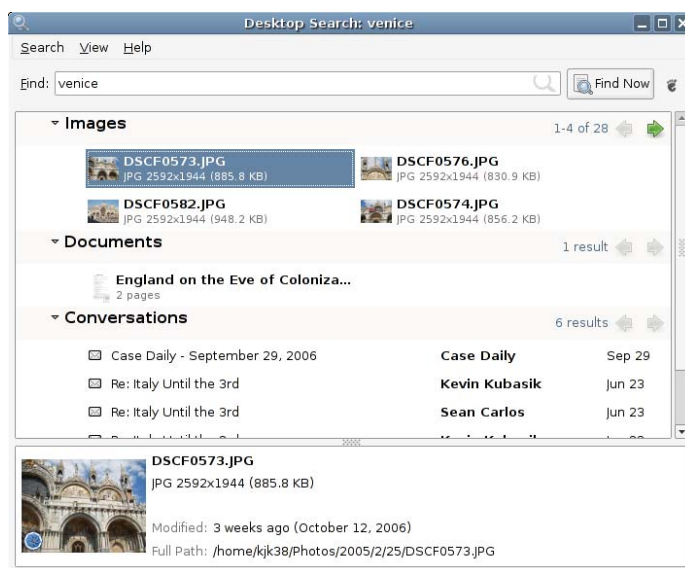
Korak koji danas većina planera aplikacija preduzima sastoji se u dizajniranju interfejsa koji integriše lako pristupačna polja pretrage u aplikaciju. To eliminiše potrebu da korisnici otvaraju posebnu aplikaciju za pretragu kada žele nešto da pronađu. Međutim, rešenje sa integrisanim poljem za pretragu nije savršeno. Pretpostavimo da korisnik traži neku konkretnu informaciju za koju zna da ju je već video (recimo, adresa ustanove ili telefonski broj), ali ne može da se seti u kom tipu dokumenta (Word dokument, veb strana ili PowerPoint slajd) ju je video. U tom slučaju, korisnik bi morao da pokrene više aplikacija kako bi učitao i pretražio odgovarajuće vrste dokumenata unutar tih aplikacija ne bi li pronašao željenu informaciju.

Da bi se razrešio ovaj problem razvijene su aplikacije koje funkcionišu kao interfejsi unificiranog pretraživanja. Ovim aplikacijama se u opštem slučaju zadaje skup pitanja i stavki koje se indeksiraju. Na primer, korisnik može da zatraži indeksiranje celokupne svoje elektronske pošte ili fajlova u svom matičnom folderu. Pošto su sve stavke indeksirane, pretraživanje je vrlo brzo i interfejs može biti dizajniran da odmah prikaže rezultate. Primeri ove vrste aplikacija su Goog-

le Desktop (Windows), Microsoft Windows Desktop Search (Windows), Copernic Desktop Search (Windows), Beagle (Linux) i Spotlight (Mac OS X).

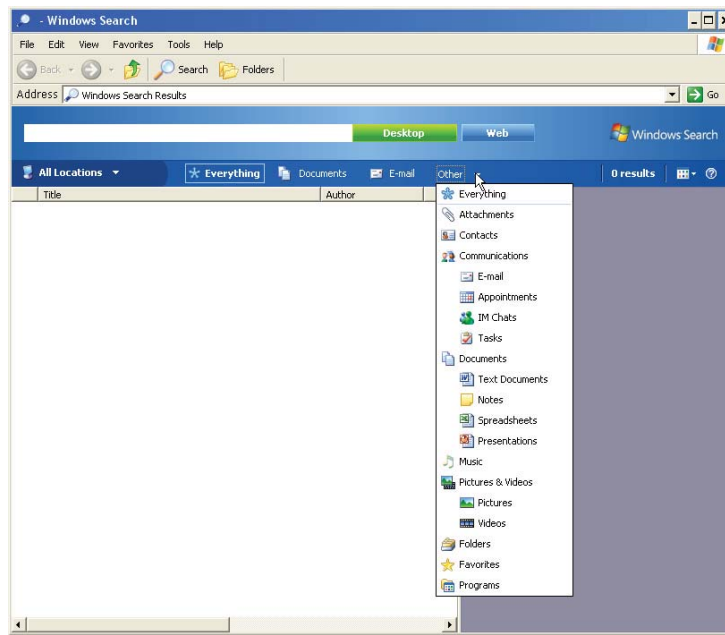


Slika 2.1 Copernic Desktop Search



Slika 2.2 Beagle (Linux)

Microsoft je razvio svoj alat Windows Desktop Search (WDS) za personalno indeksiranje i pretraživanje za svoj Windows operativni sistem. Ovaj alat kreira i upravlja indeksom korisničkih fajlova na računaru i omogućava korisniku da lako postavi upit indeksu na Windowsovoj paleti poslova.



Slika 2.3 Windows Desktop Search

Inicijalno, WDS indeksira Desktop i folder MyDocuments, kao i e-poštu Microsoft Outlooka. Korisnik može lako da doda i druge lokacije. Servis indeksiranja radi u pozadini, ažurirajući indeks kadgod se nova stavka kreira, obriše ili premesti. Niz mera je preduzeto da WDS ne utiče na korisnikov komfor pri aktivnom radu. Na primer, WDS zaustavlja indeksiranje kada korisnik aktivno koristi računar. Takođe, korisnici laptop računara imaju opciju da prekinu indeksiranje kada računar koristi baterijsko napajanje.

WDS zadovoljava principe privatnosti, tj. ako korisnik nema prava pristupa fajlu ili resursu, nema ni WDS. Osim toga, indeks se smešta u zaštićenu oblast na hard disku, obično u korisnikov Windows profil. Takođe, indeks je kriptovan iz bezbednosnih razloga da bi se sprečila zloupotreba kopiranjem na drugu lokaciju.

Sprovedena su brojna istraživanja u oblasti upravljanja personalnim informacijama (engl. *Personal Information Management – PIM*). Ta istraživanja i projekti pomogli su da se razviju tehnologije koje omogućavaju korisnicima da smeštaju, pristupaju i efektivno pretražuju informacije na svojim personalnim računarima. Zbog specijalne prirode svojih ciljeva PIM spajaju istraživače iz širokog spektra naučnih i inženjerskih disciplina, počevši od kognitivne psihologije (gde se termin upravljanje personalnim informacijama prvo pojavio [8], preko upravljanja bazama podataka i pretraživanja informacija. Konkretno, u radu [9] smatra se da će PIM biti jedan od najvećih istraživačkih izazova u narednim godinama.

Cilj ove teze je dizajniranje i implementiranje aplikacije za tzv. multimodalno desktop pretraživanje personalnih elektronski generisanih poslovnih dokumenata, tj. efektivno pronalaženje tih dokumenata unutar personalnog računara na osnovu upita koji kombinuje tekst i sliku. To je važan problem zato što desktop mašina za pretraživanje treba da olakša lociranje informacije koja nam je potrebna, čak i kada je ona zatrpana u hrpama nepovezanih informacija.

Dokument isporučuje informaciju koristeći više modaliteta, uključujući tekst, raspored/stil i slike. Model multimodalnog dokumenta se definiše kao skup više objekata koji mogu biti tekstualni i slikovni. Tekstualni objekat predstavlja blok teksta. Slikovni objekat predstavlja sliku/grafiku

Logo znaci se uobičajeno koriste u poslovnim i zakonskim dokumentima kao deklaracija izvora i vlasništva dokumenta. Problem detekcije i prepoznavanja logo znaka je od velikog značaja kod manipulacije elektronskim poslovnim dokumentima jer omogućava da se identifikuje izvor dokumenata na osnovu organizacije iz koje je potekao. Suočavajući se sa svakodnevnim povećanjem broja elektronskih poslovnih dokumenata koji su u opticaju, prepoznavanje logoa postaje praktičan i pouzdan dodatak u pretraživanju tekstualnog sadržaja dokumenata. Poznavajući tematski sadržaj elektronskog poslovnog dokumenta (profaktura, potvrda, odluka, rešenje itd.) sužava se skup dokumenata za pretraživanje a logo obezbeđuje dodatni oblik indeksiranja koji omogućava efektivnije pretraživanje izdvojenog skupa. Za datu veliku kolekciju dokumenata, traženje specifičnog logoa je veoma efikasan način pretraživanja dokumenata pripadajuće organizacije.

Standardno, jedan modalitet, bilo tekst ili slika, koristi se za pretraživanje sadržaja u multimodalnim dokumentima.

Sistemi multimodalnog pretraživanja informacija koriste i tekstualne i slikovne objekte. Pretraživanje informacija koje koristi ili tekstualni ili slikovni modalitet ima i prednosti i nedostatke. Sistemi multimodalnog pretraživanja informacija se zasnivaju na jednomodalnim sistemima pretraživanja informacija i mnogo su složeniji od njih. Mogu da se koriste kombinovani upiti koji uključuju i tekst i slike. Oba sadržaja, i tekstualni i vizuelni su indeksirani i različite tehnike kombinovanja teksta i slika su razvijene.

Multimodalno pretraživanje informacija je ogromna istraživačka oblast. Blisko je povezana sa sledećim istraživačkim oblastima: pretraživanje teksta, pretraživanje slika, pretraživanje skeniranih dokumenata i klasifikacijom dokumenata. Svaka od ovih istraživačkih oblasti je prilično velika i veoma dinamična, tako da je nemoguće napraviti sveobuhvatni pregled tekućeg stanja.

Uprkos zamašnom i aktuelnom istraživanju kako u oblasti pretraživanja teksta, tako i u oblasti pretraživanja slika, ostaje još mnogo toga da se uradi kako bi se došlo do efikasnog komplementarnog integrisanja ova dva pristupa u jedno efektivno okruženje.

Komercijalni i socijalni uspeh klasičnih i tehnika za pretraživanje informacija na vebu izgleda da implicira da bi desktop pretraživanje moglo uspešno da se osloni na korišćenje alata koji su razvijeni za te zadatke. Međutim, multimodalno desktop pretraživanje je drugačije u mnogo čemu od pretraživanja u nepoznatom repozitorijumu podataka, kao što su digitalna biblioteka ili veb. Možda najistaknutija razlika je ta što korisnici traže na svojim personalnim računarima uglavnom konkretne fajlove za koje znaju da postoje.

3. Kombinovano pronalaženje teksta i slike – presek stanja

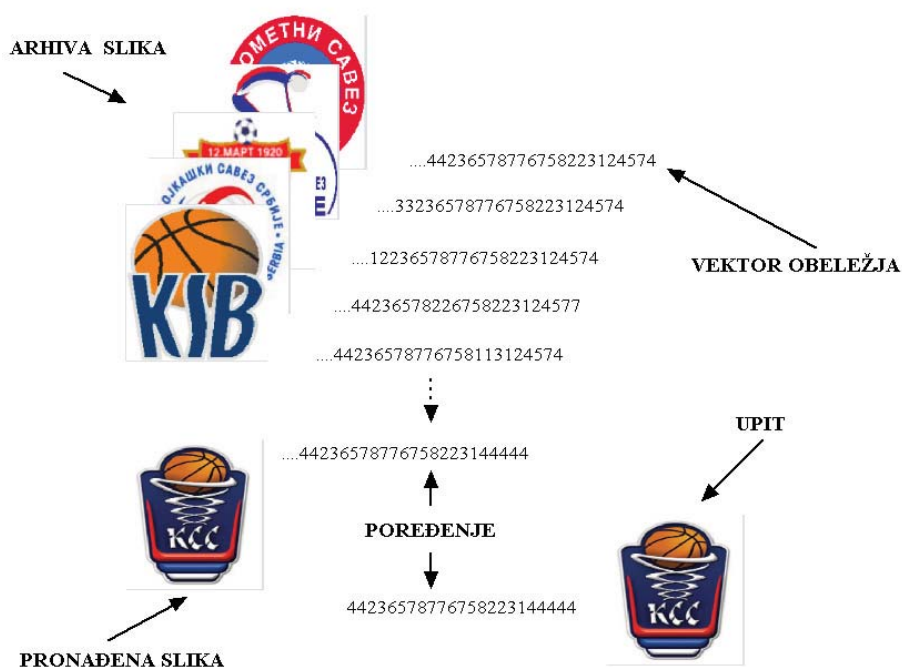
Praksa pronalaženja informacija sve do ranih 70-tih godina prošlog veka bila je deo svakodnevnih obaveza bibliotekara. Istoričari tvrde na osnovu nekih nalaza da praksa pronalaženja informacija datira još od vremena prvih bibliotečkih sistema registrovanih u istoriji ljudske civilizacije. Poznato je da je biblioteka u Aleksandriji raspolagala sa katalogiziranih i klasifikovanih 500 000 jedinica. Od tih vremena pa nadalje, razvijale su se teorije i metodologije za bibliotečko pronalaženje informacija koje su bile usmerene na štampane tekstualne dokumente. Većina od njih je još uvek primenjiva na poslove opšteg pronalaženja informacija. Uz nešto malo generalizacije i modifikacije, moguće ih je prilagoditi scenarijima pronalaženja multimedijalnih informacija i obezbediti uvid u problem pronalaženja multimedijalnih informacija iz opšte perspektive informacionog sadržaja [10].

3.1 Dokument i upit

Informacioni sistem sastoji se od dva glavna tipa entiteta: dokumenta i upita. Dokumenti mogu biti bilo kog formata, sa bilo kojom vrstom interne strukture u smislu organizovanja informacija koje sadrže. Sistem za pronalaženje informacija tretira sve dokumente u kolekciji kao nestrukturirane fajlove. Dokumenti se mogu reprezentovati sa surogat dokumentima. Surogat dokument je obično koncizna reprezentacija originalnog dokumenta koja je izdvojena procesom obrade za konkretne svrhe i sadrži samo mali deo informacije koju kompletni dokument nosi (sadrži). Neki primeri surogata dokumenta su apstrakti članaka i ikone slika. Oni su pogodniji za efikasno skladištenje, brz pristup i obradu. Upit je pokazatelj (indikacija) korisnikovog zahteva za informacijom. On može imati više različitih oblika, od vrlo konciznog Boolovog upita, preko onih sa vrlo detaljnom specifikacijom vrste dokumenta koju korisnik zahteva, do onih sa jednim ili više uzoraka dokumenta (slučaj slika upita) koji se zadaju kao primeri vrste željenog dokumenta.

Istraživači posmatraju upite i surogat dokumente kao specijalne vrste dokumenata koji nose većinu istih karakteristika kao regularni dokumenti. Proces pronalaženja se ogleda u pronalaženju dokumenata koji odgovaraju dokumentu upitu prema postavljenom kriterijumu posredstvom procedure obrade koja se primenjuje i na upit i na dokumente u arhivi [11].

Surogat dokument (kao i upitni surogat) treba da zadrži što više korisnih informacija dok šum treba da je u što većoj meri uklonjen. Neki istraživači definisali su obeležje dokumenta kao ekstrahovanu informaciju iz objekta i korišćenu u toku procesiranja upita [12]. Prema tome, svi objekti u informacionom skladištu sistema pronalaženja mogu da se objedine kao dokumenti, a svi dokumenti se obično procesiraju da bi se generisali surogati dokumenata, pre nego što se upit definiše. Surogati dokumenata, specijalna kompaktna reprezentacija skupa svih bitnih obeležja dokumenta, su relativno statični i smeštaju se kao trajni deo informacionog sistema, sa dužim životnim vekom nego što ga imaju efemerni upiti. Na sledećoj slici prikazana je tipična arhitektura *Content Based Information Retrieval (CBIR)* sistema sa dokumentima slikama, surogat dokumentima (vektori obeležja) i slikom upita. U ovom radu se pod obeležjem dokumenta smatra element informacije koji je ekstrahovan iz originalnog dokumenta u svrhu indeksiranja i pronalaženja.



Slika 3.1 Arhitektura tipičnog sistema za pronalaženje slika na osnovu sadržaja

3.2 Obeležja i pronalaženje

Obeležje dokumenta je važan koncept u pronalaženju informacija. Ono definiše način na koji se dokumenti porede jedan sa drugim. Takođe, definicija obeležja dokumenta implicira nekoliko osnovnih metodologija pronalaženja informacija. U suštini, pronalaženje informacija je proučavanje statističkih karakteristika nestrukturiranih podataka. U slučaju pronalaženja teksta, obeležje dokumenta je term koji je indeksiran i može imati oblik reči, fraze ili N-grama.

Pronalaženje informacija pokušava da pronađe statističke uzorke obeležja dokumenta u podacima bez razumevanja strukture dokumenta ili implikacija obeležja. Prvo, svi dokumenti su nestrukturirani, što znači ili je struktura podataka nepoznata ili semantika komponenta podataka nije poznata. U pronalaženju informacija dokument je definisan kao objekt informacija sa nepoznatom strukturom [13]. Nasuprot strukturalnim reprezentacijama koje su popularne u drugim istraživačkim oblastima kao što su sistemi za upravljanje relacionim bazama podataka (engl. *Relational Database Management System – RDBMS*), procesiranje prirodnih jezika (engl. *Natural Language Processing – NLP*) i tehnike proširenog markerskog jezika (engl. *Extended Markup Language – XML*), glavnina besplatnih tekstualnih dokumenata, audio zapisa i pikseljskih reprezentacija slika su nestrukturirani podaci.

Drugo, pronalaženje informacija ne čini nikakav napor u interpretiranju značenja termova koji su indeksirani. Niti čini bilo kakve apriorne pretpostavke o njihovim značenjima. Jedina stvar koja se koristi da bi se donela odluka o pronalaženju su statistički uzorci obeležja dokumenta.

Treće, sva obeležja se tretiraju kao nezavisna jedna od drugih u trenutku procesiranja. To je dublja pretpostavka od druge i teorijski može biti opasna. Redundantna, dvosmislena priroda jezika koji koristimo određuje da postoje korelacije između obeležja. Takođe, različite interpretacije mogu se primeniti na istu reč u različitim kontekstima, što znači da interpretacija obeležja nije samostalna, nezavisna i slobodna od konteksta. Međutim, u praksi, svi modeli pronalaženja teksta rade fino pod tim pretpostavkama i pomažu uklanjanju mnogih barijera i doprineli su velikom uspehu koji je postignut u zadnjoj dekadi prošlog veka. Pojava mnogih uspešnih mašina za pretraživanje veba je dobar znak za uspeh teorije. Ranije navedene tri pretpostavke se obično navode kao „*bag of words*” pretpostavka u pronalaženju tekstualnih informacija, gde se svaka nezaustavna reč koristi kao indeksirani term bez obzira na njeno aktuelno značenje i internu strukturu dokumenta.

Ove pretpostavke, ustvari imaju više smisla kod pronalaženja multimedijalnih informacija, kod koga su multimedijalni objekti/dokumenti zapisi svojih fizičkih svojstava: 1) ne postoji interna semantička struktura za te zapise; 2) čovekova percepcija tih fizičkih svojstava preko senzorskog sistema nije neminovno povezana sa interpretacijom sadržaja i 3) ili su zapisana fizička svojstva nezavisna jedna od drugih ili mi možemo da ekstrahujemo samo nezavisna svojstva iz objekta i izostaviti bilo koju moguću redundantnu informaciju. Ova svojstva se mogu koristiti kao indeksna obeležja multimedijalnih dokumenata. Sasvim je opravdano da u pronalaženju slika kao specijalnom slučaju, možemo da usvojimo isti skup pretpostavki ali pod drugačijim imenom kao „*bag of pixels*”. Ova šema može da isključi takve tehnike kao što je prepoznavanje objekata iz velike mape u osnovnom smislu, kao što je uradilo pronalaženje tekstualnih informacija sa procesiranjem prirodnog jezika. Međutim, baš kao što se pokazalo kod pronalaženja teksta, očekuje se da kombinacija tehnika generiše mnogo više snage nego svaka od njih pojedinačno.

Ekstrakcija obeležja se obično dizajnira da odgovara potrebama za specijalnom informacijom imajući u vidu kontekst domena informacije. Posebna pažnja treba da se posveti tipovima upita na koje se očekuje da sistem odgovori, kao i prirodi kolekcije dokumenata. Mada je to jedan od najvažnijih koraka u pronalaženju, ne postoje opšta pravila koja su primenjiva na varijetete problema. Dokument može da bude indeksiran ručno (identifikovanje ključne reči od strane autora publikacije) ili automatski od strane računara. Obeležja mogu biti pažljivo odabrana da ciljaju konkretan tip sadržaja, tj. indeksiranje tekstualnih dokumenata sa kontrolisanim rečnikom kao što su termini iz tezaurusa ili se obeležja mogu masovno automatski ekstrahovati iz dokumenata

a zatim odabrati prema njihovoj relevantnosti za pronalaženje. Pokazalo se u praksi da se često postižu dobri rezultati kombinovanjem navedena dva pristupa zbog činjenice da nije verovatno da će oba pristupa praviti iste greške kod pronalaženja nerelevantnih dokumenata, već da će se pre složiti po pronalaženju zaista relevantnih dokumenata.

3.3 Metodologija pronalaženja

Imajući definisan koncept obeležja dokumenta, proces nalaženja može se posmatrati kao proces mapiranja, kod koga se dokumenti iz prostora dokumenata projektuju u prostor obeležja, posle koga sledi proces poklapanja/poređenja prema modelu pronalaženja koji grupiše dokumente sa sličnim obeležjima.

Prema usvojenoj pretpostavci o raspodeli vrednosti obeležja, koristi se više modela pronalaženja za donošenje odluka o pronalaženju. Ne doprinose sva obeležja podjednako razlikovanju sadržaja dokumenata. Raspodela vrednosti takođe utiče na njihov kvalitet kao diskriminirajućih obeležja. Različiti modeli pronalaženja koriste različite mehanizme kod određivanja doprinosa svakog od obeležja odluci o pronalaženju.

Do sada, više modela za pronalaženje informacija pokazalo je veliki uspeh, uključujući Boolov model, model vektorskog prostora, probabilistički model, model fazi poklapanja i modeli analize citiranosti. Pored toga, veštačke neuralne mreže i tehnike obrade prirodnog jezika se takođe koriste. Ovi modeli imaju vrlo različite poglede i pretpostavke o raspodeli obeležja. Shodno tome, definicija o sličnosti dokumenta znatno varira od modela do modela. Na primer, u nekim slučajevima, postoji merljiva distanca dokumenta koja je inverzno proporcionalna sa merom sličnosti; dok u drugim sistemima, sličnost dokumenta može da se izvede samo indirektno iz verovatnoće da su dva dokumenta slična jedan drugom. Ovaj potonji slučaj obuhvata kriterijum pronalaženja određen istovremenim pojavljivanjem podatka i/ili korisnikove reakcije koji nisu karakteristike samih dokumenata već veza nametnutih kroz korisnikovo ponašanje tokom traženja informacija. Prema tome, relevantnost dokumenta može biti bolji term nego sličnost dokumenta u opisivanju kriterijuma pronalaženja. Važno je posmatrati proces pronalaženja kao proces koji poklapa/poređi dokumente i upit, koji je predstavnik korisnikovog informacionog zahteva. Usled masovnog uspeha modela vektorskog prostora, mera sličnosti dokumenta se pogrešno smatra kao jedini kriterijum za pronalaženje.

Nije iznenađujuće da je u praksi većina modela dala relativno približne nivoe performansi. Ovo nesumnjivo ostavlja prostor za dalja istraživanja različitih modela pronalaženja, kombinovanje različitih modela i uvođenje novih tehnika.

3.4 Multimodalni dokumenti

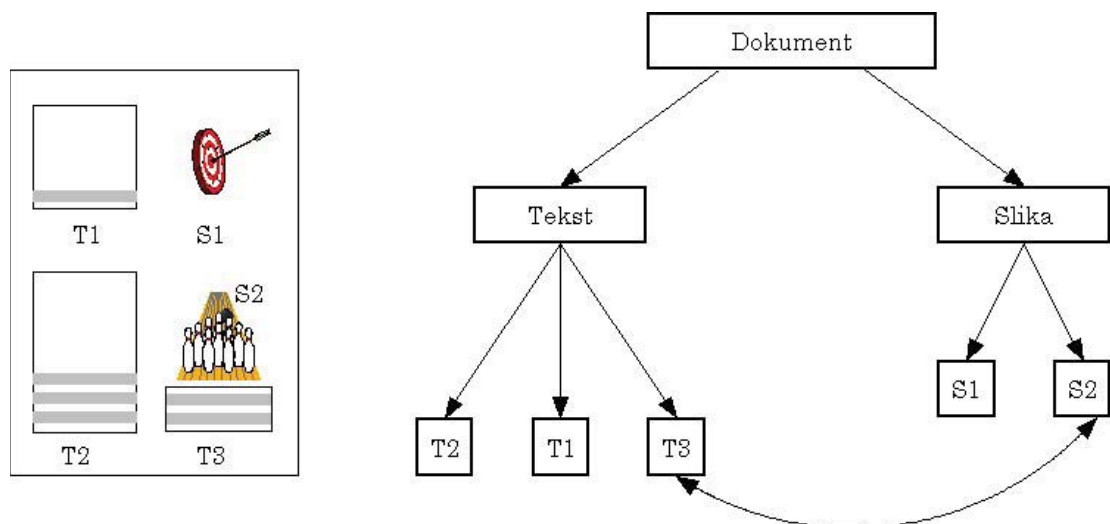
Dokument prenosi informaciju koristeći više modaliteta, uključujući tekst, raspored/stil i slike. Na primer, članci u časopisima obično imaju slike koje ilustruju eksperimentalne rezultate a naslovi obično imaju veću veličinu fonta od tela teksta. Primeri multimodalnih dokumenata pri-

kazani su na slici 3.2. Indeksiranje i pronalaženje samo pomoću teksta je tradicionalna metoda u sistemima za pronalaženje informacija. Širenjem Interneta i pojavom digitalnih biblioteka nametnula se potreba za razvojem tehnika za inteligentno indeksiranje i pronalaženje multimodalnih dokumenata.



Slika 3.2 Primeri multimodalnih dokumenata

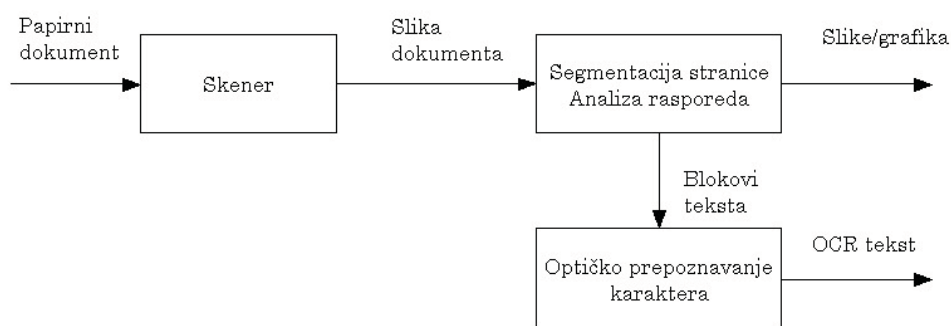
Postoji široka lepeza računarski dostupnih multimodalnih dokumenata. Na osnovu izvora kreiranja postoje dve glavne kategorije, elektronski dokumenti i slike dokumenata. Elektronski dokumenti su proizvod računarskih programa za obradu dokumenata. Primeri takvih dokumenata su naučne publikacije u PDF formatu, kancelarijski dokumenti u DOC formatu, veb stranice u HTML formatu itd. Slike dokumenata nastaju skeniranjem papirnih formi dokumenata ili iz faks mašina i obično se reprezentuju u TIFF formatu.



Slika 3.3 Model multimodalnog dokumenta

Model multimodalnog dokumenta može se definisati kao skup više objekata koji mogu biti tekstualnog (T) ili slikovnog (S) tipa. Tekstualni objekat predstavlja blok teksta a slikovni objekat predstavlja sliku. Slike obuhvataju grafičke slike, tj. dijagrame, grafikone, crteže, mape i slike iz realnog sveta, tj. crno-bele i kolor fotografije. Neki blokovi teksta nisu povezani sa slikama i oni se zovu nezavisni tekst. Drugi blokovi teksta su povezani sa slikama, kao što su potpisi ispod slika ili skup ključnih reči kojima su označene slike i nazivaju se kolateralni tekst. Neke slike nemaju kolateralni tekst. Na slici 3.3 prikazan je primer multimodalnog dokumenta. Blok teksta T3 povezan je sa slikom S2. T1 i T2 su nezavisni blokovi teksta. Slika S1 nema kolateralni tekst. Desni deo slike prikazuje strukturu hijerarhije tog multimodalnog dokumenta i relacije između objekata. Primeri sa slike 3.2 mogu se predstaviti tim modelom.

Za razliku od elektronskih dokumenata, kod slika dokumenata ne postoje eksplicitni objekti, stoga se obrada slika dokumenata oslanja na tehnike analize slike kako bi se detektovali tekstualni i slikovni objekti. Uprošćen dijagram analize slike dokumenta prikazan je na slici 3.4. Prvo se kroz segmentaciju stranice i analizu rasporeda slike dokumenta odvajaju tekst i slike. U toj fazi kolateralni tekst može se pridružiti slikama. Da bi se dobio tekstualni sadržaj, primenjuje se optičko prepoznavanje karaktera (engl. *optical character recognition* – OCR). Kod analize slike dokumenta mogu se pojaviti šum i greške, što se mora uzeti u obzir kod potonjeg indeksiranja i pronalaženja.



Slika 3.4 Procedure analize slike dokumenta

3.5 Tehnike indeksiranja

Tekstualni sadržaj može da se indeksira a takođe i vizuelni sadržaj. U sledećoj sekciji dat je pregled indeksiranja tekstualnog sadržaja, koje obično koristi sekvencu indeksnih termova da reprezentuje tekst. Različite vrste obeležja slike mogu se ekstrahovati da reprezentuju vizuelni sadržaj. Pregled indeksiranja vizuelnog sadržaja dat je u nastavku.

3.5.1 Indeksiranje tekstualnog sadržaja

Podrazumeva se da je tekst u elektronskim dokumentima bez šuma, nasuprot tekstu u slikama dokumenata (skenirani dokumenti) gde se javljaju OCR greške. Indeksiranje kolateralnog teksta

koje se koristi kod pronalaženja slika na osnovu teksta je slično sa indeksiranjem teksta bez šuma. Međutim, pronalaženje kolateralnog teksta nije trivijalan zadatak kod nekih dokumenata.

Indeksiranje teksta bez šuma

Indeksni termini (ili ključne reči) su osnovne jedinice za predstavljanje tekstualnog sadržaja dokumenata. U slučaju teksta na engleskom jeziku, indeksni termini se dobijaju korišćenjem sledećih procedura [14]:

1. Obavlja se leksička analiza teksta da se ekstrahuju iz dokumenata reči kandidati koje će se koristiti kao indeksni termini. U toku leksičke analize javljaju se razni problemi, takvi kao što je kako se odnositi prema ciframa (ISO 2000), znacima interpunkcije (text.doc), hifenaciji (crticom spojiti/odvojiti reči; on-line) itd.
2. Opciono, uklanjanje tzv. stop (funktionalnih) reči. Stop reči su reči koje se prilično često pojavljuju, kao što su *the, of, in*. Uklanjanje stop reči obično poboljšava preciznost i značajno redukuje veličinu strukture indeksa. Može da redukuje odziv.
3. Opciono, reči se „stemuju” (engl. *stemm*). Različite reči mogu nositi slična značenja (search, searching). Steming ima za cilj da grupiše zajedno reči sa zajedničkim oblikom. Steming obično poboljšava odziv i redukuje veličinu indeksne strukture. U nekim slučajevima može da redukuje preciznost.
4. Opciono, fraze koje sačinjava više reči se identifikuju kao dodatni indeksirajući termini.
5. Opciono, obavlja se sintaksna analiza rečenice i biraju se reči iz raznih gramatičkih klasa, kao što su glagoli, imenice itd. Obično se biraju imenice pošto se smatra da one apsorbiraju glavninu semantike dokumenta.

Ne postoji ubedljiv dokazni materijal da navedeni opcioni koraci kod procesiranja teksta daju konzistentno poboljšanje performansi pronalaženja, posebno kada je kolekcija dokumenata opšta i heterogena [14]. To je razlog zašto neke mašine za pretraživanje veba indeksiraju sve reči.

Indeksiranje konstruiše invertovani indeks od pokazivača reč-dokument. Invertovani fajl je dizajniran za brzo spajanje listi dokumenata i izračunavanje sličnosti.

Pronalaženje kolateralnog teksta

Indeksiranje kolateralnog teksta je slično indeksiranju teksta bez šuma. Međutim, postoje neizvesnosti lociranja kolateralnog teksta u nekim multimodalnim dokumentima, na primer kod veb stranica. Za razliku od novinskih tekstova, većina slika na veb stranicama nema eksplicitni potpis. Kod veb stranica, kolateralni tekst slike može da se ekstrahuje iz više izvora, kao što su naslov stranice, ime fajla slike, alternativni tekst i okružujući tekst koji obuhvata pravce gore, dole, levo i desno [15].

Kolateralni tekst može biti nekompletan ili relevantan. Ako nije razmatran tekst oko slike (tj. korišćeni su samo naslovi u arhivi vesti [16]), kolateralni tekst može da bude nekompletan. Sa druge strane, ako se koristi sav tekst veb stranice kao kolateralni tekst, neki tekst može biti irelevantan. Razvijene su razne tehnike za nalaženje okružujućeg teksta povezanog sa slikom. Na primer, Mukherjea i Chost koriste kriterijum vizuelne i sintaksne distance između slika i poten-

cijalnih potpisa da povežu tekst sa slikama [17].

Pošto relevantnost ili važnost raznih izvora teksta može biti različita, neki sistemi daju različite težine tekstu iz različitih izvora. Kod *ImageRovera*, reč relevantna sa slikom se identifikuje na osnovu svoje frekvencije pojavljivanja u HTML tekstu, svog položaja u odnosu na sliku i svog stila [18]. Rečima sa specifičnim HTML tagovima daju se veće težine (ponderi). Donošenje odluke o dodeljivanju pogodnih težina relevantnosti različitih izvora je težak problem.

Indeksiranje tekstualnog sadržaja slika dokumenata

Zbog grešaka optičkog prepoznavanja karaktera potrebne su robusne tehnike indeksiranja tekstualnog sadržaja koji potiče iz skeniranih dokumenata. Sveobuhvatni pregled indeksiranja slika dokumenata dat je u [19], [20]. Postoje dva osnovna pristupa indeksiranju:

Indeksiranje tekstualnog sadržaja koristeći OCR rezultate – Ovaj tip indeksiranja zahteva potpunu konverziju teksta slika dokumenata pomoću OCR-a. Predloženo je više tehnika koje su se bavile OCR greškama. Lopresti koristi metodu N-grama da indeksira OCR tekst [21]. N-grami razlažu dokument u n-karakter termove. Na primer, „DOCUMENT” se razlaže u nekoliko trigrama: DOC, OCU, CUM, UME, MEN i ENT. Svaki trigram je indeksirani term. Ovo N-gram indeksiranje stvara indeksirajuću strukturu velike veličine. Drugi pristup primenjuje korigovanje OCR grešaka pomoću rečnika ili statistike jezika a zatim tekst bez šuma indeksira [22]. Ovaj pristup obično ne može da potpuno ukloni OCR greške.

Indeksiranje slika teksta bez OCR-a – OCR je sklon da pravi greške kod nekvalitetnih slika dokumenata (recimo, faksovi ili dokumenti kopirani kopir-mašinom, stari dokumenti i rukopisi). Tekstualni sadržaj može se indeksirati direktno iz slika bez OCR-a. Na primer, tehnika Character Shape Codes zasnovana je na obeležjima slike karaktera i koristi se da indeksira tekstualni sadržaj dokumenta [23]. Kod procedure indeksiranja, prvo se slika segmentira na nivoima reči i karaktera. Zatim se svaki karakter u reči mapira u Character Shape Code bez identifikovanja. Ovi kodovi zajedno formiraju Word Shape Token za svaku reč, koji se koristi kao indeksni term. Ovaj pristup može da bude jeftiniji od potpune OCR konverzije i može da bude robusniji u odnosu na šum. Takođe, funkcioniše kod svih jezika i skriptova. Nije pogodan za primenu kod reči sa uzdužnim karakteristikama pošto oni ne mogu da se segmentiraju. Alternativno, može se koristiti indeksiranje slika reči [24]. Obeležja na nivou slike reči se kodiraju bez segmentacije karaktera i OCR-a.

3.5.2 Indeksiranje vizuelnog sadržaja

Obeležja slike mogu se ekstrahovati na globalnom nivou, tj. iz cele slike, ili lokalnom nivou, tj. iz regiona ili objekata slike. Da se dobiju lokalna obeležja slike, slika se često prvo deli na delove [25]. Najprostiji način je da se slika podeli u nezavisne delove. Ova vrsta podele ne generiše perceptualno značajne regione ali je način reprezentovanja globalnih obeležja slike sa finijom podelom. Alternativni metod je da se slika segmentira u istaknute regione prema nekom kriterijumu, kao što su boja i tekstura [26]. Segmentacija u regione nije trivijalan zadatak. Nesumnjivo, postoje neizvesnosti u segmentaciji regiona.

Obeležje (engl. *feature*) je izvedeni atribut dobijen transformacijom originalnog objekta pomoću

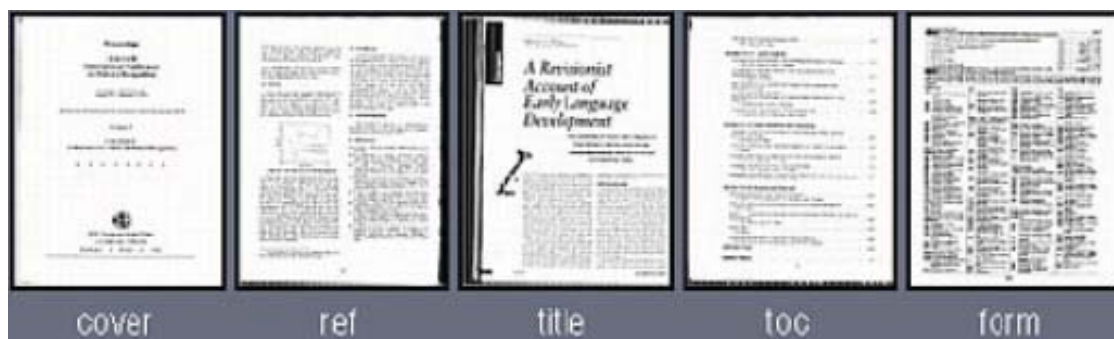
algoritma za analizu slike. Obeležje se tipično predstavlja nizom brojeva i često se naziva vektor obeležja (engl. *feature vector*).

Obeležja boje, teksture, oblika i prostornog rasporeda su najčešće korišćena vizuelna obeležja u pronalaženju slika na osnovu sadržaja.

Indeksiranje vizuelnog sadržaja slika dokumenata

Kod slika dokumenata vizuelni sadržaj može poticati od slika unutar slike dokumenta. Indeksiranje slika unutar dokumenta je isto kao indeksiranje vizuelnog sadržaja kod CBIR. Razlika je u tome što su slike kod većine CBIR sistema kolorne slike sa motivima iz prirode, tako da se obeležja boje opsežno istražuju i koriste. Nasuprot tome, većina slika dokumenata su crno-bele slike ili slike sive skale, usled čega se različiti skup obeležja koristi.

Celokupna slika dokumenta može da se tretira kao jedna slika i indeksira za pronalaženje na osnovu vizuelne sličnosti. Slike dokumenata na kojima je pretežno zastupljen tekst imaju sasvim drugačije strukturalne karakteristike od slika prirode i grafičkih slika kod CBIR sistema. Na slici 3.5 prikazani su primeri slika dokumenata sa osobenim strukturalnim obeležjima. Strukturalna obeležja se ekstrahuju kao rezultat analize rasporeda elemenata na dokumentu. Na primer, reprezentacija modifikovanog XY-stabla se koristi za pronalaženje stranice dokumenta [27]. Reprezentacija XY-stabla je dobro poznat pristup za opisivanje fizičkog rasporeda dokumenta [28]. Koren XY-stabla se pridružuje slici celokupnog dokumenta. Dokument se deli na regione koji su razdvojeni belim oblastima. Svaki čvor stabla pridružuje se regionu dokumenta.



Slika 3.5 Primeri različitih tipova slika stranica dokumenata

3.6 Modeli pronalaženja

Model vektorskog prostora [29] je najpopularniji model za pronalaženje teksta. Kolekcija dokumenata $D = (d_1, \dots, d_m)$ može da se posmatra kao vektori u vektorskom prostoru velike dimenzionalnosti. Svaki term u indeksu jezika $(t_1, \dots, t_n)$ postaje nezavisna dimenzija (ili obeležje).

Dokument d_i se reprezentuje kao n-dimenzionalni vektor $\vec{d}_i = (w_{i1}, \dots, w_{in})$. w_{ji} je težina od t_j u d_i koja izražava koliko dobro obeležje t_j opisuje sadržaj dokumenta d_i . Upit se smatra kratkim dokumentom i reprezentuje se kao vektor u istom vektorskom prostoru kao i dokumenti. Funkcija

pronalaženja izračunava sličnost između vektora opisa svih dokumenata i vektora opisa upita, a zatim vraća k dokumenata rangiranih prema bliskosti njihovih vektora vektoru upita.

Koriste se razne funkcije sličnosti, kao što je Jaccard, unutrašnji proizvod i kosinusna sličnost [14]. Kosinusna sličnost se najčešće koristi. Za normalizovane vektore, kosinus je jednostavno unutrašnji proizvod, kao što se može videti iz sledeće jednačine.

$$\text{sim}(d_i, d_k) = \vec{d}_i \cdot \vec{d}_k = \sum_{j=1}^n w_{ji} w_{jk} \quad (3.1)$$

gde su $\vec{d}_i, \vec{d}_k$ normalizovani vektori dokumenta.

Model vektorskog prostora vraća rangirane dokumente u redosledu njihove potencijalne korisnosti za pretraživača.

Ponderisanje termova

Način ponderisanja termova utiče na efektivnost modela vektorskog prostora. Indeksni termovi u kolekciji dokumenata imaju različitu frekvenciju. Postoje tri glavna faktora kod ponderisanja termova:

1. **tf** (*term frequency*), frekvencija terma u dokumentu. Termovi koji su frekventni u dokumentu važni su indikatori sadržaja dokumenta.
2. **df** (*document frequency*), broj dokumenata koji sadrže term. Termovi koji su frekventni u kolekciji nisu korisni u razlikovanju sadržaja dokumenta. Obično se koristi **idf** (*inverse document frequency*) da reprezentuje važnost terma.
3. Dužina dokumenta. Dokumenti obično variraju u dužini. Duži dokumenti imaju tendenciju da se bolje kotiraju pošto koriste više termova i iste reči mogu da se ponavljaju.

Razvijene su razne metode za ponderisanje termova. Najpopularnije je $tf \cdot idf$ ponderisanje. Ovo ponderisanje se koristi u više varijacija. Jednostavno $tf \cdot idf$ ponderisanje je dato jednačinom (3.2). Ponderisanja se obično normalizuju da ublaže efekat dužine dokumenta.

$$tf \cdot \ln \frac{\text{broj_dokumenata}}{df} \quad (3.2)$$

Probabilistički modeli

Verovatnoća obezbeđuje principijelnu osnovu za neizvesno rezonovanje. Prema principu rangiranja verovatnoće, ako se dokumenti posmatraju nezavisno optimalni redosled vraćenih dokumenata je po opadajućoj verovatnoći važnosti [30]. Cilj probabilističkog modela je da izračuna verovatnoću da je dokument (D) relevantan za dati upit (q), tj. $P(D \text{ is relevant} | q)$. Pregled raznih

probabilističkih modela dat je u [31]. Oni se razlikuju po tome kako procenjuju verovatnoću relevantnosti.

Jezički modeli za pronalaženje

Probabilistički jezički modeli su postali vrlo popularni za pronalaženje teksta od 1998. godine [32]. Jedan dokument se predstavlja kao generativni probabilistički model termova. Svaki model dokumenata definiše raspodelu verovatnoće nad termovima u rečniku. Takav model se ocenjuje na osnovu uzorka teksta koji je reprezentativan za taj model. Obično se koriste unigrami umesto reči, odnosno termovi su nezavisni za dati specifični jezički model. Pretpostavlja se da su termovi upiti nezavisni ishodi u modelu dokumenta. Dokumenti se rangiraju prema verovatnoći $P(D|q)$, tj. relevantnosti modela dokumenta D ako je dat upit q . To rangiranje je isto i prema verovatnoći $P(q|D)$ (verovatnoći upita generisanog u modelu dokumenta), na osnovu Bajesove formule

$$P(D|q) = p(q|D) * p(D) / p(q)$$

u ovoj formuli $p(q)$ je konstanta, a *a priori* verovatnoća $p(D)$ se često smatra istom za sve dokumente. Jezički modeli se moraju „ispeglati” da bi se pozitivne verovatnoće mogle dodeliti termovima upita koji se ne pojavljuju u datom dokumentu. Koriste se različite tehnike „peglanja” koje utiču na efektivnost jezičkih modela. Na primer, Berger i Lafferty za peglanje modela koriste prilagođene metode iz statističkog mašinskog prevođenja [33].

Jezički modeli su konceptijski jednostavni i lako razumljivi. Oni su bazirani na čvrstoj teorijskoj osnovi i prirodno koriste standardne statistike, a ne heuristike. Pored toga, moguća je primena najnovijih metoda iz statističke obrade prirodnih jezika. Mada pretpostavka o nezavisnosti termova često nije ispunjena, jezički modeli su se pokazali efektivnim.

3.6.2 Modeli pronalaženja slika na osnovu sadržaja

Za dati primer slike ili skice kao upita, CBIR sistem ekstrahuje ista vizuelna obeležja iz upita koja koristi i za indeksiranje slika u kolekciji. Metode rangiranja sličnosti koriste se za rangiranje vraćenog skupa slika prema njihovoj sličnosti sa slikom upitom. Model vektorskog prostora i probabilistički model iz pronalaženja teksta adaptirani su za korišćenje u CBIR sistemima. Razvijene su i druge specijalne tehnike za pronalaženje slika koje koriste razne reprezentacije obeležja.

Model vektorskog prostora

Model vektorskog prostora se najčešće koristi u CBIR sistemima. Kolekcija slika i upiti su reprezentovani kao vektori obeležja u n -dimenzionalnom vektorskom prostoru. Za razliku od pronalaženja teksta, ovde nije široko zastupljena samo jedna mera sličnosti. Antani je dao pregled pristupa u poklapanju sličnosti slika [34]. Kriterijum za izbor mere sličnosti obuhvata efikasnost izračunavanja i sposobnost zahvatanja sličnosti između slika. Različiti skupovi obeležja mogu da

koriste različite mere sličnosti. Na primer, sličnost obeležja teksture se obično meri koristeći Euclidovu distancu ili Mahalanobisovu distancu (koja inače razmatra korelaciju različitih obeležja). Presek histograma (dat jednačinom 3.3) postao je popularan za merenje sličnosti boja na osnovu histograma boja [35]. Izbor mere sličnosti značajno utiče na performanse pronalaženja kod CBIR sistema. Pošto CBIR sistem može da koristi različite skupove obeležja, biranje pogodnih mere distance za svaki skup obeležja slike i optimalno kombinovanje sličnosti izračunatih iz različitih skupova obeležja slike postaje vrlo bitno.

$$sim(h, g) = \frac{\sum_{j=1}^n \min(h[j], g[j])}{\min \sum_{k=1}^n h[k], \sum_{i=1}^n g[i]} \quad (3.3)$$

gde su h i g histogrami boje sa n kategorija.

Probabilistički model

Ovi modeli su adaptirani za korišćenje u CBIR sistemima. Slika se posmatra kao generativni probabilistički model koji definiše raspodelu verovatnoće nad vizuelnim obeležjima. Prilikom traženja, sistem meri verovatnoću da je slika upita generisana od strane svakog modela. Potrebna je pogodna raspodela verovatnoće da se zahvate glavne karakteristike slike. Vasconcelos je pravio poseban *Gaussian Mixture Model (GMM)* za svaku sliku [35], [36]. Slika je deljena u blokove veličine 8x8 piksela. Deset *DCT* koeficijenata iz svakog kanala boje su obeležja za svaki blok. Podrazumeva se da se svaki blok generiše iz miksa Gaussian izvora, a broj Gaussian komponenti je fiksiran za sve slike u kolekciji. Algoritam maksimizacije očekivanja koristi se za pronalaženje *GMM* parametara. Broj Gaussian komponenti je važan. Premalo komponentata ne može da zahvati (apsorbuje) najvažnije aspekte slike. Suviše komponenti može da izazove prebačaj, tj. ne postoji dovoljno podataka da se proceni povećani broj parametara. Obično se koristi manji broj Gaussian komponenti (između 4 i 32).

Drugi modeli pronalaženja

Za obeležja koja se ne reprezentuju u vektorskom formatu, kao što su deskriptori oblika ili deskriptori prostornog rasporeda, merenje sličnosti je mnogo složenije. Povezano je sa prepoznavanjem objekata u prepoznavanju uzoraka ili kompjuterskoj viziji. Veltkamp i Hagedoom dali su pregled tehnika poklapanja oblika [37].

3.7 Kombinacija teksta i slika

Tekst i slike su posebni modaliteti koji se koriste da se predstave sadržaji u dokumentu. Svaki od njih ima svoje prednosti i nedostatke, što će kasnije biti navedeno. Dok tekst i slike mogu

ponaosob biti dvosmisleni, zajedno se obično dopunjuju. To je zato što pisci, kada pišu tekst, teže da izostave ono što je vizuelno očigledno u slici (recimo boja cveća) a pominju svojstva koja je veoma teško izvesti pomoću vizije (recimo vrste cveća). Ingwersenov kognitivni model pretraživanja informacija, koji predviđa da bi kombinovanje metoda koji koriste različite kognitivne strukture verovatno trebalo da bude efektivnije u pretraživanju nego pojedinačna metoda, obezbeđuje teoretske osnove za kombinovanje teksta i slika.

Mada postoji znatan kvantum objavljenih i aktuelnih istraživanja i u pretraživanju teksta i u pretraživanju slika, ostaje mnogo toga da se uradi da bi se uvidelo koliko efektivno ovi pristupi mogu da dopune jedan drugog, kao i kako bi modaliteti teksta i slika mogli „bezšavno” da se integrišu u jedno okruženje. Kasnije će biti razmotreni multimodalni *Information Retrieval (IR)* sistemi, kao i različite tehnike kombinovanja teksta i slika.

3.7.1 Prednosti i nedostaci modaliteta teksta i slika

Modalitet teksta - Pretraživanje teksta se uspešno koristi u velikim heterogenim kolekcijama dokumenata, kao što su pretraživanje veb stranica i digitalnih biblioteka. Prednosti su da se tekstualni sadržaj iz dokumenata može lako predstaviti pomoću skupa termova i da korisnici mogu da predstavljaju svoje semantičke informacije na prirodan način pomoću teksta.

Predstavljanje teksta skupom termova neizbežno vodi ka izvesnom gubitku semantika u tekstu. Postoje dva fundamentalna problema kada se koristi tekst za pretraživanje:

Polisemija: reči često imaju višestruka značenja i različite tipove korišćenja. Na primer, kada korisnik koristi upit *kosa* misleći na poljoprivrednu alatku, rezultati pretraživanja mogu obuhvatiti nerelevantna dokumenta gde se *kosa* odnosi na deo ljudske glave. Ovaj problem dvosmislenosti postaje još ozbiljniji u heterogenim kolekcijama dokumenata, kao što su veb stranice.

Sinonimija: IR mogu da ne pretraže relevantne dokumente koji sadrže sinonimne termine. Na primer, kada korisnik želi da nađe informacije o kolima, ako upit sadrži samo termin „kola”, relevantni dokumenti koji sadrže termin „automobil” neće biti pretraženi.

Korišćenje teksta za pretraživanje slike ima iste prednosti i nedostatke kao kod pretraživanja teksta. Pretraživanje slika zasnovano na tekstu izloženo je većim ograničenjima od pretraživanja teksta. Postoje određene neizvesnosti u nalaženju kolateralnih blokova teksta. Kolateralni tekst može biti irelevantan i nekompletan. Neke slike možda uopšte nemaju kolateralni tekst. Neke slike su ručno anotirane sa par ključnih reči. Ručna anotacija slika je skupa i subjektivna. Takođe, anotacije normalno ne opisuju vizuelni sadržaj slike, kao što je boja, tekstura ili oblik. Sclarof i još neki autori pokazali su da neke slike ne mogu da se anotiraju zato što je teško opisati rečima njihov vizuelni sadržaj [18].

Modalitet slike - Slika vredi više nego hiljadu reči. Sa jedne strane, ta izreka pokazuje da su slike moćan modalitet za komuniciranje informacija; sa druge strane, ona kaže da su slike složene za opisivanje.

Jedna slika može biti opisana na različite načine, recimo preko svog vizuelnog sadržaja (tj. kako izgleda) ili preko svog semantičkog sadržaja (tj. šta je sve na slici prikazano). Kod pretraživanja slika, različiti korisnici mogu imati različite informacione zahteve kod iste slike. Na primer, stubičasti

dijagram koji pokazuje statistiku populacije može odgovarati informacionim zahtevima korisnika koji traže stubičaste dijagrame na osnovu vizuelnog sadržaja ili zahtevima drugog korisnika koji traži informaciju o statistici populacije na osnovu semantičkog sadržaja.

CBIR omogućava korisnicima da pretražuju slike na osnovu vizuelnog sadržaja slike, kao što je boja, tekstura i oblik. Snaga CBIR sistema je u tome što indeksirana obeležja mogu biti nezavisna od domena, tj. obeležja boje, oblika i teksture nisu ograničena na konkretan domen slike. Osim toga, automatsko indeksiranje kojim raspolažu ovi sistemi je efikasno i bez čovekove subjektivnosti.

CBIR ima ograničenja. Ono se ne koristi toliko široko kao pretraživanje teksta. Jedan od fundamentalnih problema je semantički jaz, tj. ne postojanje veze od zahteva u obliku semantičke informacije do vizuelnog sadržaja [38]. Rezultati pretraživanja zasnovani na sličnostima pukog vizuelnog sadržaja ne poseduju obavezno semantička značenja koja su interesantna za korisnika. Na primer, upit koji koristi primerak slike zelene jabuke na osnovu obeležja boje može vratiti sliku trave, jabuke ili zeleno obojenog zida. Procesi grupisanja obeležja slike u značajne objekte i pridruživanja njima semantičkog opisa su teški. U dosadašnjoj praksi, semantički sadržaji slika se dobijaju iz kolateralnog teksta.

Kod CBIR postoji tzv. problem „nulte stranice” [18]. Ako korisnik nema na raspolaganju primerak slike, biće teško da započne pretraživanje. Kod nekih sistema dizajniran je složen interfejs za upit koji omogućava korisniku da skicira upit (sliku). Međutim, složen interfejs za upit može da stvori otpor kod korisnika.

Multimodalni IR mogu da kombinuju dobre strane iz oba modaliteta i prevaziđu njihova ograničenja. Korišćenje slika može da pomogne u prevazilaženju dvosmislenosti kod pretraživanja teksta. Korišćenje teksta može da pomogne da se suzi semantički jaz kod pretraživanja slika.

3.8 Pregled multimodalnih IR sistema

Multimodalni IR sistemi razlikuju se po složenosti modela dokumenta koji koriste i metodama kombinovanja. U tabeli 3.1 dat je pregled multimodalnih IR sistema. Prva kolona pokazuje referencu i ime eksperimentalnog sistema ako je na raspolaganju.

Ovde su multimodalni sistemi klasifikovani pomoću sledećeg kriterijuma:

Složenost modela dokumenta (druga kolona u tabeli)

Većina modela dokumenta u razmatranim multimodalnim IR sistemima su jednostavni, kod kojih je dokument slika anotirana sa ključnim rečima. Postoji par složenih modela multimodalnog dokumenta [39], [40]. Ovi modeli su slični sa modelom dokumenta prezentiranim na slici 3.2. Dokumenti sadrže i tekst i slike, a tekst obuhvata nezavisni tekst i kolateralni tekst.

Tehnike kombinovanja teksta i slike (treća kolona)

Tehnike kombinovanja teksta i slike razmatraju se kasnije. U principu postoje tri pristupa:

Modaliteti teksta i slike se sekvencijalno koriste.

Modaliteti teksta i slike se simultano koriste, ili linearno ili nelinearno kombinovani.

Probabilističke metode se koriste da integrišu tekst i slike.

Da li se koristi povratna informacija o značajnosti i kako se koristi (četvrta kolona)

Korisnikova povratna informacija o značajnosti je važna tehnika za poboljšanje efektivnosti

IR sistema. Neki sistemi se oslanjaju na korisnikovu povratnu informaciju o značajnosti da kombinuju tekst i sliku, dok drugi sistemi ne poseduju tu funkciju.

Obeležja slike korišćena u sistemima (peta kolona)

Različita obeležja slike se indeksiraju u razmatranim sistemima, što je navedeno u tabeli 3.1. Tekstualna obeležja su slična u svim sistemima; ona koriste skup ključnih reči i većina sistema koristi $tf*idf$ ponderisanje. Glavna razlika između sistema je u stepenu korišćenja procesiranja prirodnog jezika. Na primer, kod nekih sistema tekstualni blokovi se dalje procesiraju sa tzv. POS (engl. *part-of-speech*) tagiranjem (određivanje vrsta reči) da se dobiju imenice.

Tabela 3.1. Pregled multimodalnih IR sistema

Referenca/ sistem	Model dokumenta	Kombinacija teksta i slike	RF	Svojstva slike
	Slika sa ključnim rečima.	Probabilistički model (hijerarhijski).	Ne	Svojstva iz segmentiranih regiona (boja, energija orijentacije, veličina regiona, lokacija, konveksnost, prvi momenat).
	Slika sa ključnim rečima.	Probabilistički model (LDA).	Ne	Svojstva iz segmentiranih regiona (veličina, pložaj, boja, tekstura i oblik).
DocBrowser	Faks dokumenta.	Sekvencijalna.	Ne	Razna vizuelna obeležja od logoa, potpisa itd.
iFind	Veb slike sa kolateralnim tekstom.	Linearna kombinacija.	Da. Preformulisanje upita i učenje semantičke mreže	Globalna svojstva. Histogram boje; momenti boje, koherentnost boje, wavelet tekstura, Tamura tekstura.
	Slike sa ključnim rečima.	Probabilistički model.	Ne	Svojstva iz segmentiranih regiona.
	Strukturirani dokumenti koji sadrže tekst i slike.	Nelinearna kombinacija. Fazi logičko rezonovanje.	Ne	Oblik ili boja.
	Slike sa ključnim rečima.	Linearna kombinacija.	Da. Preponderisanje.	Histogram boje, svojstva teksture, kao pravilnost, zrnatost i histogrami orijentacije
WEBMARS	HTML dokumenti.	Linearna kombinacija.	Da. Preformulisanje upita.	Višestruki skupovi svojstava uključujući boju, teksturu i prostorni raspored.
ImageRover	Veb slike sa kolateralnim tekstom.	Linearna kombinacija sa semantičkim indeksiranjem (LSI) primenjenim na tekst.	Da. Preponderisanje.	Histogram boje i histogram dominantne orijentacije.
VisualSeek, WebSeek	Veb slike sa kolateralnim tekstom.	Sekvencijalno.	Da. Preformulisanje upita.	Histogram boje.
MMIR	Veb slike sa kolateralnim tekstom.	Linearna kombinacija.	Ne. Off-line učenje.	Tehnike detekcije lica, histogram boje.
	Veb slike sa kolateralnim tekstom (samo naslovi veb stranica).	Latentno semantičko indeksiranje primenjeno na ujedinjeni vektor slike i svojstva teksta.	Ne	Histogram boje u HSV prostoru boja i anglogrami boje.
	Slike sa ključnim rečima.	Linearna kombinacija.	Da. Asocijacija reči sa RF.	Momenti boje, wevelet momenti i svojstva koja se oslanjanju na detekciju ivica.

3.9 Tehnike kombinovanja

Glavnina istraživanja o kombinovanju teksta i slika izvršena je u zajednici koja se bavi istraživanjima pretraživanja slika, gde se kolateralni tekst kombinuje sa slikama da se poboljša efektivnost pretraživanja slika. Tema istraživanja u razmatranim naučnim radovima je otkrivanje relacija između kolateralnog teksta i slika. Na osnovu relacija između modaliteta teksta i slike, kolateralni tekst može se koristiti da se nađu semantički koncepti slike i prosledi semantičke koncepte neobebeženim (neanotiranim) slikama koristeći oba modaliteta. Tekst i slika mogu biti implicitno pridruženi koristeći latentno semantičko indeksiranje (objašnjeno kasnije). Relacije između teksta i slika mogu se otkriti kreiranjem semantičkih mreža (takođe opisano kasnije).

U zajednici koja se bavi pretraživanjem teksta, urađeno je malo istraživanja u smislu korišćenja slika da bi se poboljšalo pretraživanje teksta. Ako relevantna slika postoji u multimodalnom dokumentu, ona može dosta doprineti finalnom rangiranju dokumenta. Paek i Smith istraživali su neke stvari u vezi sa tim pitanjem [41]. Jedna stvar je procenjivanje da li je slika povezana sa tekstualnim kontekstom stranice. Slike su klasifikovane u sadržajne slike (povezane sa tekstom stranice) i ne-sadržajne slike (takve kao oglasni baneri i logo slike). Sadržajne slike se koriste u rezimeu dokumenta.

Posle tabelarnog pregleda multimodalnih sistema sledi pregled različitih tehnika kombinovanja kolateralnog teksta i slika.

3.9.1 Sekvencijalno korišćenje modaliteta teksta i slike

Modaliteti teksta i slike mogu da se koriste sekvencijalno i samo jedan modalitet se koristi u svakom upitu. U suštini, modaliteti teksta i slike u ovom pristupu nisu integrisani. Oni se indeksiraju i pretražuju odvojeno. Većina IR sistema, uključujući sisteme za pretraživanje skeniranih dokumenata koji omogućavaju korisnicima da pretražuju jedan ili drugi modalitet koriste ovu vrstu pristupa. Jedan od ovih modaliteta može se koristiti kao prvi u zavisnosti od uloge ta dva modaliteta.

Pretraživanje slika na osnovu teksta može se prvo koristiti da bi se obezbedio inicijalni skup slika da bi se startovao CBIR. CBIR se koristi da prečisti rezultate pretraživanja slika na osnovu teksta. Modalitet teksta može se koristiti da se kategorizuju dokumenti u semantičke kategorije. Kod sistema **WebSeek** kolateralni tekst se koristi da se obavi polu-automatska klasifikacija slika u taksonomiju semantičkih kategorija [42]. Korisnik može da prelistava skup slika unutar kategorije. Podudaranje na osnovu sličnosti histograma boje se koristi za nalaženje slika sa sličnim obeležjima boje unutar kategorije ili u kompletnom katalogu. Može se prvo uraditi CBIR a zatim se modalitet teksta koristi da se prečiste rezultati upita ili obezbedi bolje prelistavanje rezultata upita [17].

Pretraživanje i teksta i slika je obezbeđeno u nekim sistemima za pretraživanje skeniranih dokumenata. Na primer, **DocBrowser** pretražuje faksimile [43]. Vizuelni sadržaj obuhvata logo kompanije na faksu, ručne potpise, cele stranice dokumenta i reči neidentifikovane od strane OCR modula. Tekstualni sadržaj je rezultat OCR modula.

U specijalnim slučajevima, modalitet teksta može pomoći indeksiranje slika. Kod **PICTION** sistema [44], prvo se analizira potpis slike da se identifikuje očekivani broj lica i njihove očekivane relativne pozicije. Zatim se detektor lica primenjuje na ograničeni deo slike. Slično, kod projekta **MARIE**, potpisi slika se analiziraju pomoću tehnika prirodnog jezika da pomognu identifikovanje oblika u slikama [45], [46].

3.9.2 Simultano korišćenje oba modaliteta

Kad se postavlja upit mogu se koristiti oba modaliteta. Kompozitni upit se formira da obuhvata i tekst i sliku. Ako primerak slike nije obezbeđen, neki sistemi koriste slučajni vektor kao vektor vizuelnog obeležja kako bi omogućili korisniku da istraži različite delove prostora obeležja [47]. U toku pretraživanja, sistem kombinuje sličnosti izračunate iz teksta i slike. Tekst i slike mogu da se indeksiraju odvojeno. Postoje razne metode kombinovanja. One su grupisane u dve opšte kategorije: linearna i nelinearna kombinacija.

Linearna kombinacija

Linearna kombinacija je jednostavna metoda i najčešće se koristi. Kompozitni upit q ima dva dela: tekstualni upit q_t i sliku upita q_m . Sličnost između upita q i multimodalnog dokumenta D_p , koji takođe ima dva dela, tekst D_t i slike D_m , definiše se na sledeći način:

$$\text{sim}(q, D_t) = a \text{sim}_t(q_t, D_t) + (1-a) \text{sim}_m(q_m, D_m) \quad (3.4)$$

Kombinovana sličnost je ponderisana suma sličnosti izračunatih iz oba modaliteta. Faktor a je inter-modalni faktor. Intra-modalni faktori su faktori unutar prostora obeležja teksta i slike. Efektivnost linearne kombinacije zavisi od izbora faktora i drugih stvari. Različiti sistemi razlikuju se u izračunavanju mera sličnosti za tekst i slike i faktoru a za kombinovanje teksta i slike.

Postoje tri načina da se postavi inter-modalni faktor. (1) Faktor se može postaviti ručno prema prethodnom znanju. Na primer, dvama modalitetima su dati jednaki faktori za linearno kombinovanje sličnosti iz tekstualnih obeležja (koristeći dot (skalarne) proizvode) i vizuelnih obeležja (koristeći Euklidove distance) u sistemu **iFind** [48]. (2) Faktori se mogu naučiti „off-line”. Na primer, da bi se pronašao optimalan skup faktora za više modaliteta, koristi se faza obuke da se nauči grupa optimalnih faktora za odabrani skup reprezentativnih upita u **MMIR** sistemu [49]. U fazi pretraživanja, pojedinačni modeli za različite modalitete se linearno kombinuju koristeći skup faktora reprezentativnog upita koji je najbliži tekstu koji je korisnik postavio. (3) Faktor se može „on-line” podesiti na osnovu korisnikove povratne informacije [39], [47]. Metode za postavljanje intra-modalnih faktora su iste kao metode podešavanja faktora koji se koriste kod pretraživanja teksta ili kod CBIR sistema.

Nelinearna kombinacija

Nelinearna kombinacija se koristi u nekim sistemima. U ovom odeljku će biti navedene neke reprezentativne tehnike nelinearne kombinacije.

Fazi logika

Neki sistemi koriste fazi logičko rezonovanje da kombinuju više modaliteta [40], [50]. Meghini i drugi kombinovali su obeležja koja se odnose na tekst, slike i strukturu dokumenta [40]. Sadržaji dokumenta i upiti opisani su pomoću logičkog jezika, Description Logic [51]. Autori tvrde da Description Logic jezik ima objektno-orijentisane karakteristike koje ga čine posebno pogodnim za rezonovanje o hijerarhijama strukturisanih objekata.

Iterativno pojačanje

Iterativni pristup se koristi da se istraži uzajamno pojačanje između slika i njihovog kolateralnog teksta [52]. Prvo, matrica sličnosti blokova teksta i matrica sličnosti slika se kreiraju iz matrice dokument-term i matrice slika-obeležje, respektivno. Iterativno pojačanje se zasniva na jednačini 3.5, adaptiranoj iz pristupa koji su predložili Kandola i drugi [53]. Sličnost jednog modaliteta se prosleđuje drugom modalitetu. Pojačanje može da poboljša ili smanji sličnost dva objekta. I jedno-modalno pronalaženje i linearna kombinacija više modaliteta su specijalni slučajevi jednačine 3.5.

Ako je $a = 1$ ili $b = 1$, to je jednomodalno pronalaženje. Ako je $l = 1$, to je linearna kombinacija. Kada je l postavljeno da bude manje od 1, prosleđena sličnost je slabija od originalne sličnosti. U eksperimentima [52], iterativno pojačanje je pokazalo bolje rezultate sa linearnom kombinacijom u odnosu na jednomodalno pronalaženje korišćenih ili tekstualnih ili vizuelnih obeležja.

$$\begin{cases} \hat{K} = a K + (1-a) l Z \hat{G} Z^T \\ \hat{G} = b G + (1-b) l Z^T \hat{K} Z \end{cases} \quad (3.5)$$

gde su

K ($M \times M$) matrica sličnosti M slika,

G ($N \times N$) matrica sličnosti N blokova teksta,

Z ($M \times N$) matrica veze. Veza postoji ako je slika povezana sa blokom teksta.

3.10 Tehnike za poboljšanje efektivnosti IR

Različite tehnike su razvijene da se poboljša efektivnost i efikasnost IR sistema. Ovde ćemo ukratko opisati nekoliko reprezentativnih tehnika koje se koriste u multimodalnim IR, uključujući latentno semantičko indeksiranje, semantičke mreže, korisnikovu reakciju i klasifikaciju/klasterizaciju dokumenata.

3.10.1 Latentno semantičko indeksiranje

Latentno semantičko indeksiranje (LSI) se izvorno koristi da razreši probleme sinonimije i povezuje dokumente koji koriste različite termine [54]. Generalna ideja je da se mapiraju dokumenti i termini u predstavu niske dimenzionalnosti. Mapiranje se vrši primenom SVD dekompozicije (engl. *Singular Value Decomposition*) na matricu dokument-term i selektujući samo nekoliko prvih (obično 100 – 300) karakterističnih vektora sa najvišim karakterističnim vrednostima.

Ovo osigurava da prostor niske dimenzionalnosti (naziva se latentni semantički prostor ili prostor koncepta) reflektuje semantičke asocijacije, pošto se slični termini mapiraju na slične lokacije u latentnom semantičkom prostoru. Upit q se takođe mapira u ovaj prostor. Sličnost između upita i dokumenta se izračunava na osnovu unutrašnjeg proizvoda u ovom semantičkom prostoru i invarijantna je prema izboru reči za slične koncepte.

LSI pokušava da pronađe dokumente na osnovu koncepta u dokumentima, umesto direktnog poklapanja ključnih reči koje se koristi kod modela klasičnog pronalaženja teksta. LSI obično poboljšava odziv a takođe poboljšava i preciznost. Ima ograničenja. Ose u latentnom semantičkom prostoru nisu ljudima rezumljive. SVD algoritam je zahtevan i kompleksnost je $O(N^2K^3)$. N je broj termova plus dokumenti a K je broj dimenzija u latentnom semantičkom prostoru. Određivanje optimalnog broja dimenzija (K) nije trivijalno.

LSI u multimodalnim IR

Latentno semantičko indeksiranje prilagođeno je za multimodalne IR. Postoje dva pristupa: LSI primenjeno samo na tekst i LSI primenjeno na objedinjeni vektor obeležja koji sadrži i obeležja teksta i obeležja slike.

LSI primenjeno na tekst

Kod ovog pristupa, LSI se prvo primenjuje na tekst. Zatim se sličnost izračunava na osnovu vektora obeležja teksta linearno kombinuje sa sličnošću izračunatom od vektora obeležja slike. Ovaj pristup koristi ImageRover da apsorbuje iz HTML dokumenata statistike teksta koje su relevantne za slike [18]. Prema eksperimentalnim izveštajima vidi se da se maksimalne performanse pronalaženja postižu kada se koriste i vizuelni i tekstualni sadržaj. Ponekad, performanse su slabe kada nema korelacije između slike i pratećeg teksta (slučaj banera i logo znakova na veb stranici). Pošto se LSI primenjuje samo na tekst, ovaj pristup ne može da apsorbuje odnose istovremenog pojavljivanja između obeležja slike i ključnih reči teksta.

LSI primenjeno i na tekst i na slike

Kod ovog pristupa, prvo se kombinuju vektori obeležja teksta i slike, zatim se LSI primenjuje na objedinjeni vektor obeležja [16], [55]. Vektori obeležja slike različito se tretiraju u različitim sistemima. Realni vektori obeležja slike dobijeni iz statistika histograma boja mogu se direktno koristiti [16]. Dimenzija vektora obeležja slike je 100. Pošto se koristi mali skup podataka, dimenzija vektora obeležja slike je samo 43. Ovo je neobično; obično dimenzija tekst prostora je par hiljada. Obeležja boje i teksture mogu se kvantizovati [55]. Cilj kvantizacije je da se dobije diskretni skup desetina hiljada vizuelnih „termova” koji se mogu pojaviti (jednom ili više puta) ili ne pojaviti u dokumentu. Tehnika kvantizacije koju je predložio Squire može se koristiti da se obezbedi da dobijeni termini obeležja slike imaju sličan tip i raspodelu kao tekstualni termini [56].

Primenjivanjem LSI na objedinjene vektore obeležja mogu se implicitno asociirati neka obeležja slike sa skupom istovremeno prisutnih ključnih reči. Međutim, ima ograničenja. Teško je

kombinovati više skupova obeležja u jedan vektor obeležja pošto različiti tipovi obeležja mogu zahtevati različite metrike sličnosti.

3.10.2 Semantičke mreže

Semantičke mreže su prvo korišćene u pronalaženju teksta kako bi se razrešio problem sinonimije. Semantičke mreže pokazuju termine i njihove relacije. Upit može da se proširi pomoću reči koje su povezane sa terminima upita. Semantička mreža može biti tezaurus opšte namene, kao što je WordNet [57]. Međutim, pokazalo se da proširivanje upita automatski selektovanim sinonimima iz unapred napravljenog tezaurusa daje slabe rezultate na velikoj, heterogenoj kolekciji dokumenata [58]. To se dešava zato što relacije apsorbirane u takvom tezaurusu možda nisu valjane u lokalnom kontekstu datog korisnikovog upita. Semantička mreža može se kreirati preko korisnikove reakcije. Statistička analiza o pojavljivanju i istovremenom pojavljivanju terma u skupu relevantnih dokumenata vrši se tako da automatski konstruiše tezaurus [59].

Semantičke mreže koje koriste modalitete teksta i slike

Kod multimodalnih IR, semantičke mreže mogu da se kreiraju koristeći modalitete i teksta i slike. Bazirajući se na pretpostavci da semantički povezane slike mogu biti vizuelno slične i obrnuto, semantičke mreže kod multimodalnih IR povezuju ključne reči sa slikama ili regionima slika. Relacije između ključnih reči se kreiraju korišćenjem kombinacije obeležja teksta i slike.

Veruje se da semantičke mreže igraju važniju ulogu u multimodalnom IR nego u sistemu za pronalaženje teksta. Semantičke mreže mogu da se koriste da se proširi upit na različite načine: proširivanje tekstualnog upita koristeći povezane ključne reči, proširivanje tekstualnog upita koristeći vizuelni sadržaj povezan sa ključnim rečima i proširivanje upita zasnovanog na primeru slike koristeći tekst povezan sa slikama. Pored proširivanja upita, semantičke mreže imaju i druge primene. Semantičke mreže mogu da pomognu u sužavanju semantičkog jaza. Takođe, one mogu da pomognu u kros-modalnom pronalaženju, tj. korišćenje teksta za pronalaženje slika bez anotacija.

Razlikuju se dva tipa semantičkih mreža, flat i hijerarhijske. Flat semantičke mreže pretpostavljaju jednostavnu relaciju između slika i ključnih reči, tj. sve ključne reči su direktno pridružene slikama ili vizuelnim obeležjima slika. Ne postoje hijerarhijske ili druge relacije između semantičkih ključnih reči. U hijerarhijskim semantičkim mrežama, ključne reči na višem nivou nisu direktno povezane sa slikama, već sa ključnim rečima nižeg nivoa. Kod građenja prvih semantičkih mreža oslonac je bila umnogome korisnikova interakcija i ručni rad. Na primer, kod sistema Chabot korisnici mogu da definišu koncepte preko interfejsa [60]. Koncepti su pridruženi kombinaciji vizuelnih i tekstualnih predikata da definišu njihove vizuelne reprezentacije (tj. koncept žuti cvet je pridružen ključnoj reči cvet i žutoj boji). Da bi kreirali semantičko-vizuelni tezaurus, korisnici moraju ručno da označavaju regione slike sa vidljivim ključnim rečima [61].

3.11 Korisnikova povratna informacija o značajnosti

3.11.1 Uvod u povratnu informaciju o značajnosti

Povratna informacija o značajnosti (engl. *relevance feedback*) je dobro poznata tehnika za poboljšanje efektivnosti pretraživanja teksta [62]. Tom tehnikom uklanja se usko grlo koje može da se pojavi usled formulisanja nedovoljno kvalitetnog inicijalnog upita od strane korisnika. Sistem formuliše bolji upit na osnovu korisnikove povratne informacije posle inicijalnog upita. Povratna informacija o značajnosti je važnija istraživačka tema za CBIR sistem nego za sistem za pretraživanje teksta. Razlog za to leži u činjenici da je lakše korisniku da proceni da li su ponuđene slike značajne nego u slučaju tekstualnih dokumenata zato što slike otkrivaju korisniku svoj sadržaj neposredno. Drugi razlog leži u semantičkom jazu koji je izraženiji kod CBIR sistema usled čega je povratna informacija o značajnosti neophodnija. Pregled tehnika interakcije u pronalaženju slika dat je u [63], [64].

Osnovna ideja povratne informacije o značajnosti kod pronalaženja teksta ogleda se u prečišćavanju originalnog upita dodavanjem novih termova iz značajnih dokumenata (tj. proširivanje upita (engl. *query expansion*)) i pojačavanjem važnosti termova upita koji su se pojavili u značajnim dokumentima (tj. preponderisanje termova (engl. *term re-weighting*)). Prečišćavanje upita se radi po osnovi korisnika. Jedan od najpoznatijih pristupa prečišćavanju upita je Rocchio algoritam [65]. Vektor modifikovanog upita dat je sledećom jednačinom:

$$\vec{q}_m = a \vec{q}_0 + b \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - g \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \quad (3.6)$$

gde je:

$\vec{q}_m$ – vektor modifikovanog upita;

$\vec{q}_0$ – vektor originalnog upita;

a, b, g – težinski faktori (ponderi);

D_r – skup vektora poznatih značajnih dokumenata;

D_{nr} – skup vektora poznatih neznačajnih dokumenata.

Ovaj algoritam pomaže da se novi upit približava značajnim dokumentima a udaljava od neznačajnih dokumenata. Rocchio algoritam je prilagođen za primenu u CBIR i multimodalnim sistemima.

Povratna informacija o značajnosti može se posmatrati i kao problem binarne klasifikacije [64]. Značajni dokumenti se označavaju kao pozitivni a neznačajni dokumenti kao negativni. Cilj je da klasifikator nauči da bolje klasifikuje dokumente koji će se pretraživati. Kod ovog zadatka klasifikacije postoji niz problema a ističe se problem malog obučavajućeg skupa u odnosu na prostor obeležja visoke dimenzionalnosti. Inače, ova problematika se aktivno istražuje u CBIR zajednici.

3.11.2 Povratna informacija o značajnosti kod multimodalnog pronalaženja

Korišćenje povratne informacije o značajnosti kod multimodalnih IR je slično korišćenju kod pronalaženja teksta ili pronalaženja slika. Razlika je u tome što se koriste oba modaliteta. Na primer, Lu i saradnici su modifikovali Rocchio algoritam da uključi i tekstualni i slikovni sadržaj [66]. Chen i saradnici koristili su povratnu informaciju o značajnosti da podese težinske faktore intra-modaliteta i inter-modaliteta [48]. Oni su izvršili eksperimente na slikama prikupljenim sa različitih veb sajtova sa različitim sadržajima i uočili su da pronalaženje slika na osnovu sadržaja korišćenjem obeležja niskog nivoa daje relativno dobre rezultate kod nekih kategorija kao što su *sun* i *watterfall*. Međutim, kod drugih kategorija kao što su *Clinton* i *summer*, rezultati nisu tako dobri pošto su te reči apstraktnije. Različiti težinski faktori su dati modalitetima teksta i slike uzimajući u obzir različite upite. Težinski faktori su takođe podešavani među rečima kod obeležja teksta. Sistem *ImageRover* koristi korisnikovu povratnu informaciju da selektuje odgovarajuće metrike sličnosti za različite tipove obeležja slike i podesi težinske faktore za različite modalitete [18].

3.12 Grupisanje i klasifikacija dokumenata

3.12.1 Uvod u klasifikaciju i grupisanje dokumenata

Grupisanje (engl. *clustering*) i klasifikacija dokumenata su povezani sa IR. Oni su korisni za organizovanje i pregledanje dokumenata u IR sistemima. Grupisanje i klasifikovanje dokumenata deli većinu karakteristika sa IR [67]. Veruje se da su tehnike korišćene u multimodalnim IR i u multimodalnom grupisanju/klasifikovanju dokumenata prilično povezane i da se međusobno potpomažu.

Klasifikacija dokumenta (ili nadzirana kategorizacija) je nadzirano učenje kod koga je definisan skup klasa a obučavajući dokumenti su označeni za svaku klasu. Postoje razni klasifikatori, takvi kao što je KNN (k-Nearest Neighbour), naivni Bajes i SVM (Support Vector Machine). Sebastian je dao opsežan pregled kategorizacije teksta [67]. U [68] dat je pregled klasifikacije slika dokumenata na osnovu obeležja slike, obeležja rasporeda elemenata dokumenta, obeležja teksta i kombinacije raznih obeležja. Klasifikacija slika se koristi da se dodele slikama semantički koncepti. Anotacija slika može se posmatrati kao zadatak klasifikacije slika. Razlika se ogleda u tome što više labela može da se dodeli slici ili regionima slike [69] .

Grupisanje dokumenata je nenadzirano učenje, tj. nisu potrebni označeni obučavajući uzorci. Ova tehnika vrši grupisanje dokumenata u skupove sličnih objekata. U literaturi [70] navedeni su brojni algoritmi za grupisanje dokumenata. Ovi algoritmi mogu da se podele u dve grupe, u one koji stvaraju hijerarhijske grupe, kao što je HAC (*Hierarchical Agglomerative Clustering*), i one koji stvaraju ravne particije, kao što je *K-means* grupisanje i SOM (*Self-Organizing Map*) algoritam.

Grupisanje i klasifikacija dokumenata su velike istraživačke oblasti. Dosta se koriste kod jedno-modalnih dokumenata, bilo teksta ili slika.

Ovde će biti dat kratak pregled tehnika grupisanja i klasifikovanja koje kombinuju tekst i

slike. Slično kao kod multimodalnih IR, utvrđeno je da su rezultati grupisanja ili klasifikovanja kombinovanjem teksta i slika bolji nego rezultati kada se koristi samo tekst ili samo slike [71]. Ono što se može zapaziti je da ne postoji mnogo objavljenih radova o istraživanju u kombinovanju teksta i slika za grupisanje i klasifikovanje.

3.12.2 Multimodalna klasifikacija

Nesumnjivo da je istraživački izazov postići automatsko dodeljivanje semantičkih labela slikama. Učinjeni su određeni pokušaji u korišćenju teksta i slika za klasifikovanje slika [71]. U postojećim sistemima za klasifikaciju slika obično je definisano samo par semantičkih klasa. Na primer, sistem koji su predložili Paek i drugi, odvaja slike u dva glavna semantička koncepta, unutrašnji (engl. *indoor*) i spoljašnji (engl. *outdoor*) [71]. Obeležja teksta su termini ekstrahovani iz potpisa slika. Ponderisana suma se koristi kod kombinovanja rezultata izračunatih iz obeležja teksta i slike. WebSeer klasifikuje slike u fotografije, portrete i računarski generisane crteže koristeći i sadržaj slike i kolateralni tekst iz veb stranica [15]. Očigledan izuzetak je sistem klasifikovanja kod WebSeek [42]. Definisana je prilagođena semantička ontologija za opštu kolekciju veb slika (oko 650000) i više od 2000 semantičkih klasa je definisano polu-automatski.

Modaliteti teksta i slike mogu se u sistemu klasifikovanja primeniti odvojeno umesto istovremeno. Na primer, kod šeme hijerarhijske klasifikacije koju su predložili Lu i Drew [72], obeležja teksta i slike koriste se sekvencijalno na različitim nivoima klasifikacije.

3.12.3 Multimodalno grupisanje

Modaliteti teksta i slike se koriste sekvencijalno ili istovremeno u sistemima grupisanja multimodalnih dokumenata. Primer sekvencijalnog korišćenja više modaliteta je Scatter/Gather sistem [73]. Sistem pomaže korisniku da progresivno sužava kolekciju na mali broj interesantnih elemenata, slično Scatter/Gather paradigmi za pregledanje tekstualnih dokumenata [74]. Scatter/Gather iterativno prečišćava pretragu rasturajući kolekciju u mali broj grupa (klastera), a zatim korisnik prikuplja interesantne grupa za ponovno rasturanje. Skup obeležja, iz različitih modaliteta izračunava se za svaki dokument i smešta kao vektor. Obeležja teksta obuhvataju reči teksta koji okružuje sliku i reči pridružene svakoj slici, URL slike, sve tagove i tekst hiperlinka. Obeležja slike obuhvataju histogram boja i meru kompleksnosti boja. Dokumenti se inicijalno grupišu na osnovu obeležja teksta. Grupisanje se vrši korišćenjem standardnog K-means algoritma sa unapred postavljenim brojem klastera. Zatim se klasteri selektovani od strane korisnika re-klasteruju na osnovu obeležja slika.

Kod nekih sistema, modaliteti teksta i slike se koriste istovremeno. Na primer, obeležja teksta i obeležja (niskog nivoa) slike se integrišu u vektor obeležja za grupisanje slika [75]. Svaki klaster se smatra perceptualnim konceptom. Pokazano je da su oba deskriptora obeležja (tekst i vizuelni) korisni u ekstrahovanju perceptualnog znanja iz anotiranih slika. U radu [76] predložen je sistem hijerarhijskog klasterovanja. On je izveden iz Hofmannovog *Hierarchical Aspect Models* za tekst [77] i koristi obeležja teksta i obeležja slike. Kod ovog sistema, slike sa pridruženom reči cveće (flowers) mogu se razvrstati u pridružene klastera koji imaju pretežno crveno cveće (red flowers), pretežno žuto cveće (yellow flowers) i pretežno scene zelene bašte (green garden scenes).

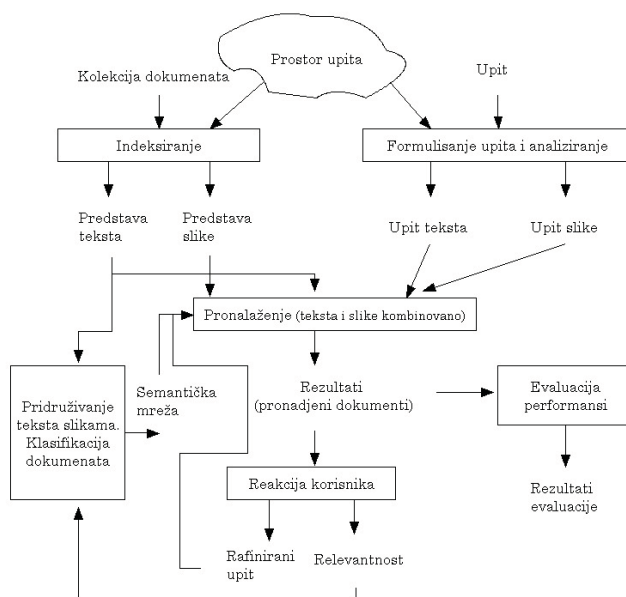
3.13 Rezime

Multimodalni sistemi za pronalaženje informacija bazirani na jedno-modalnim sistemima za pronalaženje informacija, uključujući pronalaženje teksta, pronalaženje slika na osnovu sadržaja, pronalaženje slika na osnovu teksta i pronalaženje slika dokumenta.

Pronalaženje zasnovano na modalitetu teksta je dobro istražena oblast i tehnike iz te oblasti prihvaćene su i prilagođene za korišćenje u drugim IR sistemima.

Pronalaženje zasnovano na modalitetu slike je složenije od pronalaženja zasnovanog na modalitetu teksta. Slika se može opisati slojevito i na više načina koristeći različite skupove obeležja. Evidentno je da postoji semantički jaz između korisnikovih informacionih zahteva i vizuelnog sadržaja slike.

Dijagram multimodalnog IR sistema prikazan je na slici 3.6. Slika sumira različite aspekte ovog tipa sistema. Multimodalni IR sistem indeksira i tekstualni i vizuelni sadržaj. Prostor upita reprezentuje skup mogućih upita. On može biti Boolova kombinacija reči i fraza, recimo „sunce AND ptice” ili može biti slika, na primer upit „pronađi sliku koja je slična zadatoj”, ili upit može biti formulisan kao kombinovani upit, koji sadrži i tekst i sliku. Indeksiranje zavisi od tipa upita koji se očekuju. Formulacija upita takođe zavisi od tipa upita. Kod pretraživanja različite tehnike se koriste za kombinovanje rezultata tekstualnog i slikovnog modaliteta. Multimodalni IR pokušavaju da kombinuju prednosti tekstualnog i slikovnog modaliteta i prevaziđu ograničenja oba modaliteta. Važno je pronaći relaciju između ta dva modaliteta. *Ad-hoc* eksperimenti su pokazali da jedan modalitet može doprineti da drugi modalitet bude informativniji i precizniji a kombinacija dva modaliteta obično poboljšava performanse.



Slika 3.6 Dijagram multimodalnog IR sistema

4. Relevantnost

Danas već postoje brojni sistemi za pronalaženje informacija koji pokušavaju naći pravi put do prave informacije. No čak i oni često dolaze do neispravnih i necelovitih informacija. Tako se kao jedan od glavnih problema nametnuo problem značajnosti (relevantnosti). Kako u tom moru informacija pronaći onu nama bitnu, relevantnu informaciju, postalo je ključno pitanje u oblasti informatičkih nauka. Pokazalo se da nije samo teško pronaći relevantnu informaciju, nego je problem i odrediti i šta je to relevantno. Kojoj od pronađenih informacija možemo dati taj pridev, zašto i kome je ona uistinu relevantna, samo su neka od pitanja koja su brojne naučnike navela da iznesu svoje teorije i gledišta na ovaj problem.

4.1 Osvrt na početak

Dokumentacija i pronalaženje a potom i skladištenje i pronalaženje informacija (eng. information storage and retrieval) imali su nesumnjivo uticaj na definisanje predmeta i područja informacijske nauke, ili su se čak „transformisali u informacijsku nauku” (J. S. Shera, D. B. Cleveland). Dokumentalistika ima svoju predistoriju i računa se da traje od 1895. kad su P. Otlet i H. La Fontaine osnovali *Institut international de bibliographie*, čiji je zadatak bio da obrađuje zapisano ljudsko znanje iz svih vrsta dokumenata pa i onih neverbalnih kao što su mape, slike, dijagrami. Tada se počela naglašavati obrada sadržaja dokumenta bez obzira na oblik prikazivanja, pa se time i prekinula praksa nužnog linearnog raspoređivanja dokumenata na police. Napravljen je novi pomak u predmetu proučavanja i obrade dokumenata, i to od prvobitnog oblika (knjige, što je predmet bibliotekarske delatnosti) na sadržaj dokumenata; nova je disciplina, dokumentalistika, definisana kao postupak skupljanja, klasifikovanja i distribucije svih dokumenata, svih vrsta i svih delatnosti. Takvo određenje dokumentalistike zadržalo se u Evropi i nakon II svetskog rata, dok su se u SAD počeli koristiti novi izrazi pronalaženje informacija te pohranjivanje i pronalaženje informacija, da bi 1961. na *Konferenciji o obrazovanju kadrova za informacijsku nauku* na *Georgia Institute of Technology* u Atlanti, definitivno bio odbačen i naziv dokumentalistika, a umesto dotadašnjih preuzet je novi izraz informacijska nauka (engl. *Information Science*).

Dokumentalistika se zasniva na mogućnosti obrade sadržaja dokumenta nezavisno od njegove forme (od njegovog materijalnog oblika); informacijska tehnologija tu je mogućnost približila stvarnosti: određeni sadržaji (podaci) mogu se obrađivati nezavisno od forme dokumenta.

Sadržaj dokumenta (podaci sadržani u dokumentima) postao je nezavisan od dokumenta: time je otvoren put konstituisanju informatike/informacijske nauke. Taj praktični i teorijski pomak prate i institucionalne promene; na primer, godine 1968. Američki institut za dokumentaciju (*The American Documentation Institute - ADI*) promenio je naziv u Američko društvo za informacijsku nauku (*The American Society for Information Science - ASIS*).

Presudan uticaj na stvaranje informacijske nauke (njenih postavki, teorija i metodologije) imao je, i još uvek ima niz nauka i disciplina što su nastale pre ili posle drugog svetskog rata; za većinu tih nauka komunikacija je zajedničko područje, iako se međusobno razlikuju po predmetu, metodama i ciljevima. Istovremeno sve te nauke utiču jedna na drugu, one su „interdisciplinarne” ili „transdisciplinarne”.

1938. semiotika (Charles Morris)

1939. teorija odlučivanja

1944. teorija igara (John von Neuman, Oskar Morgenstern)

1944. računarske nauke (John von Neuman, Alan Mathison Turing)

1945. dokumentalistika

1948. teorija informacija (Claude Shannon, Warren Weaver)

1948. kibernetika (Norbert Wiener)

1950. opšta teorija sistema (Ludvig von Bertalanffy)

1951. strukturalna lingvistika (Noam Chomsky, Zellig S. Harris)

1950-tih formiraju se komunikacijske i bihejviorističke nauke

1961. informacijska nauka.

Informacijska nauka razvija svoju metodologiju koristeći se spoznajama raznih disciplina. Koristi se metodama semiotike (koju neki smatraju teorijskom osnovom informacijske nauke) pa se sintaksa, semantika i pragmatika koriste u planiranju sistema za pronalaženje informacija i u stvaranju jezika za pretraživanje informacija. Teorija informacija za informacijsku nauku nije važna samo kao metoda za kodiranje i dekodiranje informacija već je dugo imala i mnogo važniju metodologijsku ulogu jer je bila paradigma za tumačenje komunikacijskih procesa; metodama matematičke logike informacijska nauka se koristi u strukturiranju jezika za pretraživanje informacija i za formalizaciju logičkih procesa za pretraživanje. Istorija knjige i nauka o nauci važni su za informacijsku nauku jer doprinose razumevanju komunikacijskih formi (knjige, periodike, ...) kroz istoriju, odnosno razumevanju uslovljenosti svake komunikacijske forme istorijskim prilikama.

4.2 Principi i teorijske osnove

S. C. Bradford [78] prvi je počeo koristiti termin relevantan na način kao što se taj termin danas koristi u informacijskoj nauci. On je između 1930-tih i 1940-tih godina pisao o člancima „relevantnim za temu”. Pojavom i razvojem sistema za pronalaženje informacija (od 1950-tih) glavni cilj tih sistema postao je pretraživanje i osiguravanje relevantnih informacija. Tako se i pre pojave informacijske nauke pravi razlika između raspoloživih i relevantnih informacija.

Pojavom informacijske nauke relevantnost je postala i ostala ključni pojam za procenjivanje informacijske delotvornosti (prakse) i najvažniji teorijski pojam za procenu razmene informacija i razumevanje komunikacijskog procesa. Naime, predmet i područje informacijske nauke određen je problemima razmene znanja generalno, odnosno razmenom (za)pisanog znanja posebno. Na

teorijskom nivou informacijska nauka se zato bavi proučavanjem ponašanja i učinaka informacije kao fenomena, a komunikacije kao procesa; na praktičnom nivou relevantne informacije uvek se dobijaju uspostavljanjem određenih selektivnih mehanizama u komunikacijskom procesu.

U oba slučaja relevantnost je ključni pojam koji intuitivno i u praksi prepoznajemo kao „potrebnu”, „nužnu”, itd. informaciju, ali koji je na teorijskom planu teže odrediti. Postoji mnoštvo različitih teorijskih interpretacija relevantnosti koje polaze od zajedničke premise: relevantnost je mera delotvornosti između izvora i odredišta u komunikacijskom procesu; budući da je mera odnos relevantnost je takođe odnos [79].

Tumačenja relevantnosti zasnivaju se na teorijskim radovima i zakonima što se javljaju već 1930-tih godina, ali su tek 1970-tih privukli pažnju velikog broja istraživača. Tada dolazi do naglog razvoja bibliometrije i bibliometrijskih istraživanja tj. do primene kvantitativnih postupaka za analizu svojstava zapisanog znanja, ali i za analizu razmene informacija u komunikacijskom procesu generalno.

S. C. Bradford [80] je istraživao odnos relevantnih članaka o određenoj temi prema vrsti i broju časopisa u kojima se ti članci pojavljuju. Empirijska istraživanja potvrdila su zakon o distribuciji članaka koji je Bradford definisao (1948) na sledeći način: „... ako su naučni časopisi svrstani u niz prema opadajućoj produktivnosti članaka o određenom predmetu, oni se mogu podeliti na nukleus periodike prvenstveno posvećen tom predmetu i na nekoliko grupa ili zona što sadržavaju isti broj članaka kao i nukleus, a pritom će broj časopisa u nukleusu i u zonama što slede biti u odnosu $1 : n : n^2 : n^3 \dots$ ”.

Tabela 4.1 Bradfordov zakon

zona	broj naslova časopisa	broj članaka
1	9	429
2	59	499
3	258	404

Ovaj zakon bio je i još uvek je podloga mnogim teorijskim radovima i empirijskim istraživanjima ali i osnova za različite interpretacije. On upozorava na činjenicu da npr. 20% naslova časopisa objavljuje oko 80% relevantnih članaka o određenom predmetu. Ali isto tako, od ukupnog broja članaka o nekom predmetu jednu trećinu možemo naći u specijalizovanim časopisima za tu struku, drugu trećinu možemo naći u srodnim časopisima a za treću trećinu članaka ne možemo predvideti gde će se pojaviti.

A. Lotka [81] istraživao je produktivnost autora u određenom naučnom području. Utvrdio je da se, kad se zna broj autora koji su napisali samo jedan članak, može predvideti i broj autora koji su napisali dva, tri, četiri itd. članaka. Broj ljudi koji stvaraju n članaka aproksimativan je odnosu $1/n^2$. Na primer, ako se želi odrediti broj autora koji su napisali dva članka ($n = 2 \Rightarrow 4$), onda se broj autora koji su napisali samo jedan članak (1) deli sa četiri (n^2).

Tabela 4.2 Lotkin zakon o produktivnosti autora

Autori	Članci
100	1
25	2
11	3
6	4
4	5

Ovakva distribucija produktivnosti autora gotovo se redovno javlja u prirodnim naukama (ili preciznije u hard-sciences) ali nije jednaka za sve nauke. Istraživanja o produktivnosti autora, na primer u informacijskoj nauci, pokazuju da bi faktor produktivnosti trebao biti $1/n$ pre nego $1/n^2$.

G. K. Zipf [82] je istraživao distribuciju reči u tekstu. On je ustanovio sličnu pojavu kao i u pre spomenutim zakonima: mali broj reči pojavljuje se vrlo često. Ako se reči poredaju po frekvenciji pojavljivanja onda je umnožak redosleda pojavljivanja s frekvencijom pojavljivanja konstantan:

$$fr = c$$

Tabela 4.3 Zipfova distribucija

reči	redosled	frekvencija	red. x frekvencija
the	1	245	245
of	2	136	274
terms	3	98	294
to	4	81	324
a	5	65	325
and	6	61	366
in	7	55	385
were	8	52	416

Grafički se distribucija reči prema Zipfovom zakonu može prikazati kao hiperbolička kriva. Zipfov model distribucije reči u osnovi je jednak modelu Bradfordovog zakona, samo što je primenjen u drugom području. Zipfov zakon našao je primenu u planiranju jezika za indeksiranje i u planiranju administrativnih poslova biblioteka.

I u nizu drugih istraživanja otkrivene su zakonomernosti u strukturiranju informacijskih fondova i znanja. D. de Solla Price [83] analizirao je mrežu citata (eng. citation networks) i utvrdio da je broj članaka citiranih iznad proseka vrlo mali i da tvori „research front”. D. J. Urquhart [84] je istraživao korišćenje periodike u velikim naučnim bibliotekama i ustanovio da su korisnici usmereni na vrlo mali broj naslova. T. Saračević je analizirao distribuciju informacija dobijenih pretraživanjem informacijskih sistema i ustanovio da se distribucija onih dokumenata što su ocenjeni kao relevantni potčinjava Bradfordovom zakonu.

4.3 Informacijski procesi i sistemi za obradu

Jedan od najtežih problema u obradi informacija je uskladištiti sve informacije što su sadržane na dokumentima, a potom ih pretraživati na zahtev korisnika. Kako broj uskladištenih informacija stalno raste, to je nužno precizno upravljanje informacijskim sistemima, tj. pravilan odabir informacijskog jezika za skladištenje i pronalaženje informacija. Dokumentacijski (informacijski) jezici su veštački jezici što se koriste u informacijskim sistemima, a u svrhe indeksiranja, skladištenja i pretraživanja. Postoje tri glavna tipa dokumentacijskih jezika: 1. hijerarhijski i/ili facetni klasifikacijski sistemi, 2. (kontrolisane) liste termina, 3. tezaursi.

Klasifikacijski sistemi su vrsta dokumentacijskih jezika koji grupišu pojmove u klase, koristeći najčešće hijerarhijske relacije (generičke odnose) da uspostave odnose između klasa. Uobičajena je podela klasifikacijskih sistema na: univerzalne, specijalizovane i orijentisane prema zadatku. Najpoznatiji univerzalni klasifikacijski sistemi jesu: *Deweyeva decimalna klasifikacija* (*Dewey Decimal Classification - DDC*), *Univerzalna decimalna klasifikacija* (*UDK*), *BSO* (*Broad System of Ordering - sistem širokog razvrstavanja*), *Međunarodna patentna klasifikacija* (*IPC*), itd.

Kontrolisane liste termina su veštački jezici koji se koriste za koordinirano indeksiranje, tj. za opisivanje sadržaja dokumenta pomoću termina odabranih iz kontrolisanog rečnika. Primeri ovog tipa jezika za indeksiranje jesu: liste predmetnih odrednica, liste ključnih reči i druge liste termina za indeksiranje. Zajedničko je svim ovim listama da se ne određuju semantičke relacije između termina za indeksiranje.

Tezaursi su kontrolisani i dinamički dokumentacijski jezici semantički i generički povezanih naziva koji pokrivaju određeno područje znanja. Sastoje se od deskriptora (normiranih ključnih reči) i nedeskriptora (zabranjenih termina, npr. sinonima, uz koje obično stoji uputnica na preferirani termin). U tezaursima se najčešće određuju sledeći odnosi između termina: ekvivalencije (sinonimije), hijerarhije (generički, partitivni itd.) i srodnosti. Prema broju jezika razlikuju se jednojezični i višejezični tezaursi; prema nivou obrade tezaursi se dele na makrotezaurse i mikrotezaurse.

Pod indeksiranjem se podrazumeva skup principa, metoda, postupaka i njihovih primena u informacijskom procesu, kojima se dokumentu dodeljuju termini (ključne reči ili kodovi), a u cilju opisivanja njihovog sadržaja i kasnije njihovog ponovnog pronalaženja. Sam postupak indeksiranja podeljen je u tri operacije: 1) analizu dokumenta u cilju selekcije pojmova koji prema mišljenju indeksera mogu biti od interesa za korisnike indeksa, 2) „prevod” odabranih pojmova, sadržaja ili reči u neki kontrolisani ili standardizovani dokumentacijski jezik, 3) odabir i slaganje indeksnih termina prema pravilima dokumentacijskog jezika.

Postoje dva osnovna načina indeksiranja: a) slobodno (kada se neposredno iz teksta dokumenta biraju ključne reči bez obzira na njihovu formu i semantičke odnose), i b) kontrolisano (kada se za pretraživanje koriste samo oni termini koji su unapred određeni dokumentacijskim jezikom). Najbolje rezultate pretraživanja daje metoda koordinatnog indeksiranja (metoda koja u fazi indeksiranja koristi indeksne termine na istom nivou, tako da se u fazi pretraživanja može koristiti bilo koja kombinacija termina). Ova metoda dobija široku primenu 1950-tih godina uvođenjem Uniterm sistema, ali i sistema za automatsko pretraživanje. Automatsko indeksiranje (automatska selekcija ključnih reči iz dokumenta) nalazi svoju praktičnu primenu posebno u

publikovanju indeksa „permutovanih naslova” (indeks koji ponavlja sve termine iz naslova tako da se svaki termin javlja kao odrednica).

4.4 Definicija relevantnosti

Reč relevantnost potiče od latinske reči *relevare* koja znači podići, uspraviti. Relevantno znači nešto što je znatno, važno, bitno, jezgrovito, istaknuto itd. Relevantna informacija bi prema tome bila potrebna, neophodna informacija. No šta je zapravo potrebna informacija? Kako odrediti šta je relevantno? U informatici postoje različite teorijske interpretacije relevantnosti, njene definicije i zakonitosti. Ipak, svima je zajedničko ovakvo objašnjenje: „relevantnost je mera delotvornosti između izvora i odredišta u komunikacijskom procesu”. Ona je mera kao što je na primer vreme, ali za razliku od vremena, za sada nemamo pouzdanog sata koji bi izmerio relevantnost.

Budući da relevantnost ne možemo objasniti bez komunikacijskog procesa, trebali bi razjasniti i taj pojam. Dakle, za komunikaciju kao i za relevantnost ili pak za informaciju postoje brojne definicije i različita gledišta. Ovde ćemo komunikacijski proces gledati kao razmenu znanja. Ta razmena znanja pak ima svoje izvore i svoja odredišta koji su u dinamičnoj i povratnoj vezi te mogu menjati svoje uloge. Znanje, odnosno informacije koje putuju od izvora prema odredištu (ili obrnuto) mogu menjati, nadopunjavati, poništiti ili reorganizirati postojeće informacije u odredištu (ili izvoru). Ta promena učinak je razmene znanja. „Razmena znanja je delotvorna kada i ako informacija što se prenosi iz jednog fonda rezultira promenama u drugom”. Dakle, komunikacijski proces je delotvoran ako informacija što se prenosi iz izvora promeni odredište, a relevantnost je mera te promene. Na primer, ako se komunikacijski proces događa između knjižara koji je izvor i korisnika koji je odredište, relevantnost ćemo meriti prema tome u koliko je meri korisnik dobio ono što je želeo. Ako je informacija koju je dobio od knjižara povećala ili na bilo koji drugi način promenila ono što je korisnik već od ranije znao, onda je relevantnost u tom komunikacijskom procesu bila veća. Ako pak korisnik nije saznao ništa novo, relevantnost je mala.

Danas u svetu postoje različite vrednosti – etičke, društvene, političke, filozofske, religijske, pravne itd. Neke od njih imaju različita gledišta na iste stvari. Ljudima su neke vrednosti važnije od drugih, ali ipak svi se slažu u jednom – ono što je vredno to je i bitno, relevantno. Međutim, ko će odrediti šta je vredno, odnosno šta je relevantno? Mora li ono što je jednom odredištu vredno i relevantno, biti i drugom odredištu jednako vredno i jednako relevantno? Naučnici su i na to dali odgovor: „relevantnost je subjektivan koncept jer uključuje ljudsko prosuđivanje”. Ono što je na primer istraživaču relevantna informacija, profesoru u školi će se učiniti potpuno nebitnom. Upravo zbog tih razloga teško je naći jednu zajedničku meru relevantnosti. No relevantnost se ipak meri i imamo neke prilično uspešne metode.

Relevantnost se može posmatrati s različitih aspekata u zavisnosti od toga šta utiče na nju, odnosno u kakvoj je poziciji i kojoj ulozi u komunikacijskom procesu. Ovde će biti reči o relevantnosti sa aspekta sistema.

„Teorija o relevantnosti sa aspekta sistema zasniva se na postavci da je relevantnost rezultat internih učinaka i manipulacije sistemom”. Kao što je već ranije rečeno, sistemi za pronalaženje in-

formacija vrše selekciju informacija. Pretpostavlja se da su dokumenti koje sistem dobije pronalženjem relevantni, a oni koji nisu izdvojeni nisu relevantni. Međutim, istraživanja su pokazala da sve pronađene informacije ne moraju nužno biti relevantne odredištu. Takođe, pokazalo se da nekada brojne relevantne informacije ne budu pronađene. Stoga se relevantnost počela iskazivati u procentima preciznosti (*precision*) i odziva (*recall*).

Preciznost i odziv su klasične mere uspešnosti pronalženja informacija te se koriste još od kraja četrdesetih i s početka pedesetih godina dvadesetog veka. Preciznost je odnos pronađenih relevantnih informacija prema ukupnom broju pronađenih informacija u bazi. Odziv ili potpunost je odnos pronađenih relevantnih informacija prema ukupnom broju relevantnih informacija u bazi. Prikazano tabelarno i formulom to bi izgledalo ovako:

Tabela 4.4. Izračunavanje preciznosti i odziva

Dokumenti	Relevantni	Nerelevantni	
Pronađeni	a	b	a + b
Nepronadjeni	c	d	c + d
	a + c	b + d	

Premda ovakva merenja samo kvantifikuju relevantnost, ona se i dalje najviše primenjuju. Stoga, iako su precizna merenja relevantnosti problematična, odziv i tačnost i dalje su korisni kao okvir za konceptualizaciju postupaka pronalženja.

4.5 Prevazilaženje problema

Čovekov mozak je ogromna neuralna mreža sa enormnim mogućnostima. Identifikaciju uzoraka obavlja sa neverovatnom brzinom, tj. memorisanje i pronalženje informacija. Kadgod sretnemo nama poznatu osobu, naš mozak obavi brzo pretraživanje da locira njenu sliku i asocira je sa drugim memorisanim podacima.

Kada čitamo dokument, ista neuralna mreža analizira zapisane jezičke izraze, ekstrahuje koncepte iz njih, pravi skup apstrahovanih koncepata i smešta ih u memoriju, pamti ih. To reprezentuje naše shvatanje tog dokumenta. Ako nas neko pita o čemu je taj dokument, reći ćemo, na primer „o mašinama za pretraživanje” ili „o relevantnosti”.

Mozak je ekstrahovao suštinske koncepte iz dokumenta ignorišući okolni šum. Isti proces je na delu kod procenjivanja relevantnosti dokumenta. Mozak pokušava da poklopi koncepte koje treba da nađe sa konceptima unutar dokumenta. U slučaju da su svi koncepti nađeni i da su svi dovoljno istaknuti u odnosu na druge koncepte, tada se dokument smatra relevantnim.

Dakle, ljudi traže koncepte. U prirodnim jezicima, koncepti se obično prenose imenicama. Tekst može da se shvati ako su druge vrste reči isključene izuzev imenica, ovo se obično dešava kada brzo čitamo. Ovo znači da imenice mogu da se koriste za ekstrahovanje koncepata iz dokumenta. Prema tome, imenice su važnije od drugih reči. Već je ranije rečeno da nemaju sve reči istu težinu

unutar dokumenta, ali ne samo zato što se češće ili ređe pojavljuju u drugim dokumentima, već zato što su ili nisu koncepti.

Problem postavljanja upita proističe iz činjenice da korisnika koji postavlja upit najčešće interesuju dokumenta vezana za određeni koncept, onako kako se koncept definiše u semantičkim mrežama reči. Upiti za pretraživanje tekstualnih sadržaja se, međutim, po pravilu sastoje od jedne ili više ključnih reči, tačnije niski koje odgovaraju tim rečima, povezanih logičkim operatorima i/ili. Izbor ključnih reči, odnosno odgovarajućih niski je očigledno od suštinskog značaja za kvalitet rezultata koji će biti dobijeni kao odgovor na upit. Na prvi pogled, osnovni problem leži u tome što korisnik, prilikom formulacije upita, može propustiti da u upit uključi neke od reči kojima se koncept označava. Time se smanjuje *odziv* sistema, parametar koji predstavlja odnos relevantnih dokumenata dobijenih upitom i broja ukupno raspoloživih relevantnih dokumenata u tekstualnim resursima koji se pretražuju. Ovaj problem bi, na prvi pogled, mogao efikasno biti rešen prostim proširenjem upita, odnosno dodavanjem reči koje je korisnik propustio da navede. Međutim, proširenje skupa reči kojima se označava koncept u upitu, iako u načelu doprinosi odzivu, može imati i suprotan efekat. Naime, kako je veliki broj reči višeznačan, a postoji i velika homonimičnost oblika, dodavanje novih reči u upit može dovesti do porasta broja irelevantnih dokumenata u odgovoru na upit čime se umanjuje *preciznost*, koja predstavlja odnos dobijenih relevantnih dokumenata i ukupnog broja dobijenih dokumenata. S obzirom na ovu međuzavisnost odziva i preciznosti, reči odnosno niske koje se koriste u upitu moraju biti pažljivo odabrane kako bi se obezbedila optimalna ravnoteža između ova dva parametra.

Izbor niski termova u upitu se dodatno komplikuje kada su u pitanju jezici sa bogatom morfološkom strukturom kakav je srpski jezik. Pored osnovnog oblika ključnih reči, u upite tada često treba uključiti i niske koje odgovaraju morfološkim oblicima ključnih reči. Pojedini veb pretraživači, kao što je Google, pokušavaju da delimično reše ovaj problem. Tako se upiti na ovom pretraživaču za srpski jezik od nedavno proširuju, ali očigledno korišćenjem neke vrste stemera, programa za odsecanje sufiksa i flektivnih nastavaka i svodenje na koren reči. Time se problem fleksija rešava samo delimično i svakako ne na sistematičan način. Kao što je to često slučaj sa stemerima, pored (nekih) flektivnih formi stemer okuplja i srodne reči. Tako Google upit za reč *prevodilac* nudi i odgovore koji sadrže *prevod*, dok upit koji sadrži složenicu *slati poruku* daje samo strane koje sadrže glagol *slati* u infinitivu ili glagolsku imenicu *slanje* (sending) a izostavlja brojne stranice koje sadrže druge forme ovog glagola kao, na primer, *šaljem poruku*. Pri tome korisnik nema nikakvu kontrolu nad načinom na koji Google proširuje upit. Konačno, kada se radi o srpskom, postoji i problem dva pisma, ćirilnog i latiničnog. Postavljanje upita korišćenjem jednog pisma podrazumeva i dobijanje odgovora koji sadrže samo dokumenta na tom pismu, što ne mora nužno biti i korisnikova namera.

Jezički resursi, kao što su elektronski rečnici i semantičke mreže reči pružaju mogućnosti za sistematično rešavanje izloženih problema pri postavljanju upita. Pri tome je, u cilju postizanja ravnoteže između odziva i preciznosti, potrebno korisniku obezbediti što veću fleksibilnost u izboru niski od kojih će se formirati konačni upit. Integracija više resursa daje korisniku mogućnost da upit proširi morfološki, semantički itd.

Egzaktno pretraživanje koje mi danas koristimo može samo da bude privremena metoda ka shvatanju prirodnog jezika i sposobnosti ekstrahovanja odgovarajućih koncepata iz postavljenih upita i dokumenata.

5. Funkcionisanje mašine za pretraživanje

Da bi se dizajnirala aplikacija za pretraživanje informacija mora se dobro poznavati funkcionisanje ove vrste aplikacija.

Anatomija mašine za pretraživanje

Konstrukcija mašine za pretraživanje može se razložiti na tri različite kategorije:

- Procesiranje dokumenta
- Indeksiranje podataka
- Procesiranje upita

Procesiranje dokumenta se bavi pret-procesiranjem podataka, indeksiranje podataka opslužuje odgovarajuće indeksirane podataka a procesiranje upita podnosi i pronalazi rezultate od mašine za pretraživanje

5.1 Procesiranje dokumenta

Pošto tekst ne može direktno da se interpretira od strane algoritama pretraživanja, procedura indeksiranja treba da mapira tekst u odgovarajuću predstavu njegovog sadržaja [85]. Procesor dokumenta upravo radi to. On priprema, procesira i unosi dokumente koje će mašina za pretraživanje dodati u svoj indeks. Mnoštvo problema se javlja kada se procesira velika količina podataka. Čak i najprostiji poslovi postaju komplikovani kada su podaci enormne veličine. Dalje, parsiranje podataka je teško jer postoji veoma malo strukture u različitim dokumentima.

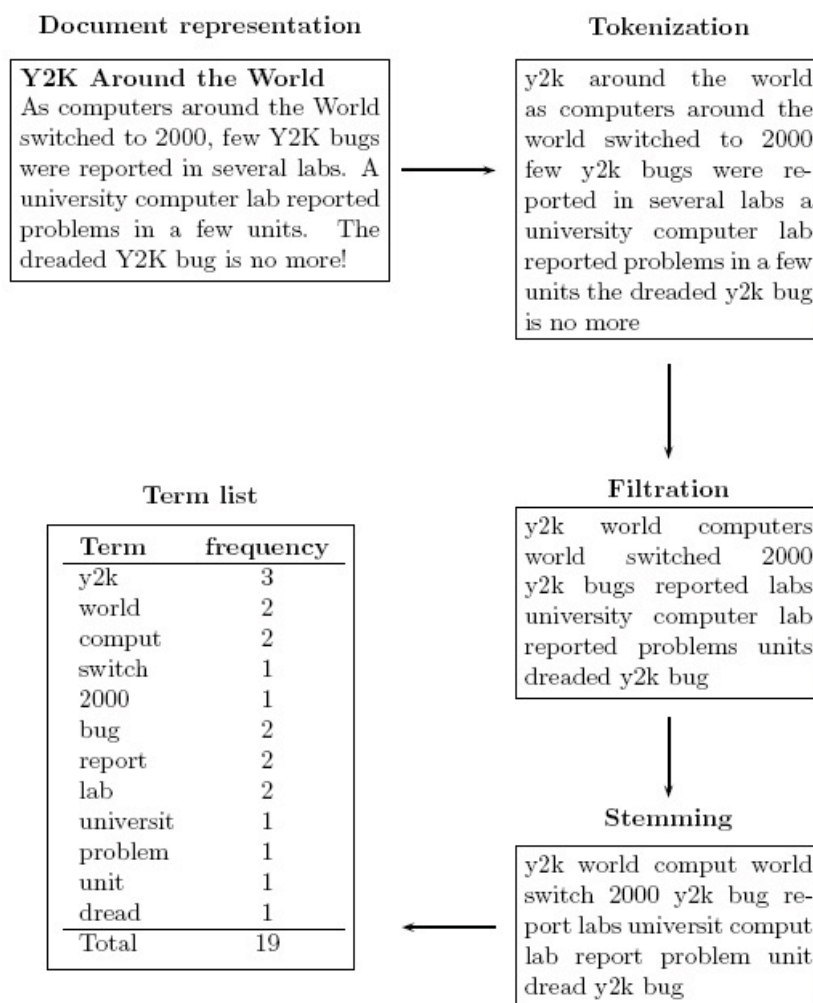
Kada se implementira procesor dokumenta nastaju razni problemi koji se odnose na parsiranje i identifikovanje indeksibilnih elemenata (engl. *terms*). Kako procesor dokumenta predstavlja tekst zavisi od izbora elemenata teksta koji se smatraju značajnim (leksička semantika) i koja kombinaciona pravila smatra značajnim za te elemente (kompoziciona semantika). Uobičajeno je da se ne obraća pažnja na kompozicionu semantiku teksta i tekst se predstavlja kao histogram termova koji se pojavljuju u tekstu, dakle pridržavajući se samo leksičke semantike.

Da bi kreirao histogram, procesor dokumenta obavlja neke ili sve sledeće korake:

1. Normalizuje strim dokumenta u predefinisani format

2. Prekida strim dokumenta u željene pretražive jedinice
3. Identifikuje potencijalne indeksibilne elemente u dokumentima
4. Uklanja tzv. zaustavne (engl. *stop*) reči
5. Redukuje termine na njihove osnovne oblike (engl. *stem*)
6. Ekstrahuje stavke indeksa
7. Izračunava težine (težinske faktore - *pondere*) termina

Ovi koraci su veoma važni u procesu kreiranja mašine za pretraživanje pošto termini koje procesor dokumenta identifikuje su termini koji kasnije mogu biti traženi.



Slika 5.1 Važni koraci u obradi dokumenta

Tekst je izuzetno dinamičan i može da sadrži stotine hiljada karaktera, simbola, znakova interpunkcije, brojeva itd. Usled toga ima mnogo problema u pokušaju da se kod takvih podataka identifikuju indeksibilni elementi. Pored različitih pravila parsiranja i opsluživanja podataka, procesor dokumenta treba da olakša potonja izračunavanja selektovanjem samo termina koji su važni i relevantni bez gubljenja vrednijih informacija. Ovaj kompromis je jedan od realnih izazova u implementiranju dobrog procesora dokumenata.

Normalizovanje i identifikovanje indeksibilnih termova

Pre svega, tekst može da se raščlani u termove na očigledan način a to je prekid označen belinama, tj. definisanjem pojedinačnih reči kao terma. Ovaj pristup se na engleskom naziva *bag-of-words* pošto se tekst posmatra kao torba reči, tj. ne obraća se pažnja na kompozicionu semantiku.

Znaci interpunkcije mogu da podele rečenice ali se takođe koriste i u drugim kontekstima, nazivi promenljivih u programskom kodu, recimo *x.id*. Uklanjanje znakova interpunkcije ne može se obavljati nekritički iz rečenica, Jedno rešenje je da se znaci interpunkcije zamenjuju sekvencom specijalnih karaktera, recimo *<x>*. U tom slučaju prethodno navedena promenljiva bi imala oblik *x<x>.id*. Da bi se izbeglo gubljenje informacija iz pojedinačnih reči i da se proces ne bi mnogo iskomplikovao, preporučuje se da se beline koriste kao graničnici reči a da se znaci interpunkcije uklanjaju samo ako reč počinje ili se završava sa znakom interpunkcije.

Kada se pronadu indeksibilni termovi, teško je reći koja reč ili grupa reči je dovoljno važna da treba da se indeksira. Veličina ispisa slova reči može da nagovesti važnost, recimo naslovi se obično ispisuju većim fontom. Jedan od pristupa je da se rečima ispisanim velikim slovima da veća važnost jer se tako obično ispisuju skraćenice organizacija ili ističu neki koncepti.

Imenice obično nose najveću semantičku težinu i druge vrste reči mogu da se uklone. Da bi se odredile vrste reči, koreni pojedinih reči itd., potrebno je koristiti tehnike računarske obrade prirodnih jezika koja se oslanja na lingvistička pravila za konkretan jezik.

5.2 Ukratko o leksikografiji

Pre nego što se krene u dalje razmatranje pronalaženja informacija, biće dat kratak osvrt na neke opšte pojmove kao što su jezik, znak, leksika, reč, leksička jedinica i leksem.

Jezik je moguće definisati kao sistem znakova koji ljudi upotrebljavaju u jezičkoj delatnosti kao jednom od oblika ljudske komunikacije. Jezik je, dakle, sistem znakova, kôd. Skup tih znakova što čine jezički sistem naziva se leksika. Leksika je uopšteno govoreći „sveukupnost reči koje ulaze u sastav nekog jezika ili narečja”. Postoji razlika između jezičkih znakova nižeg nivoa (morfemi, reči) i jezičkih znakova višeg nivoa (sintagma, rečenica, tekst ili diskurs). Među mnogim definicijama reči najprikladnijom smatra se ona Leonarda Bloomfielda koja kaže da je reč „minimalna slobodna forma, dakle onaj najmanji oblik koji ima određeno značenje kada stoji sam za sebe” (Bloomfield, 1933). Nije dakle iznenađujuće što je jezički znak najnižeg nivoa koji leksikografija opisuje upravo reč. Sintagma je pak (uz mnoge druge definicije) autonomni i kompleksni jezički znak koji ima određenu sintaktičku i semantičku samostalnost. U slučaju da sintagma pokazuje veliki stupanj samostalnosti, zadatak leksikografije je opisivanje i tih sintagmi. Njen zadatak je, naime, da opisuje znakove od kojih se tvori jezički iskaz, i to one koji imaju svoje jedinstveno značenje, bez obzira na oblik ili broj elemenata koje ti jezički znakovi sadrže. Iz tog je razloga praksa u leksikografiji da se ne govori samo o rečima ili pak rečima i sintagmama, već uopšteno o leksičkim jedinicama.

Kao sinonim leksičkoj jedinici često se navodi pojam leksem. Leksem je osnovna jedinica leksike koja se sastoji od jedne ili više reči i koja tvori jedan pojam, kao i jedinica leksičkog sistema koja

obuhvata sve oblike i sva značenja neke reči. Iz tog razloga može se zaključiti da leksičke jedinice u odnosu prema leksici treba posmatrati kao predstavnike celokupne leksike.

Leksičke jedinice se kao predstavnici leksike sastoje od tri elementa: oblika, značenja i funkcije. Pod funkcijom leksičke jedinice treba razlikovati njenu sintagmatsku odnosno paradigmatsku osu. Sintagmatska osa se odnosi na jezičke elemente koji leksičkoj jedinici mogu prethoditi ili one koji joj mogu slediti dok se paradigmatska osa odnosi na druge oblike leksičke jedinice koji se mogu ostvariti na njenom mestu.

Govoreći o leksikografiji kao naučnoj disciplini koja se bavi popisivanjem leksike nekog jezika, bitni su sledeći jezički pojmovi: leksičko značenje, menjanje oblika reči, kombinacije reči i promene u jeziku. Leksičko značenje se odnosi na značenje kao element leksičke jedinice dok se menjanje oblika reči i kombinacije reči odnose kako na oblik (izvođenje i slaganje), sintagmatsku osu funkcije (rekcije, nevezane kombinacije) tako i na paradigmatsku osu funkcije (paradigma, vezane kombinacije) kao elemente leksičkih jedinica. Variranja u jeziku ne odnose se ni na jedan od tri osnovna elementa leksičke jedinice, no ona su takođe od velikog značaja za leksikografiju.

5.3 Osnovni pojmovi računarske obrade prirodnog jezika

Oblici su različiti likovi u kojima se jedna reč može pojaviti i koji se razlikuju gramatički, ali ne i po osnovnom značenju. Na primer, *hoda* i *hodam* su oblici iste reči.

Za svaku vrstu reči moguće je dogovorno odrediti kanonski, polazni oblik - lemu. Lema neke imenice je njen nominativ jednine, a lema glagola infinitiv. Na primer, neki od oblika leme hodati su: hodam, hodao, hodamo, itd. Ponekad nije moguće jednoznačno odrediti lemu neke reči. Na primer, reč vode ima sledeće leme: vod, voda, voditi.

Generalno, kada dva oblika imaju isti pisani oblik, a različito značenje, onda govorimo o homografiji. Postupak nalaženja leme zovemo lematizacija.

MSD ili morfosintaktički deskriptor sadrži informacije o gramatičkim kategorijama, tj. vrsti reči, te dodatne informacije vezane uz tu reč (na primer, rod, broj i padež za imenice).

Engl. stemmer je program ili algoritam kojim se određuje koren (najčešće leksički, tj. korenski morfem) neke reči. Na primer, stemmer bi mogao naći da je koren reči osećajnost oseć, ili da je koren reči noga nog. Većina stemmera po određenim pravilima uklanja sufikse zadate reči te često rezultat nije ispravan koren, već samo aproksimacija korena. Na primer, stemmeri za engleski jezik dade *drie* umesto *dry* za reč „dries”.

Najpoznatiji algoritam za engleski jezik je algoritam Martina Portera. To je ujedno i najčešće korišćen engleski stemmer te je postao *de facto* standard. Valja napomenuti da je stemmer morfološki siromašnog jezika poput engleskog znatno lakše napraviti od stemmera morfološki složenog jezika poput srpskog.

Pojavnica je svako pojedinačno pojavljivanje reči u nekom tekstu, a različnica je jedinstveni oblik pojavnice u istom tekstu. Na primer, rečenica „Ko se mača lati, od mača će i stradati.” ima 9

pojavnica, ali samo 8 različenica jer se različenica mača ponovila dva puta.

Zaustavne (engl. *stop words*) reči su reči koje imaju gotovo isključivo gramatičku funkciju, ili se javljaju vrlo često u nekom dokumentu. Zaustavne reči se ne prikazuju u popisu lema, niti u popisu različenica. Takođe, lematizacija i stemming neće se obavljati nad takvim rečima. Primeri zaustavnih reči su engleske reči *the, a, I* a srpske reči *je, pa, to* itd.

Ignorisane reči su reči uglavnom specifične za neko područje koje ne donose gotovo nikakve dodatne informacije indeksatoru. Nad tim rečima obavlja se lematizacija, ali se ne prikazuju u popisima lema i različenica.

Srpski jezik je morfološki vrlo bogat, a to predstavlja dodatnu teškoću pri računarskoj obradi teksta. Rešavanje problema lematizacije te korenovanja (engl. *stemming*) preduslov je za bilo kakvu mašinsku obradu prirodnog jezika. Za većinu svetskih jezika, a pogotovo za engleski, postoji niz javno dostupnih alata i razrađenih algoritama. Za srpski jezik takvi alati su u postupku nastajanja. *Grupa za jezičke tehnologije Univerziteta u Beogradu*, koja deluje pod okriljem *Matematičkog fakulteta Univerziteta u Beogradu* bavi se pored ostalog razvojem jezičkih i tekstualnih resursa (različiti tipovi e-rečnika, leksička baza podataka itd.) [86].

Tokeni

Individualni karakteri iz teksta bez konteksta nisu korisni i moraju se grupisati u tekstualne jedinice koje se nazivaju tokeni. Token može biti jedna reč ili term, ali može biti i sekvenca karaktera, fraza, e-mail adresa itd. Tokenizacija je tzv. *bottom-up* pristup u obradi teksta kod koga se najmanjom jedinicom teksta smatra token. Druge metode obuhvataju razdvajanje teksta u rečenice i raščlanjavanje (parsiranje) svake rečenice. Ovde će uglavnom biti reči o *bottom-up* pristupu iz više razloga. Prvo, tekst iz nekih izvora je vrlo teško raščlaniti i često ne može da se deli u rečenice. Drugo, rečenice koje su ekstrahovane mogu imati više različitih raščlanjavanja i teško je odlučiti koje raščlanjavanje najbolje odgovara značenju koje je autor zamislio. Treće, potrebno je mnogo manje vremena za tokenizaciju teksta nego za raščlanjavanje teksta. Zadatak ekstrahovanja značenja i razumevanja teksta obično se prepušta modulima višeg nivoa koji rade sa većim odlomcima teksta.

Lista tokena je pogodna za gradnju nekoliko tipova aplikacija koje se oslanjaju na tekst. Svakako najpoznatiji tip je mašina za pretraživanje koja pronalazi u dokumentima tekst koji se podudara sa termovima upita. Termovi upita su definisani listom tokena koji su upisani u polje pretraživanja aplikacije. Drugi primer su aplikacije koje pronalaze najbolji odgovor na pitanje. Osećanje izraženo u dokumentu takođe može da se detektuje na osnovu tipa prideva korišćenih u tekstu. Različiti uzorci teksta mogu se porediti da bi se procenila sličnost ili detektovao plagijat. Mogu se pronaći najpopularnije fraze navedene u tekstu dokumenta. Takođe, lista tokena je dovoljna da se dokument na osnovu nje rasporedi u pripadnu kategoriju sa prihvatljivom tačnošću.

Ekstrakcija teksta

Ekstrakcija teksta nije neposredno povezana sa tokenizacijom, ali je neophodan prethodnik za ekstrakciju tokena iz formatiranog fajla. Formatirani tekstualni fajlovi su obično obogaćeni bojama, fontovima i stilovima kako bi poprimili vizuelno dopadljiv izgled. Tekst filter kreiran

za konkretni tip formatiranog dokumenta u većini slučajeva ekstrahuje sam tekstualni sadržaj, ignorišući bilo kakve meta podatke. Nažalost, time se gubi deo informacija, kao na primer autorovo formatiranje nekog dela teksta većom veličinom fonta kako bi mu dodelio veću važnost u odnosu na ostali sadržaj. Generalno, tekst filteri ne zadržavaju tu informaciju.

Osvrt na obradu srpskog jezika

Polje tehnologije ljudskih jezika konačno je dovoljno sazrelo te se mnogo veća pažnja posvećuje razvoju lingvističkih resursa za jezike kojima govori manje od deset miliona ljudi (male jezike). Razvoj mnogih (usmerenih) aplikacija poput komandovanja i upravljanja, diktiranja, prepoznavanja govorne pošte, audio pretrage, automatskog prevođenja novinskih tekstova i skupštinskih zapisnika dostigle su visok kvalitet pogodan za praktičnu upotrebu. Sistemi prirodnih jezika prošireni su na veći krug jezika. EU sada ima 23 zvanična jezika i koristi veći broj mašinski čitljivih rečnika (za npr. finski, češki, mađarski, slovenački ...).

Započete su i promovišu se standardizacija, lingvistička normalizacija i revitalizacija malih jezika, o čemu svedoči i povećan broj veb strana na manje korišćenim jezicima.

Automatska obrada malih jezika treba da prevaziđe niz poteškoća koje proističu iz njihovog osobenog statusa.

a) Kako ovi jezici imaju malo maternjih govornika, imaju i malo lingvista kojima je to maternji jezik i još manje računarskih lingvista. Zbog toga može doći do poteškoća u primenjivanju tradicionalnih pristupa tagiranju, parsiranju itd. koji se zasnivaju na pravilima.

b) Slaba finansijska podrška koju ovi jezici dobijaju izgleda da takođe praktično isključuje pristupe bazirane na pravilima, s obzirom na količinu ljudskog rada koju ovi pristupi obično zahtevaju. Ovaj problem bi se mogao prevazići usvajanjem računarskih okvira izvedenih na osnovu drugih jezika.

v) Korpusni pristupi su primenjivi jedino ako su dostupni odgovarajući korpusi. S druge strane, formiranje korpusa je zahtevno i u vremenskom i u finansijskom smislu, i zahteva dobru lingvističku koncepciju, posebno ako je u pitanju korpus opšte namene.

g) Pristupi bazirani na primerima izgleda u ovom smislu više obećavaju ukoliko su potrebni specifični primeri, a ne korpusi opšte namene. Takođe se čini i da je lakše da se implementira kompilacija specijalnih primera nego da se pišu formalna pravila. Ipak, malo se zna o izvodljivosti ovog obrasca u slučaju manjinskih jezika.

d) Mogu se razviti, ili su već u upotrebi tehnike plitkog znanja koje koriste specifične osobine jezika ili jezičkih porodica. Ovo, s druge strane, može ometati transfer takvog pristupa od jednog jezika ka drugim jezicima. Neke tehnike bi tako mogle da funkcionišu sa analitičkim, ali ne i sa aglutinativnim jezicima itd. Različiti sistemi pisanja takođe mogu da spreče primenjivost jednog jednostavnog pristupa na druge jezike.

Što se tiče srpskog jezika, prethodna obrada tekstualne kolekcije mogla bi da se sastoji od tri glavne faze:

- (1) konverzija ne-ASCII srpskih slova (ć, č, đ, š i ž) u odgovarajuće ASCII transkripte (cx, cy, dx, ex, i zx),
- (2) odbacivanje i
- (3) konflacija reči.

Dok su konverzija ne-ASCII slova i odbacivanje prilično jednostavne operacije, za osmišljavanje i razvoj uspešnog algoritma za konflaciju se to ne bi moglo reći. Pritom ovaj proces ima i dalekosežne posledice na funkcionisanje sistema za pronalaženje informacija.

Srpski jezik ima dosta komplikovanu morfologiju: sedam imeničkih padeža, tri imenička roda, tri imenička broja, šest glagolskih vremena, šest glagolskih formi itd. Sasvim logično je pretpostaviti da bi kvalitet konflacije mogao imati bitne posledice na proces pronalaženja informacija.

Tagiranje vrstama reči (gramatičko tagiranje ili na engleskom *POS tagging* ili POST) je proces anotiranja reči u tekstu na osnovu toga kojoj vrsti reči pripadaju. Tagovi (etikete) se biraju i na osnovu definicije reči i na osnovu njenog konteksta (odnosa sa susednim i povezanim rečima u frazi, rečenici ili pasusu). Tagiranje vrstom reči se danas skoro isključivo obavlja u kontekstu računarske lingvistike.

Algoritmi koji su u upotrebi povezuju zasebne termine, kao i skrivene vrste reči, u skladu sa skupom opisnih tagova. Plitko parsiranje (poznato i kao podsecanje ili „lako parsiranje”) je analiza teksta u toku koje se identifikuju njegovi konstituenti (imeničke fraze, glagolske fraze, predloške fraze, itd). Međutim, od ovog procesa se ne očekuje ni da specifikuje unutrašnju strukturu niti ulogu tih fraza u glavnoj rečenici. Ovo je procedura koja se u velikoj meri koristi u obradi prirodnih jezika.

5.4 Indeksiranje podataka

Indeksiranje podataka postalo je izuzetno važan deo obrade podataka kod mašina za pretraživanje. Ako podaci nisu odgovarajuće indeksirani, ne može se očekivati da će mašina za pretraživanje dati dobre rezultate za postavljeni upit. U prethodnom odeljku su obrazloženi koraci koje je potrebno preduzeti kod pret-procesiranja tekstualnih podataka a pre indeksiranja. Iz toga se može zaključiti koliko važnu ulogu za indeksiranje podataka ima obrada dokumenata. Tokom obrade dokumenata treba da budu identifikovani termovi koji treba da se indeksiraju, a indeksator podataka mora da donese konačnu odluku šta će biti uključeno a šta će se isključiti. Najjednostavniji pristup je da se indeksira svaki term koji se pojavi u dokumentu. Međutim, takav pristup zahteva enormni skladišni prostor, što opet negativno utiče na vreme pristupa i pronalaženja podataka. U sledećem odeljku razmatra se kako termove treba indeksirati da bi se dobili pozitivni rezultati na postavljene upite.

Tradicionalno indeksiranje

Popularan način kreiranja indeksa za mašine za pretraživanje je da se dodaju termovi iz procesora dokumenata u indeks uz izračunavanje blizine terma unutar dokumenta i dodavanje i te informacije u indeks. Na taj način moguće je obaviti pretraživanje za kombinaciju reči u dokumentu, na primer za upit „računarske nauke”, mašina za pretraživanje bi najbolje rangirala

dokumente koji sadrže ta dva terma jedan do drugog. Međutim, ova dodatna informacija, blizina terma, utiče na veličinu indeksa i stoga mašina za pretraživanje može postati neefikasna. Ovo je uticalo na uvođenje tzv. invertovanog fajla. Većina tradicionalnih mašina za pretraživanje koristi tu strukturu da predstavi svoje indekse.

Kada se predstavlja jedna tako ogromna količina podataka nije praktično navoditi reči po dokumentu u indeksu. Umesto toga koristi se struktura invertovanog indeksa koja navodi dokumente po reči. Invertovani indeks je struktura indeksa koja smešta i mapira reči u svoje lokacije u dokumentu ili skupu dokumenata, omogućavajući potpuno pretraživanje teksta. Takođe, ova struktura optimizuje brzinu upita pošto upit može da traži reč i nađe dokumente koji je sadrže.

Invertovani indeks ima više varijanti ali obično sadrži referencu na dokumente za svaku reč a često i položaj reči u dokumentu. Neka imamo sledeći tekst:

$$T = \{t_0 = i \text{ love you}, t_1 = \text{you love } i, t_2 = \text{love is blind}, t_3 = \text{blind justice}\},$$

tada će invertovani indeks imati oblik dat sledećom tabelom:

Tabela 5.1 Invertovani indeks

Reč/term	Indeksne informacije
i	{ 0, 1 }
love	{ 0, 1, 2 }
you	{ 0, 1 }
is	{ 2 }
blind	{ 2, 3 }
justice	{ 3 }

Kada tražimo reči *love* AND *blind* dobićemo rezultat:

$$\{0, 1, 2\} \cap \{2, 3\} = \{2\}.$$

Potpuno invertovani indeks može se kreirati iz prethodno datog teksta T dodavanjem u indeks i lokalnog rednog broja reči u indeks, što je prikazano u sledećoj tabeli.

Tabela 5.2 Potpuno invertovani indeks

Reč/term	Indeksne informacije
i	{ (0, 0), (1, 2) }
love	{ (0, 1), (1, 1), (2, 0) }
you	{ (0, 2), (1, 0) }
is	{ (2, 1) }
blind	{ (2, 2), (3, 0) }
justice	{ (3, 1) }

Kada budemo kao upit postavili frazu *i love you* dobićemo kao rezultat t_0 i t_1 . Međutim, ako u indeksu koristimo i poziciju reči tada ćemo kao rezultat dobiti samo t_0 , što je u prethodnoj tabeli označeno podebljanim fontom.

Takođe, indeks može da obuhvati i detalje kao što su položaj početnog karaktera, ponder terma i frekvencija terma.

5.5 Obrada upita

Prilikom korišćenja mašine za pretraživanje, korisnici teže da postave par specifičnih termova kao upit. Često, upit je samo jedna reč. U takvom slučaju ne očekuje se da mašina za pretraživanje vrati relevantne rezultate, posebno ako ta reč ima više značenja, kao na primer reč *jaguar*. Ako korisnik postavi kao upit baš tu reč, jaguar, mašina za pretraživanje ne može da zna da li korisnik traži informacije o automobilu, životinji ili nekadašnjem trkačkom timu Formule 1. Beitzel je u svom istraživanju [87] ustanovio da je prosečna dužina upita 1,7 terma za popularne upite a 2,2 za sve upite. Međutim, po istraživanju Yahoo! iz 2006. godine, prosečna dužina upita se povećala na 3,3 terma.

Proširenje upita

Kod većine mašina za pretraživanje, duži upiti obično daju relevantnije rezultate pretrage. Proširenje upita se obično koristi da poboljša rezultate pretrage bilo proširivanjem izvornog upita sa dodatnim termovima i/ili reponderisanjem termova izvornog upita.

Dodavanje termova upitu može da se radi automatski, ručno ili uz asistenciju korisnika [88]. Kod automatskog proširenja upita, na primer, algoritam izračunava pondere svih termova u najbolje rangiranim rezultatima inicijalnog upita a zatim dodaje termine sa najvećim ponderom prvobitnom upitu, postavlja prošireni upit i konačno, dobijaju se rezultati pretrage.

Kod ručnog proširenja upita, korisnik sam dodaje termine inicijalnom upitu, dok kod proširenja upita uz asistenciju korisnika, sistem izračunava i prezentira korisniku listu potencijalnih termova za proširenje a korisnik zatim selektuje termine koje želi da koristi u proširenom upitu.

U [89] Qiu i Frei su prezentirali model probabilističkog proširenja upita korišćenjem automatski kreiranog tezaurusa sličnosti za pronalaženje term-term sličnosti. Pored toga, u istraživanju su koristili konceptualno znanje koje se odnosi na upit da pronađu odgovarajuće termine koji se dodaju originalnom upitu. Njihovi testovi na tri različita skupa podataka dali su kao rezultat poboljšanje u rasponu od 18 do 29%. Međutim, posle dodavanja većeg broja termova (~100), kod nekih skupova podataka javilo se pogoršanje kod rezultata pretrage.

Joshi i Aslandogan [90] su izložili model paralelnog konceptualnog proširenja upita. Njihova ideja je da predvide različite domene (koncepte) korisnikovog upita, odvojeno ih prošire, podnesu ih mašini za pretraživanje i vrate korisniku kategorizovane rezultate. Recimo u slučaju upita „jaguar”, paralelni model treba da kreira tri različite proširenja upita, podnese ih mašini za pretraživanje i vrati rezultate za svaki upit. U svojim testovima koristili su WordNet i njegove domene zajedno sa svojim korpusom kategorija da predvide koncepte upita. Rezultati istraživanja

pokazali su veliko poboljšanje relevantnosti rezultata. Takođe, ustanovili su da prosečno vreme potrošeno na procenu upita naglo opada. Ono što iznenađuje je konstatacija da je procena upita najkraća kada se paralelno proširenje upita koristi samo sa njihovim korpusom kategorija. Međutim, važno je istaći da su se oni u svom istraživanju bavili samo sa kratkim upitima (1 ili 2 terma) i da je model najbolje radio sa upitima tipa jednog terma pošto je preciznost opadala čim je dodavan drugi term.

Vecthomova i Wang [91] istraživali su efekat bliskosti (okoline) na proširenje upita. Razmatrali su tehniku koja proširuje upite koristeći termine koji se pojavljuju na određenom rastojanju od termova upita. U radu nije dat jasan zaključak, ali je navedeno da ova tehnika može da poboljša rezultate. Međutim, rastojanje je od velike važnosti i maksimalnu vrednost treba pažljivo birati.

Analiza upita

U suštini, procesor upita treba da koristi ista pravila i korake kao procesor dokumenta. Ustvari, potrebno je da primenjuje ista pravila kao procesor dokumenta kako bi termini upita imali istu formu kao termini u indeksu. Na primer, ako procesor upita ne radi podsecanje reči a procesor dokumenta to radi, tada upit koji se sastoji od terma „*computer*” neće biti pronađen iako ga ima u dokumentu pošto ga je procesor dokumenta sveo na „*compute*”, indeksirao ga u tom obliku i kao takav postoji u indeksu.

5.6 Rangiranje relevantnih rezultata

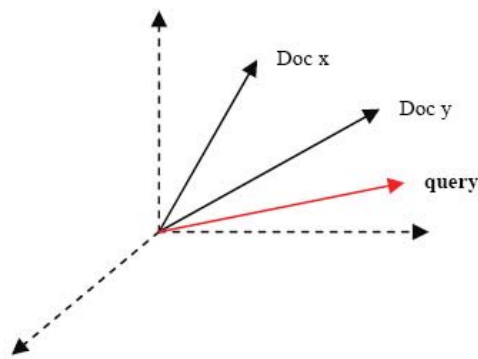
Kadgod korisnik postavi upit mašini za pretraživanje, ona vraća sortirani spisak rezultata. Taj spisak (lista) je ono što algoritam rangiranja u mašini za pretraživanje vidi kao najrelevantnije rezultate. Kod mašina za pretraživanje ti algoritmi rangiranja mogu mnogo da se razlikuju po dizajnu i performansama.

U praksi, danas se uglavnom koriste tri strategije kod rangiranja rezultata pretraživanja i to analiza linkova, model vektorskog prostora i povratna informacija o relevantnosti. Pošto se ovaj rad bavi desktop aplikacijom, to se ovde analiza linkova neće razmatrati i biće dat kratak osvrt samo na model vektorskog prostora.

Kratak osvrt na model vektorskog prostora

Ključna ideja kod modela vektorskog prostora je reprezentovanje svega (dokumenata, polja, termova ili upita) kao vektora u višedimenzionalnom prostoru. Reprezentacija upita i dva dokumenta prema njihovim težinskim članovima u vektorskom prostoru data je na slici 5.2.

Kod ovog modela može se realizovati ponderisanje termova, izračunavanje sličnosti dokumenata i rangiranje dokumenata. U procesu pronalazjenja informacija vektor upita se poredi sa vektorom drugih dokumenata izračunavanjem kosinusa ugla između upita i dokumenta. Na slici 5.2, ugao između dokumenta y i upita je manji od ugla između dokumenta x i upita. To znači da je



Slika 5.2 Model vektorskog prostora

dokument y sličniji upitu nego dokument x , izračunavajući sličnost pomoću jednačine sličnosti (5.1). Prema tome, dokument y je relevantniji za navedeni upit.

Jednačina sličnosti (5.1) izračunava vrednost sličnosti između dokumenta D_i i upita Q , gde je:

$W_{Q,j}$ - ponder terma j u upitu,

$W_{i,j}$ - ponder terma j u dokumentu i ,

a imenilac se naziva faktor normalizacije.

$$\text{sim}(Q, D_i) = \frac{\sum_{j=1}^V W_{Q,j} \times W_{i,j}}{\sqrt{\sum_{j=1}^V W_{Q,j}^2 \times \sum_{j=1}^V W_{i,j}^2}} \quad (5.1)$$

Merenje sličnosti je važno u modelu vektorskog prostora zato što se koristi za pretraživanje dokumenata prema upitu i obezbeđuje rangiranje rezultata. Rangiranje se radi posle izračunavanja vrednosti sličnosti između svih vektora i upita. Time se dobijaju mnogo tačniji i relevantniji rezultati nego kod Boolovog modela. Većina savremenih sistema za pronalaženje informacija je zasnovana na modelu vektorskog prostora.

IR sistem koji koristi ovaj model, svoj rad obavlja u tri glavna koraka. To su indeksiranje, ponderisanje članova i rangiranje.

Prvi korak je indeksiranje dokumenta. Analizom (uklanjanje stop reči, svođenje reči na osnovni oblik, provera sinonima, filtriranje itd.) dobijaju se termovi dokumenta.

Drugi korak je ponderisanje indeksiranih termova. Ovde su važna dva parametra, a to su *frekvencija terma* (*tf*) i *inverzna frekvencija dokumenta* (*idf*). Frekvencija terma je broj pojavljivanja terma u dokumentu a inverzna frekvencija dokumenta je mera pojavljivanja terma u svim dokumentima.

idf se izračunava sa $\log N / f$

gde je N ukupni broj dokumenata a f je pojavljivanje terma u svim dokumentima. Kao rezultat, ponderisanje svakog terma se izračunava kao $tf * idf$.

Na kraju, posle indeksiranja i ponderisanja, funkcija sličnosti se koristi za rangiranje dokumenata. Prema datom upitu izračunava se sličnost svakog dokumenta povezanog sa upitom. Kao rezultat dobija se rangirani izlaz, od najbližijih dokumenata ka manje sličnim dokumentima.

Latentno semantičko indeksiranje je varijanta modela vektorskog prostora. Jedna od karakteristika modela vektorskog prostora je da su vektori termova ili dokumenata nezavisni jedni od drugih. Matrica prostora dokumenata, termova ili upita je veoma velika. Da bi se to prevazišlo, latentno semantičko indeksiranje teži da redukuje matricu prostora za indeksiranje dokumenata. Logično je očekivati da redukovana matrica prostora daje bolje rezultate pretrage.

Latentno semantičko indeksiranje koristi dekompoziciju singularne vrednosti, tehniku redukcije dimenzionalnosti, da bi odredilo nekorelisane, neznčajne dokumente ili vektore termova. U principu, ovo indeksiranje se koristi za identifikovanje sinonimije (reči koje imaju isto značenje) i polisemije (reč koja ima više značenja) u dokumentima. Na primer, ako dva dokumenta imaju termine „kola” i „auto”, tada se može zaključiti da su ta dva dokumenta međusobno relevantna. Stoga, za vreme pretraživanja zadatog upitom „kola”, rezultat će sadržati i dokumenta koja sadrže term „auto”.

Latentno semantičko indeksiranje je koristan metod zato što poboljšava preciznost i relevantnost rezultata pretraživanja. Međutim, njegova glavna mana je to što je računarski zahtevan, posebno kada su u pitanju velike matrice. Takođe, potrebno mu je dosta vremena za poređenje svih vektora i redukovanje dimenzija matrice.

5.7 Poređenje svojstava biblioteka za pronalaženje informacija

U razmatranje su uzete tri biblioteke: Lucene[92], [93], Egothor [94] i Xapian [95]. Generalno, radi se o besplatnim bibliotekama otvorenog koda koje su potpuno funkcionalne. Lucene i Egothor su napisani u programskom jeziku Java, dok je Xapian biblioteka napisana u programskom jeziku C++. Lucene biblioteka je portovana u više programskih jezika, između ostalog i u C#. Sve tri biblioteke mogu lako da se integrišu u sisteme za pronalaženje informacija.

Ove biblioteke se razlikuju po implementiranom modelu pronalaženja informacija. Lucene se oslanja na model vektorskog prostora (takođe obuhvata Boolov model). Ovaj model ima moćne algoritme i metode. Takođe, danas je to najviše korišćen model u različitim softverskim bibliotekama ili mašinama za pretraživanje. Egothor se oslanja na prošireni Boolov model koji podržava osnove upite Boolovog modela (AND, OR, NOT) a ima implementirana i neka proširenja koja se oslanjaju na fazi logiku i fazi operatore. Takođe, koristi neke metode modela vektorskog prostora za rangiranje indeksiranih dokumenata. Sa druge strane, Xapian je potpuno drugačija od prethodne dve biblioteke. Oslanja se na probablistički model i stoga je kompleksnija u odnosu na konkurente.

Sve do sada razvijene i raspoložive biblioteke za kreiranje aplikacija za pretraživanje mogu da barataju samo sa tekstualnim podacima. Zbog tog ograničenja implementirano je mnoštvo nezavisnih parsera dokumenata. Ovi parseri obezbeđuju konverziju različitih tipova fajlova u običan tekst. Prema tome, Lucene, Egothor i Xapian podržavaju korišćenje fajlova formata HTML, PDF, MS Word, XML itd. Takođe, specijalni parseri dokumenata za druge tipove fajlova mogu se napisati i integrisati u aplikacije za pronalaženje informacija.

Što se tiče podrške prirodnih jezika, Lucene može da analizira i indeksira dokumente na engleskom, nemačkom, francuskom, ruskom, grčkom, holandskom, češkom, japanskom, kineskom i korejskom jeziku. Egothor navodno ima univerzalni modul za analiziranje bilo kog evropskog jezika. Međutim, bitno je naglasiti da su obavljena testiranja samo na engleskom jeziku. Xapian podržava 12 jezika, engleski, danski, holandski, finski, francuski, nemački, italijanski, norveški, portugalski, ruski, španski i švedski.

Tabela 5.3 Poređenje svojstava biblioteka za pronalaženje informacija

	Lucene	Egothor	Xapian
Programski jezik	Java	Java	C++
Licenca	Besplatan otvoren kôd	Besplatan otvoren kôd	Besplatan otvoren kôd
IR modeli	Model vektorskog prostora	Prošireni Boolov model, model vektorskog prostora	Probabilistički model
Podržani fajlovi	TXT, HTML, PDF, XML, MS Word	TXT, HTML, PDF, PS, MS Word, XLS	TXT, HTML, PHP, PDF, PS
Tipovi upita	Boolov, fazi, džokeri, opseg vrednosti	Boolov, fazi, džokeri, opseg vrednosti	Boolov, fazi, džokeri
Podržani jezici za indeksiranje	Engleski, većina evropskih jezika, kineski, japanski, korejski	Univerzalni stemer za bilo koji evropski jezik	Podržano 12 evropskih jezika
Tehnike ponderisanja	Sličnost, latentno semantičko indeksiranje	Sličnost	Verovatnoća relevantnosti, povratna informacija o relevantnosti
Struktura indeksa	Invertovani indeks, inkrementalni	Invertovani indeks, inkrementalni	Invertovani indeks, inkrementalni
Smeštanje indeksa	RAM, disk	RAM, disk	RAM, disk
Indeksiranje baza podataka	DA	DA	DA
Maksimalna veličina indeksa	2 ³² dokumenata	2 ⁶⁴ dokumenata	2 ³² dokumenata

Ovi alati koriste različite tehnike rangiranja dokumenata. Lucene koristi sistem rangiranja koji se oslanja na ponderisanje termova. Takođe, raspolaže specijalnim funkcijama, (isticanje (engl.

boosting) vrednosti termova, normalizacija vrednosti pondera) koje mogu da se koriste prilikom rangiranja. Prema tome, Lucene koristi izračunavanje sličnosti modela vektorskog prostora. Osim toga, obavlja latentno semantičko indeksiranje u cilju redukovanja matrice pretrage. Egothor takođe koristi ponderisanje i tehnike rangiranja modela vektorskog prostora. Radi rangiranja sličnosti između dokumenata da bi poboljšao relevantnost dokumenata.

Xapian koristi drugačiju metodu za rangiranje dokumenata. Pošto se oslanja na probabilistički model pretraživanja, izračunava verovatnoću relevantnosti za dokumente koji se rangiraju. Kod računanja koristi tehniku povratne informacije o relevantnosti (značajnosti) za dalje poboljšanje relevantnosti u rezultatima pretraživanja. Ove operacije su vremenski i resursno zahtevne, tako da su vremena indeksiranja veća nego kod alata Lucene i Egothor.

Zajedničko za sve tri biblioteke je da podržavaju Boolove operatore (AND, OR, NOT), korišćenje džoker znakova kod pretraživanja (*: nula ili više karaktera; ?: samo jedan karakter), fazi operacije (~: pretraga po sličnosti), upite opsega vrednosti i upite fraza. Pored toga, sve imaju implementirano inkrementalno indeksiranje. Inkrementalno indeksiranje znači da se prilikom indeksiranja novih dokumenata oni smeštaju u novi fajl koji se spaja sa starim indeksom. Na taj način, aplikacija za pretraživanje može simultano da ažurira i pretražuje indekse. To je bitno svojstvo, zato što ne mora da se zaustavlja pretraživanje dok se indeksiraju novi dokumenti.

Sva tri paketa implementiraju sopstvene strukture indeksa. Generalno, indeksiraju termine dokumenta preko parova ime polja vrednost. Indeks može da se kreira u operativnoj memoriji (RAM) ili na disku. Što se tiče ograničenja koja se odnose na veličinu indeksa, Lucene i Xapian podržavaju 32-bitne operativne sisteme, dok Egothor podržava 64-bitne. Usled toga alati Lucene i Xapian mogu da indeksiraju do 2^{32} dokumenata, dok Egothor može da indeksira do 2^{64} dokumenata.

Posle razmatranja svojstava sva tri alata, zatim uvažavajući činjenicu da se radi o desktop aplikaciji koja treba da se izvršava na Windows platformi u personalnom okruženju, izabran je paket Lucene i to verzija portovana za Microsoft Net okruženje. Za razliku od konkurenata, za Lucene postoji više literature i zajednica korisnika je vrlo aktivna. Valja naglasiti da Lucene obezbeđuje složenu (engl. *compound*) strukturu indeksa. U suštini kod složene strukture indeksa rezultati indeksiranja dokumenata se spajaju u jedan indeksni fajl. Time se obezbeđuje brže pretraživanje usled minimiziranih pristupa indeksnom fajlu. Takođe, na raspolaganju je podešavanje parametara indeksa koje omogućava podešavanje procesa pretraživanja prema korisnikovim sistemskim resursima. Zbog korišćenja latentnog semantičkog indeksiranja veličina indeksa se dodatno minimizuje.

6. Dizajniranje aplikacije

U fazi dizajniranja prvo se donose neke odluke. One se odnose na izbor programskog jezika koji će se koristiti u aplikaciji, zatim softverske biblioteke koja treba da implementira metode indeksiranja i pretraživanja, parsere dokumenata koji treba da omoguće da se informacije efikasno uskladište u pogodnom obliku i efikasno pronađu, izbor obeležja koja će dovoljno diskriminativno opisivati dokumente i omogućiti efikasno pretraživanje.

6.1 Softverska biblioteka mašine za pretraživanje – Lucene

U prethodnom poglavlju izvršeno je poređenje više softverskih biblioteka za izgradnju mašina za pretraživanje. Izbor je pao na besplatnu široko primenjivanu biblioteku Lucene, a zbog prirode okruženja, desktop aplikacija, koristiće se verzija portovana na C#.

6.1.1 Funkcionalnost biblioteke Lucene

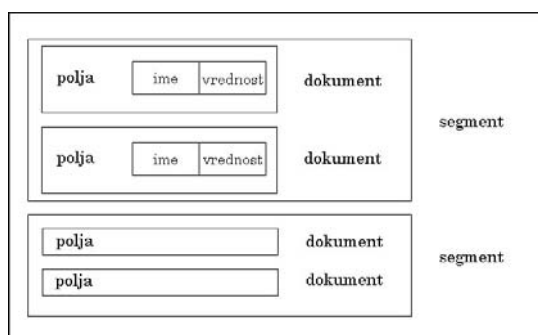
Lucene je softverska biblioteka za pronalaženje informacija koja je izvorno napisana u Javi, a postoje verzije portovane na druge programske jezike, između ostalih i za C#. Ova biblioteka obezbeđuje tzv. API (*Application Programming Interface*) funkcije za dodavanje aplikacijama funkcionalnosti potpunog tekstualnog indeksiranja i pronalaženja. Prema tome, to nije kompletno rešenje koje izvršava sve metode za implementiranje mašine za pronalaženje informacija. Ona pomaže programerima kod funkcija indeksiranja i pretraživanja da pretvore bilo koji tip podatka u tekstualnu prezentaciju, indeksira ih u strukturu indeksnog fajla i pretražuje po zadatom upitu od strane korisnika. Stoga, logika aplikacije koja se odnosi na upravljanje indeksiranjem, pretraživanje i prihvatanje upita od korisnika je u nadležnosti programera.

U tradicionalnim ili mašinama za pretraživanje prve generacije indeksiranje je obavljano po ključnim rečima i njihove tekstualne reprezentne parove je određivao programer. Kod ovog tipa pretraživanja dokumenata korisnik može da koristi samo Boolove upite (AND, OR, NOT) za pretraživanje i ima neke nedostatke kao što je zavisnost od indeksa, korišćenje samo Boolovih upita i vremenska zahtevnost. Međutim, sa novim matematičkim modelima, funkcionalnosti u mašinama za pretraživanje teksta su povećane. Pored Boolovih upita, podržani su prefiksi, fraze, džokerski znaci i fazi upiti. Rangiranje dokumenata prema pojavljivanju termova koji se pretražuju ima složenu strukturu indeksa i predmet je daljih istraživanja u cilju postizanja što efikasnijeg pretraživanja i pronalaženja.

6.1.2 Indeksiranje

Jedan od glavnih koncepata kod Lucene API funkcija je indeksiranje. Indeksiranje je pretvaranje bilo kog tipa podatka u format pretraživog indeksa. Pretvaranje obavljaju analizatori. Prvo, dokumenti se pretražuju i sav nekoristan tekst kao što su tzv. stop reči, sufixi i prefiksi se odbacuju. Detaljan opis procesa analize dat je kasnije u posebnom odeljku. Kao rezultat procesa indeksiranja kreira se indeks.

Lucene indeks sastoji se od instanci Lucene klase *Document* koje definišu indeksne dokumente. Svaki dokument sadrži polja (*Fields*) koja se sastoje od parova ime i vrednost. Primer indeksa dat je na sledećoj slici.



Slika 6.1 Struktura Lucene indeksa

Struktura Lucene indeksa je poznata kao invertovani indeks. Invertovani indeks znači da se sadržaj dokumenata analizira i da se važni termini indeksiraju kao parovi ime polja i vrednost. Svako polje sadrži više termova koji pokazuju na odgovarajuće dokumente u repozitorijumu dokumenata. Invertovani indeks olakšava pronalaženje dokumenata u sistemu i koristi ga aplikacija mašine za pretraživanje. Stoga, dokumenti se traže u poljima i njihovim vrednostima.

Kao što se vidi sa prethodne slike, Lucene indeks sadrži više segmenata. Segment se kreira kada se *heap* novog dokumenta kreira i indeksira. Usled toga svaki segment ima u sebi smešteno više dokumenata. Dokumenti se sastoje od indeksiranih polja. Kao što je već rečeno, polja sadrže najmanje delove u strukturi indeksa koji su parovi ime i vrednost. Ova polja se koriste za izračunavanje težinskih faktora (pondera) i rangiranje rezultata pretrage.

Kod procesa indeksiranja uključene su osnovne Lucene klase:

- *IndexWriter*
- *Analyzer*
- *Directory*
- *Document*
- *Field*

IndexWriter je glavna klasa Lucene operacije indeksiranja. Ona kreira inicijalni indeksni fajl za željenu putanju na računaru. Kao parametar prihvata putanju, tip analizatora i Boolov parametar

(ako je vrednost Boolovog parametra *true*, tada se kreira novi indeks „od nule”, a ako je *false*, tada ga dodaje postojećem indeksu na toj putanji). *IndexWriter* je klasa koja ima samo pristup za pisanje u indeks i koristeći svoje metode korisnici mogu dodavati dokumente u indeks u cilju pretraživanja.

Klasa *Analyzer* implementira raščlanjavanje (parsiranje) sadržaja pre kreiranja fajla indeksa. Ona analizira dokumente i odbacuje tekst koji nije koristan u aplikaciji pretraživanja.

Klasa *Directory* pokazuje mesto gde je indeks kreiran. Ona ima dve podklase, *FSDirectory* i *RAMDirectory*. Programer u skladu sa svojim potrebama odlučuje koju će koristiti. *FSDirectory* egzistira u sklopu sistema fajlova a *RAMDirectory* egzistira u operativnoj memoriji računara.

Očigledno je da *RAMDirectory* ima prednost u odnosu na *FSDirectory* jer je njen indeks u operativnoj memoriji i time je brže indeksiranje i pretraživanje dokumenata u indeksu nego kada se radi sa diskom. Međutim, kada se završi pretraživanje ili izađe iz aplikacije, indeks u operativnoj memoriji će biti obrisani. Prema tome, važno je da se indeks čuva na disku za dalja korišćenja.

Klasa *Document* reprezentuje polja u indeksu. Sastoji se od polja koja čuvaju parove ime vrednost. Kod biblioteke Lucene originalni dokument ili meta-podatak (kao što je naslov, datum kreiranja, autor dokumenta) može se povezati sa ovim poljima tako da se pretraživanje može obaviti efikasnije. Takođe, u cilju indeksiranja bilo kog dokumenta, dokument mora biti u tekstualnom formatu ili pretvoriv u tekst.

Finalna klasa indeksiranja je *Filed*. Ona reprezentuje osnovnu strukturu polja u indeksu. Svako od polja prikazuje informacije o dokumentima koji su sa njima povezani i ona se pretražuju u toku operacije pretrage indeksa.

Lucene ima četiri tipa polja koja se mogu koristiti u zavisnosti od potreba aplikacije. To su *Keyword*, *UnIndexed*, *UnStored* i *Text*. U sledećoj tabeli navedeni su tipovi polja i njihova svojstva.

Tabela 6.1: Lucene tipovi polja i svojstva

Tip polja	Analizirano	Indeksirano	Uskladišteno
Keyword		*	*
UnIndexed			*
UnStored	*	*	
Text	*	*	*

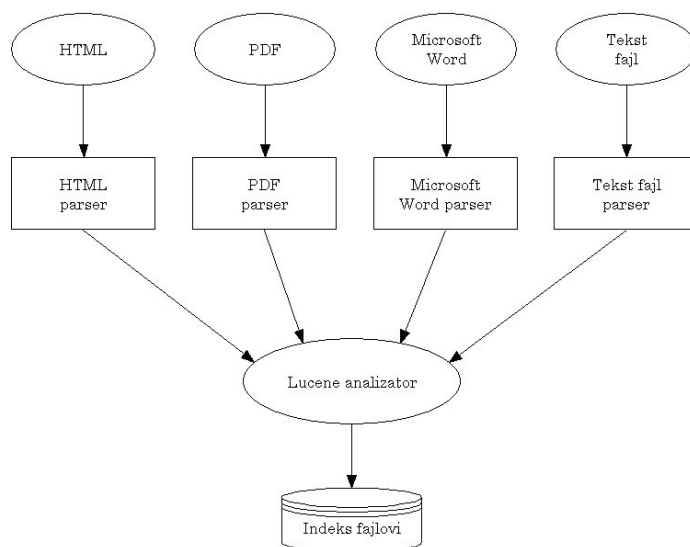
Tip polja *Keyword* se koristi za indeksiranje bilo kog teksta koji je upisan u dokument. Pogodan je za korišćenje u slučajevima kada se želi indeksiranje dokumenata uz zadržavanje originalnih vrednosti. Na primer, naslovi, URL-ovi, datumi, lična imena ili putanje dokumenata. Kao rezultat rada, kod polja tipa keyword podaci se ne analiziraju ili tokenizuju, već se indeksiraju i smeštaju.

Tip polja *UnIndexed* se koristi za, kao što njegovo ime nagoveštava, indeksiranje teksta koji neće biti analiziran niti indeksiran, već smešten u indeks. Ovo je pogodno kada se ne želi direktno

pretraživanje dokumenta ali se želi prikazivanje dokumenta u rezultatima pretrage. Mana ovog tipa polja je smeštanje dokumenta kao celine, što u slučaju velikih dokumenata utiče na rast veličine indeksa.

Polje tipa *UnStored* je suprotnost polju *UnIndexed*, tj. ono se analizira i indeksira ali se originalni sadržaj dokumenta ne smešta u indeks. Ovaj tip polja pogodan je za sadržajno velika dokumenta, kao što su „tela” HTML stranica ili bilo koji tekstualni fajl.

Polje tipa *Text* se uglavnom koristi kada korisnik želi da analizira, indeksira i smesti dokumente u svoju aplikaciju. Ovaj tip polja je pogodan za indeksiranje tekstualnih podataka male veličine kao što su naslovi, opisi ili tematika.



Slika 6.2 Glavni koraci procesa indeksiranja

Proces indeksiranja počinje čitanjem dokumenata (u ovom slučaju poslovnih dokumenata) i raščlanjavanjem (parsiranjem) njihovog sadržaja. Raščlanjavanje se radi za svaki tip dokumenata koji je pretvoriv u tekst. U principu, Lucene može da razlaže samo tekstualne podatke, međutim za druge tipove fajlova (PDF, MS Word, XML itd.) postoje specijalizovani parseri. Spisak parsera može se naći u Lucene dokumentaciji. Takođe, korisnik može da implementira svoje parsere dokumenata konvertovanjem bilo kog tipa podatka u tekst.

Posle raščlanjavanja dokumenata nastupa faza analize. Glavni koraci kod analize su svođenje reči na osnovni oblik, uklanjanje opštih reči, provera sinonima, ponderisanje dokumenata i rangiranje rezultata. Na kraju procesa analize kreira se Lucene indeks koji je osnovna struktura za mehanizam pretraživanja.

6.1.3 Pretraživanje

Proces pretraživanja je drugi glavni korak kod servisa pretraživanja informacija. Posle indeksiranja zahtevanih dokumenata, u cilju njihovog pretraživanja treba da se razviju metodi pretraživanja. Korisnik aplikacije zadaje upite koje raščlanjava parser aplikacije a zatim se vraćaju „pogoci”

(rezultati pretrage) iz memorisanog indeksa i koriste za prikazivanje rezultata korisniku.

Lucene ima četiri glavne klase koje se odnose na pretraživanje:

- IndexSearcher
- Query
- Term
- Hits

IndexSearcher se koristi za pretraživanje dokumenta iz indeksa. Ona otvara indeks u režimu samo za čitanje i koristi svoje metode da bi vratila rezultate pretraživanja. Rezultati su zatim spremni za izlaz, listanje ili sortiranje.

Klasa *Query* se koristi za definisanje korisnikovih upita. Lucene podržava više tipova upita, kao što su *BooleanQuery*, *FilteredQuery*, *MultiTermQuery*, *PhrasePrefixQuery*, *PhraseQuery*, *PrefixQuery*, *RangeQuery*, *SpanQuery* i *TermQuery*. Svi ovi tipovi upita mogu se koristiti kod pretrage ili ručnim kreiranjem ili ih Lucene klasa *QueryParser* može automatski pozvati i odrediti kom tipu korisnikovog upita pripadaju. Backus-Naur Form notacija gramatike Lucene upita je sledeća:

Query ::= (Clause)*

Clause ::= [“+”, “-”][<TERM>”:”](<TERM> (“ (” Query “)”))

Ovde <TERM> opisuje u kom indeksnom polju će termini biti traženi kao što je „Title: Java”. Oznaka (+) pokazuje da je klauzula sadržana a (-) da nije sadržana u kriterijumu pretrage.

Osnovna jedinica u upitu pretrage je klasa *Term*. Ona predstavlja bilo koji tekst u dokumentu tokom pretrage. Konstruktor ima dva parametra, jedan je polje u kojem će tekst biti tražen a drugi je sam tekst. Koristi se za konstruisanje korisnikovog upita.

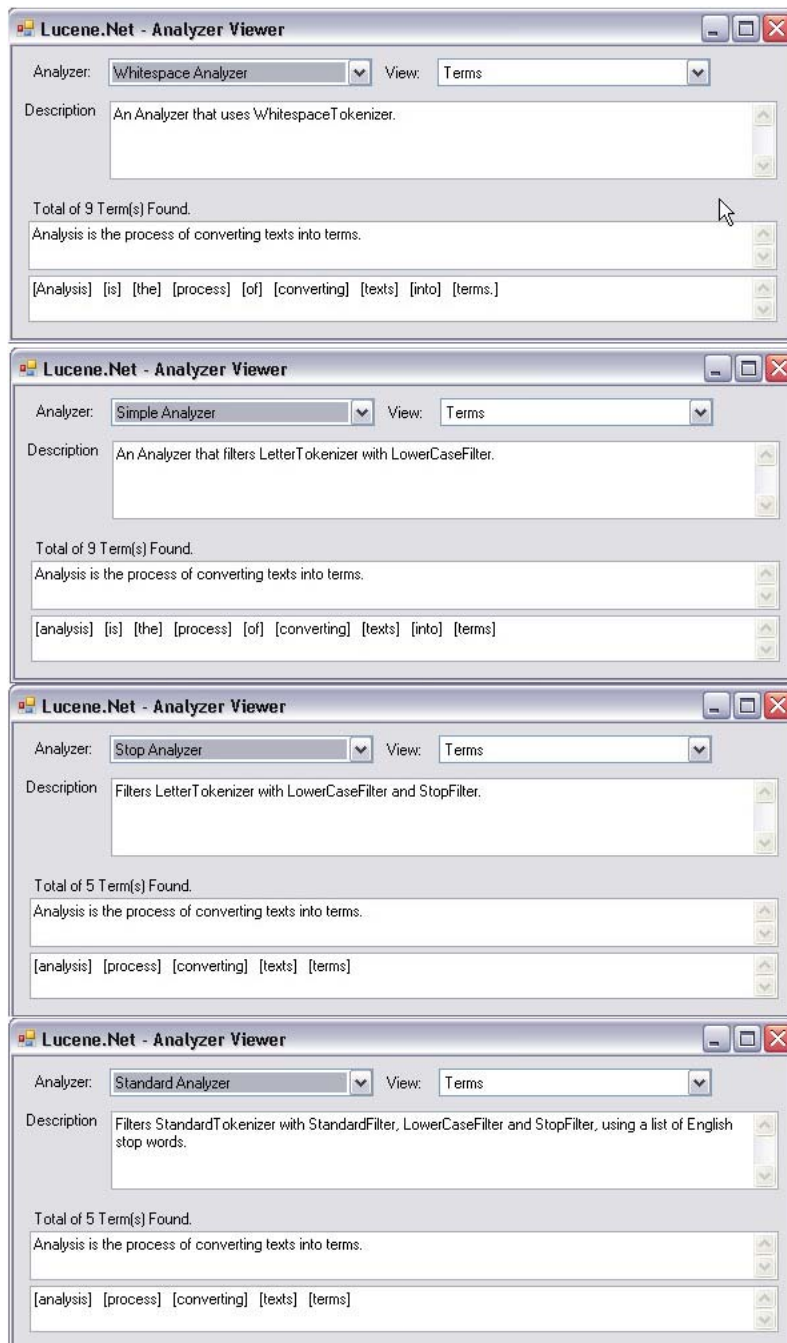
Posle konstruisanja upita pomoću klase *Query* i *Term*, klasa *IndexSearcher* pretražuje dokumente iz indeksa. Pronađene dokumente pokazuje klasa *Hits*. Rezultati se dobijaju u obliku rang liste. Implementiranjem klase *Hits*, korisnik dobija pronadene dokumente, njihov skor i ukupan broj dokumenata.

6.1.4 Analiza

Analiza je proces pretvaranja tekstova dokumenata u osnovne i indeksibilne termine. Ovde se izvode koraci tokenizacije u koje spadaju svodenje reči na osnovni oblik, odbacivanje opštih i nekorisnih reči, normalizacija (pretvaranje ispisa teksta u ispis malim slovima) i lematizacija. Takođe, u analizi se radi ponderisanje i rangiranje termina dokumenta.

Za svaku vrstu reči moguće je dogovorno odrediti kanonski, polazni oblik - lemu. Lema neke imenice je njen nominativ jednine, a lema glagola infinitiv. Na primer, neki od oblika leme hodati su: hodam, hodao, hodamo, itd. Ponekad nije moguće jednoznačno odrediti lemu neke reči. Na primer, reč vode ima sledeće leme: vod, voda, voditi. Postupak nalaženja leme zovemo lematizacija.

Osim toga, u biblioteci Lucene postoji više različitih tipova analizatora. To su *GermanAnalyzer*, *RussianAnalyzer*, *SimpleAnalyzer*, *StopAnalyzer* i *WhiteSpaceAnalyzer*. Dakle, važno je da se izabere pravi analizator za konkretnu aplikaciju. Na sledećoj slici prikazano je kako različiti tipovi analizatora obezbeđuju različite izlaze u toku analiziranja teksta.



Slika 6.3 Poređenje različitih analizatora biblioteke Lucene

WhiteSpaceAnalyzer deli tekst prema belinama (engl. *whitespaces*) u tekstu. Dakle, svaki deo se indeksira kako je napisan, što je pokazano na prethodnoj slici.

StopAnalyzer prvo deli tekstove na mestima neslovnih karaktera a zatim ih pretvara u ispis malim slovima. Na kraju uklanja tzv. stop reči koje pripadaju korišćenom jeziku.

SimpleAnalyzer je sličan *StopAnalyzer*u, tj. deli tekstove kod neslovnih karaktera a zatim ih pretvara u ispis malim slovima ali ne uklanja stop reči iz tekstova.

StandardAnalyzer može se smatrati kompozicijom prethodno objašnjenih analizatora. On obavlja sve operacije koje *WhiteSpace*, *Stop* ili *SimpleAnalyzer* rade. Sa druge strane, on izvršava specijalne operacije u skladu sa povezanom gramatikom jezika. On može da prepozna skraćenice, e-mail adrese ili specijalne reči. *StandardAnalyzer* je najčešće korišćeni analizator kod indeksiranja i pretraživanja.

6.1.5 Jednačina ponderisanja biblioteke Lucene

Važno je shvatiti kako radi algoritam vrednovanja (engl. scoring) biblioteke Lucene. On obavlja različite operacije u cilju indeksiranja i rangiranja dokumenata. Ove težinske vrednosti određuju koji dokumenti su značajni (relevantni) za dati upit. Vrednosti vrednovanja (skora) su između 0 i 1. Ako je najviše vrednovanje veće od 1, sva vrednovanja se normalizuju prema toj vrednosti. Prema tome, klasa *Hits* uvek vraća vrednosti između 0 i 1, što znači da je 1 najrelevantniji dokument a 0 najmanje relevantan dokument za zadati upit. Jednačina ponderisanja Lucene biblioteke za upit (q) i dokument (d) je:

$Score(q, d) =$

$$\frac{\sum_{t \text{ in } q} tf(t \text{ in } d) * idf(t) * getBoost(t, field \text{ in } d) * lengthNorm(t, field \text{ in } d) * coord(q, d)}{queryNorm(q)}$$

U ovoj jednačini:

$tf(t \text{ in } d)$ je frekvencija terma (t) u dokumentu (d);

$idf(t)$ je inverzna frekvencija dokumenta terma (t);

$getBoost$ vraća tzv. boost vrednost polja terma u dokumentu, što se izračunava i postavlja u toku indeksiranja;

$lengthNorm$ funkcija izračunava normalizaciju za polje terma u dokumentu;

$coord(q, d)$ izračunava vrednost koordinate prema termovima upita (q) koje dokument (d) ima.

Kod biblioteke Lucene važi da „prisustvo većeg dela termova upita pokazuje bolje poklapanje sa upitom, stoga implementacije ove metode obično vraćaju veće vrednosti kada je odnos između ovih parametara veći a manje vrednosti kada je odnos između njih manji”;

$queryNorm(q)$ izračunava vrednovanje (skor) normalizacije upita za upit koji je suma kvadrata vrednosti pondera termova (termova pretraživanih iz datog upita).

6.1.6 Parseri dokumenata

Ako programer želi da indeksira dokument, dokument mora biti u tekstualnom obliku ili pretvoriv u tekst. Prema tome, moraju postojati procesori dokumenata koji indeksiraju ne-

tekstualne podatke kao što su MS Word, PDF, PPT, XML fajlovi i drugi.

Lucene nema ugrađene uobičajene parsere dokumenata kao neke druge biblioteke. Ova biblioteka se fokusira na funkcionalnosti koje se tiču indeksiranja i pretraživanja u aplikacijama pronalaznja informacija. Međutim, postoje nezavisni komercijalni i besplatni alati koji se mogu lako integrisati u takve opšte opsluživače dokumenata. Na primer, XML podaci se mogu raščlanjavati koristeći Jakarta Commons Digester, PDF koristeći PDFBox [96], HTML koristeći Jtidy, MS Word koristeći Jakarta POI [97] itd.

6.2 Procesiranje tekstualnih dokumenata

6.2.1 Razvoj elektronskog zapisa teksta

Prvi računari bili su pravljeni pretežno za englesko govorno područje i imali su podršku samo za engleski alfabet, za brojeve, zagrade i još po neki kontrolni znak, što je činilo ukupno 128 mogućih slova (u 7 bita). To je bio tzv. ASCII ili US-ASCII standard. Kasnije je skup znakova proširen na 256 (8 bita), a „gornjih” 128 mesta je bilo korišćeno za dodatne znake. Iz navike je i ovaj prošireni ASCII nazivan ASCII, tako da tu često dolazi do zabune. Da bi postojala podrška za više jezika, smišljane su tzv. kodne strane (engl. Code Pages) koje definišu ponašanje tog dodatnog skupa slova. Osnovna kodna strana na personalnim računarima (PC437) u tom gornjem skupu znakova definiše razne grafičke znake za crtanje tekstualnih prozora i slično. Kasnije je razvijeno još puno kodnih strana koje podržavaju određene jezike. Tako postoje Latin1 (ISO-8859-1) za latinična pisma Zapadne Evrope (Francuska, Nemačka, Španija, ...), Latin2 (ISO-8859-2) i Windows-1250 za latinična pisma Istočne Evrope (naša latinica i sl.), ISO-8859-5, KOI8-R i Windows-1251 za ćirilicu Osnovni problem s kodnim stranama je to što se međusobno isključuju, tj. ceo dokument mora da bude napisan istim pismom. To uglavnom nije teško realizovati, ali ako bi bilo potrebno pomešati dva pisma, kao naprimer u nekom turističkom vodiču gde zajedno postoje tekst na srpskom, na engleskom i na francuskom, nailazi se na problem. Zbog toga se došlo do ideje da se napravi jedinstveni zapis za sve jezike – Unicode.

Pregled kodnih sistema za prikazivanje karaktera

Kodiranje podrazumeva proces prikaza informacija u nekom obliku. Jezik ljudi je kodni sistem kojim predstavljamo informacije kao nizove leksičkih jedinica, a njih pak kao zvukove odnosno mimiku. Pisani jezik je izvedeni sistem kodiranja kojim te nizove leksičkih jedinica, zvukova i gestova predstavljamo grafičkim simbolima koji sačinjavaju neki sistem pisma.

U računarskim sistemima, kodiramo pismo na način da predstavljamo grafeme i druge elemente pisanog teksta kao nizove karaktera (znakova), jedinica tekstualne informacije unutar nekog sistema za prikaz teksta. Ovi znakovi su u računar prikazani na jedini način sa kojim računarski sistemi znaju baratati: binarnim brojevima. Kodni sistemi za prikaz karaktera rade upravo to.

Svako kodiranje uključuje dve osnovne komponente: niz karaktera (znakova) i nekakav sistem za njihovo prikazivanje u obliku obradivih jedinica unutar računara. Naravno, ne postoji predodređen način na koji se ovo radi. ASCII standard je jedan od sistema, ali ne i jedini. Na taj način, isti

niz brojeva može se protumačiti na beskonačno mnogo načina, zavisno od konvencija koje se pretpostavljaju. U najjednostavnijem slučaju, koji je još uvek najrašireniji, jedan oktet bitova (bajt) odgovara jednom karakteru prema nekoj kodnoj tabeli. Ovako je moguće prikazati 256 različitih karaktera. Za pravilan način dekodiranja, mora biti poznato koja tabela se koristi.

Dok je ranije ASCII kodiranje bilo najučestalije, danas je raširen *ISO Latin 1* način kodiranja, koji se može smatrati ekstenzijom ASCII-a. ASCII je bio dovoljan samo za prikaz teksta na američkom engleskom. Nije bio čak dovoljan za britanski engleski, a kamoli za izdavaštvo na engleskom govornom području ili gotovo bilo kom drugom jeziku. Ubrzo su prodrli novi standardi i to iz dva izvora: ustanova za standardizaciju i nezavisni proizvođači softvera i hardvera. Tako su IBM (*codepage 437, 852, 855*), Apple i Microsoft (*codepage 1252 = Western = Latin 1*) stvarali standarde usput kako su im zatrebali za proizvode koje su izdavali. Ako bi neki takav standard postao široko prihvaćen, nacionalno standardizacijsko telo ga je moglo proglasiti nacionalnim standardom, a takvi su standardi ponekad postajali i međunarodni. Uz mnoštvo ovako nastalih standarda, događalo se da neki standardi podržavaju određene znakove koji ne postoje u drugim.

Takođe se dešavalo da su podržani isti setovi karaktera, ali na nekompatibilan način. Ovi problemi se danas rešavaju Unicode standardom, ali nasleđeni standardi se ne mogu jednostavno zanemariti.

Svaki komercijalni softverski proizvod je eksplicitno dizajniran da podržava određeni set standarda za kodiranje karaktera. Sve njegove operacije sa tekstom biće izvedene pod pretpostavkom da je aktivan jedan od tih standarda. Ako dakle podaci nisu tačno ili uopšte označeni kojim standardom su kodirani, doći će do neočekivanih rezultata.

Problem sa softverom baziranom na standardima se javlja ako je potrebno raditi sa kompletom karaktera koji softver ne prepoznaje. Ovo je posebno bio problem za lingviste koji rade sa manje zastupljenim jezicima. Takvi onda kreiraju vlastita rešenja za svoje potrebe koja su sasvim dobra za vlastitu primenu, ali zakazuju kada dođe do potrebe za razmenom na višem nivou. Način za izbegavanje ovih problema je upotreba standardnog kodiranja koje uključuje sve znakove. Upravo takav tip rešenja nudi *Unicode* koji se razvija sa ciljem da ima univerzalni skup karaktera koji bi pokrivao potrebe na svetskom nivou.

Kodni sistem ASCII (ISO 646)

American Standard Code for Information Interchange je oznaka za repertoar karaktera, skup kodiranih karaktera i obrazac za kodiranje. Većina postojećih kodova sadrže ASCII kao svoj podskup u nekom smislu. Naziv ACSII se koristi naširoko i često pa se tako ponekad misli na tekst uopšteno, ili tekstualnu datoteku za razliku od binarne.

ASCII po definiciji sadrži i skup kontrolnih kodova, ali pravi repertoar karaktera koji se sastoji od ispisivih znakova je sledeći:

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Slika 6.4 Tabela ispisivih ASCII karaktera

Skup kodiranih karaktera definisan ASCII standardom je sledeći: kodne vrednosti su dodeljene karakterima redom koji je naveden prethodno, počevši od 32 za prazninu (belinu; razmak) i završavajući sa 126 (tilda). Mesta 0 ÷ 31 i 127 su rezervisana za kontrolne kodove koji imaju standardizovana imena i opise, ali im primena varira.

	8	9	A	B	C	D	E	F
0	Ç	É	á	█	Ł	⌌	α	≡
1	ù	æ	í	█	⌋	⌏	ß	±
2	é	Æ	ó	█	⌏	⌏	Γ	≥
3	â	ô	ú		⌏	⌌	π	≤
4	à	ö	ñ	⌏	—	Ł	Σ	∫
5	à	ò	Ñ	⌏	⌏	⌏	σ	∫
6	â	û	ª	⌏	⌏	⌏	μ	÷
7	Ç	ù	º	⌏	⌏	⌏	τ	≈
8	ê	ý	¿	⌏	⌌	⌏	Φ	°
9	ë	ÿ	⌏	⌏	⌏	⌏	Θ	•
A	è	Û	⌏	⌏	⌌	⌏	Ω	•
B	ï	ç	¾	⌏	⌏	█	ö	√
C	î	é	¾	⌏	⌏	█	∞	²
D	ì	¥	¿	⌏	=	█	φ	²
E	Ä	É	«	⌏	⌏	█	ε	■
F	Å	f	»	⌏	⌏	█	π	

Slika 6.5 Tabela ASCII karaktera

Obrazac za kodiranje je vrlo jednostavan i prilično očigledan za sve kodove znakova gde kodni brojevi ne prelaze 255: svaki kodni broj je predstavljen kao oktet (bita) iste vrednosti. Okteti 128 ÷ 255 se ne koriste u ASCII-u.

ISO Latin 1 (ISO 8859-1)

ISO 8859-1 standard (koji je deo ISO 8859 porodice standarda) definiše repertoar karaktera koji se naziva „Latin alphabet No. 1”, obično zvan „ISO Latin 1”, te skup kodiranih karaktera. Repertoar sadrži ASCII kao podskup i kodni brojevi za te karaktere su isti kao u ASCII-u. Standard specificira i obrazac za kodiranje koji je sličan onome u ASCII-u: svaki karakter je predstavljen oktetom.

Kao dodatak ASCII karakterima, ISO Latin 1 sadrži različite karaktere sa naglascima i druga slova potrebna za pisma zapadne Evrope i neke posebne znakove. Ovi znakovi zauzimaju mesta $160 \div 255$ i to su:

WINDOWS skup karaktera

Tokom razvoja Windowsa Američki Nacionalni Institut za Standarde (ANSI) je razvijao standard koji je postao ISO 8859-1 „Latin”. Microsoft je kreirao svoju kodnu stranicu 1252 za zapadno evropske jezike na osnovu ranog nacrtu ANSI predloga i to nazvao „ANSI kodna stranica”. Kodna stranica 1252 je finalizovana pre nego ISO 8859-1 i to dvoje nisu isti: 1252 je širi od ISO 8859-1. Kasnije je Microsoft počeo da koristi „ANSI” u smislu bilo koje Windows kodne stranice, za razliku od Unicode.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8			,		„	...	†	‡		‰	Š	◁	Ś	↑	Ž	Ž
9		‘	’	“	”	•	—		™	š	▷	ś	ť	ž	ž	
A		˘	˘	Ł	◻	Ą	ı	§	"	©	§	«		—	®	Ž
B	°	±	ˆ	ı	’	μ	¶	·	,	ı	ş	»	Ł	"	ı	ž
C	Ř	Á	Â	Ă	Ă	Ĺ	Ć	Ç	Č	É	Ę	Ě	Ě	Í	Î	Ď
D	Đ	Ñ	Ñ	Ó	Ô	Ö	Ö	×	Ř	Ů	Ú	Ů	Ů	Ý	Ť	ß
E	ř	á	â	ă	ă	ĺ	ć	ç	č	é	ę	ě	ě	í	î	ď
F	đ	ñ	ñ	ó	ô	ö	ö	÷	ř	ů	ú	ů	ů	ý	ť	·

Slika 6.6 Tabela karaktera Win-1250

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	Ђ	Ѓ	„	џ	…	†	‡		‰	Љ	«	Њ	Ќ	Ѝ	Ў	
9	ђ	ѓ	’	“	”	•	—		™	љ	»	њ	ќ	ћ	ў	
A		Ÿ	ŷ	Ј	□	Г	І	§	Ё	©	€	«	¬	–	®	Ї
B	°	±	І	і	г	μ	¶	·	ё	№	€	»	ј	ѕ	ѕ	ї
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Slika 6.7 Tabela karaktera Win-1251

U ISO 8859-1 standardu, kodne pozicije 128 ÷ 159 su eksplicitno rezervisane za kontrolne svrhe, tj. odgovaraju kombinacijama bitova koje ne predstavljaju grafičke karaktere. Takozvani Windows skup karaktera (WinLatin1 ili Windows kodna stranica 1252) koristi neke od tih pozicija za ispisive karaktere. Windows skup karaktera se često naziva „ANSI skup karaktera”, ali ovo je pogrešno jer nije odobren od ANSI-a. Korišćenje kodnih karaktera pod brojevima 128 ÷ 159 u programu koji očekuje pravi ISO 8859-1 može izazvati različita ponašanja. Može se desiti da se ti karakteri ignorišu, da izgledaju kao nešto što ima smisla ili da se interpretiraju kao kontrolni karakteri.

ISO 8859 porodica

Postoji nekoliko sistema kodova karaktera koji su ekstenzija ASCII-a na isti način kao što su to ISO 8859-1 i Windows skup karaktera. Sam ISO 8859-1 je deo veće porodice znakovnih kodova ISO 8859 čiji kodovi proširuju repertoar ASCII-a na različite načine sa različitim posebnim karakteristikama (koji se koriste u raznim jezicima i kulturama). Baš kao što ISO 8859-1 sadrži ASCII karaktere i kolekciju karaktera potrebnih za jezike zapadne (i severne) Evrope, tu je i ISO 8859-2 alias ISO Latin 2 koji je konstruisan slično za jezike središnje/istočne Evrope itd.

ISO 8859 kodovi su izomorfni u sledećem smislu:

- kodni brojevi 0 ÷ 127 sadrže iste karaktere kao i ASCII
- mesta 128 ÷ 159 su neiskorišćena (rezervisana za kontrolne karaktere)
- mesta 160 ÷ 255 su promenljivi deo koji se koristi različito u različitim izdanjima

Iako se ISO 8859-1 koristio kao *de facto* standard predodređen za kodiranje u mnogim primenama, on u principu nema neku posebnu ulogu. Očekivalo se da će ISO 8859-15 alias ISO Latin 9 zameniti ISO 8859-1, budući da sadrži politički važan simbol za evro AC, no čini se da ipak nema veliku praktičnu primenu.

Unicode

Unicode je standard, od strane Unicode konzorcijuma, koji definiše repertoar karaktera i skup kodiranih karaktera koji su u potpunosti kompatibilni sa ISO 10646 i obrazac za kodiranje karaktera. ISO 10646 je uopšteniji i po prirodi apstraktniji, dok Unicode „nameće dodatna ograničenja na implementaciju da bi osigurao uniforman tretman karaktera na svim platformama, programima i jezicima”.

Unicode rešava sve one probleme oko jednoznačnosti, transformacija među različitim kodnim tabelama, raspoloživog opsega karaktera koji su isticali u dosadašnjem tekstu. Unicode standard je prihvaćen od vodećih industrijskih firmi kao što su: Apple, HP, IBM, Microsoft, Oracle, SAP, Sun, Sybase, Unisys i mnogih drugih. Unicode je potreban kod modernih standarda kao što su: XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, itd. Podržan je od strane mnogih operativnih sistema i svih savremenih čitača.

Unicode je prvobitno zamišljen kao 16-bitni kôd, ali proširen je tako da su trenutne kodne pozicije predstavljene celim brojevima u heksadecimalnom rasponu $0 \div 10FFFF$ (decimalno $0 \div 1\,114\,111$). Taj prostor podeljen je u 16-bitne „ravni”. Do nedavno, upotreba Unicodea je uglavnom bila ograničena na „osnovnu višejezičnu ravan” (BMP) koja se sastoji od raspona $0 \div FFFF$.

UTF-32 kodira svaki kodni broj kao 32-bitni celi broj, tj. kao 4 bajta. Ovo je vrlo očito i jednostavno kodiranje. Sa druge strane, neefikasno je s obzirom na broj korišćenih bajtova. Ako imamo normalni engleski tekst ili neki drugi tekst koji sadrži samo znakove iz tablice ISO Latin 1, dužina kodiranog niza bajtova je 4 puta duža u Unicodeu nego u ISO 8859-1 kodu. UTF-32 se retko koristi, osim možda u internim operacijama.

UTF-16 predstavlja svaki kodni broj u osnovnoj višejezičnoj ravni kao dva bajta. Unicode se može kodirati na druge načine, kao što su sledeći:

- **UTF-8** - kodovi karaktera manji od 128 (efektivni ASCII repertoar) se predstavljaju „kao takvi”, koristeći jedan oktet za svaki kod (karakter; znak). Svi drugi kodovi su prikazani prema relativno komplikovanoj metodi, tako da je jedan karakter predstavljen kao niz od dva do četiri okteta, od kojih je svaki u rasponu $128 \div 255$. Ovo znači, da u nizu bajtova, oni koji su u rasponu $0 \div 127$ (bajtovi s MSB-om u nuli) direktno predstavljaju ASCII karaktere, dok bajtove u rasponu $128 \div 255$ (bajtovi s MSB-om u jedinici) treba interpretirati kao stvarno kodirane prezentacije karaktera.
- **UTF-7** - svaki kôd karaktera je predstavljen nizom jednog ili više okteta u rasponu $0 \div 127$. Većina ASCII karaktera je tako predstavljena, svaki po jedan oktet, ali očito je da neke vrednosti moraju biti rezervisane za upotrebu kao „okteti za beg”, koji govore da oktet zajedno s određenim brojem okteta koji slede formira više-oktetnu prezentaciju jednog karaktera. UTF-7 se vrlo malo koristi, dok se UTF-8 favorizuje zbog efikasnosti.

Grafem - vizuelni prikaz

Važno je razlikovati koncept karaktera od koncepta grafema (*glyph*). Grafem je grafička reprezentacija konkretnog oblika koji karakter može imati kada se ispiše ili prikaže. Na primer, karakter Z se može prikazati kao podebljani Z ili iskošeni Z, ali to je još uvek reprezentacija istog karaktera. S druge strane, malo slovo z je definisano kao zaseban karakter, koji pak opet može imati različite reprezentacije. Ovo je na kraju stvar definicije. U repertoaru karaktera samo je jedno veliko slovo Z, ali moglo bi se definisati da su Z i z dva različita grafema istog elementa repertoara.

6.3 Formati poslovnih dokumenata

Pre nego što se upustimo u dalje razmatranje problematike procesiranja elektronskih poslovnih dokumenata, potrebno je utvrditi u kom se elektronskom formatu najčešće ti dokumenti pojavljuju. Jednostavnom analizom može se utvrditi da su to fromati DOC (MS Word), PDF i RTE.

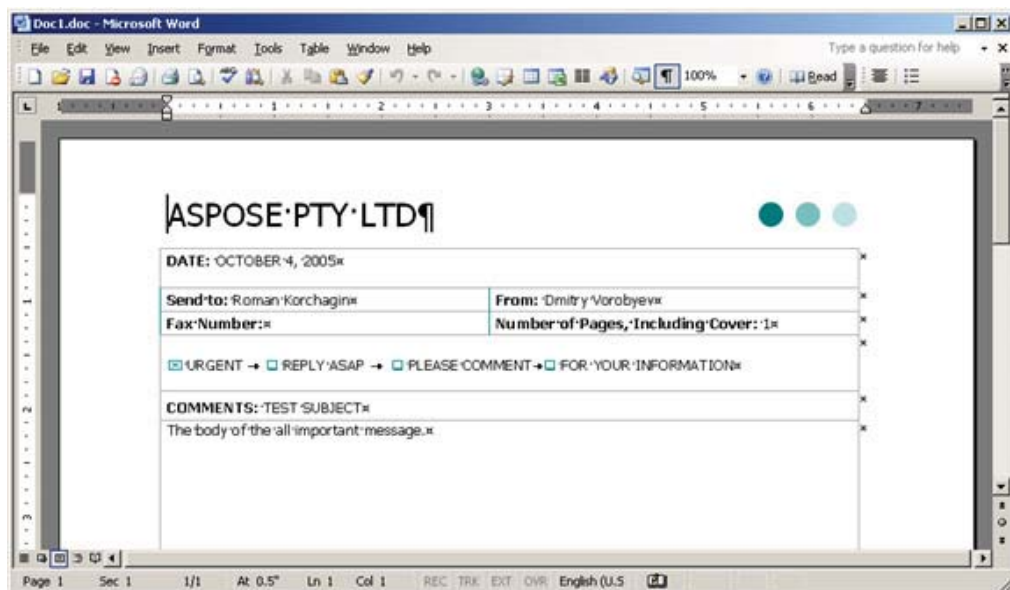
Svaki tip dokumenta ima drugačija pravila za svoju strukturu. Na primer, pravila za formatiranje ispisa „Bold” i „Not Bold” ne mogu se istovremeno primeniti na jedno slovo u Wordovom dokumentu. Kada se koristi takva struktura za programsko menjanje fajla, potrebno je omogućiti primenu svih tih pravila intuitivno ili čak implicitno. Zbog svoje fizičke strukture, dokumenti su često u nečitljivom formatu za čoveka i stoga je potreban softver za njihovo prevođenje u za čoveka čitljivu strukturu. Da bi moglo programski da se obavlja to kodiranje poželjno je da dokument ima objektno orijentisanu strukturu.

Document Object Model (referencira se kao DOM) rešava te probleme uvođenjem skupa klasa koje predstavljaju sve ponuđene opcije logičke strukture dokumenta, kreirajući tako apstrakciju strukture fajla. DOM može da učita postojeći dokument i da ga predstavi grafom objekata. Taj graf objekata, takođe može da se „kreira od nule”. Takođe, DOM nudi snimanje grafa objekata nazad u format fajla ako je to urađeno programskim kodom.

6.3.1 MS Word

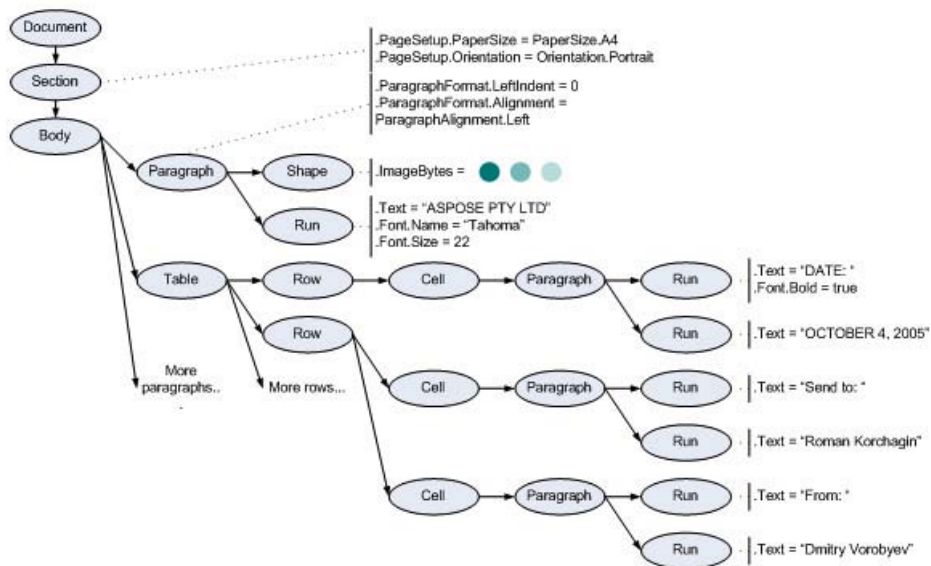
Microsoft Office je još uvek dominantan softverski paket za kancelarijsko poslovanje, rezultat toga je činjenica da je mnoštvo poslovnih dokumenata pripremljeno u MS Wordu. U problematici kojom se bavi ovaj rad, težište je na ekstrakciji teksta i slika iz ovih fajlova a manje interesovanje je za manipulisanje sadržajem.

Postoji više komercijalnih i besplatnih softverskih biblioteka koje mogu da se koriste za ekstrakciju teksta iz Wordovih dokumenata. Za razvoj ove aplikacije korišćena je Net verzija biblioteke *Apose Words* [98].



Slika 6.8 Prikaz jednog Wordovog dokumenta

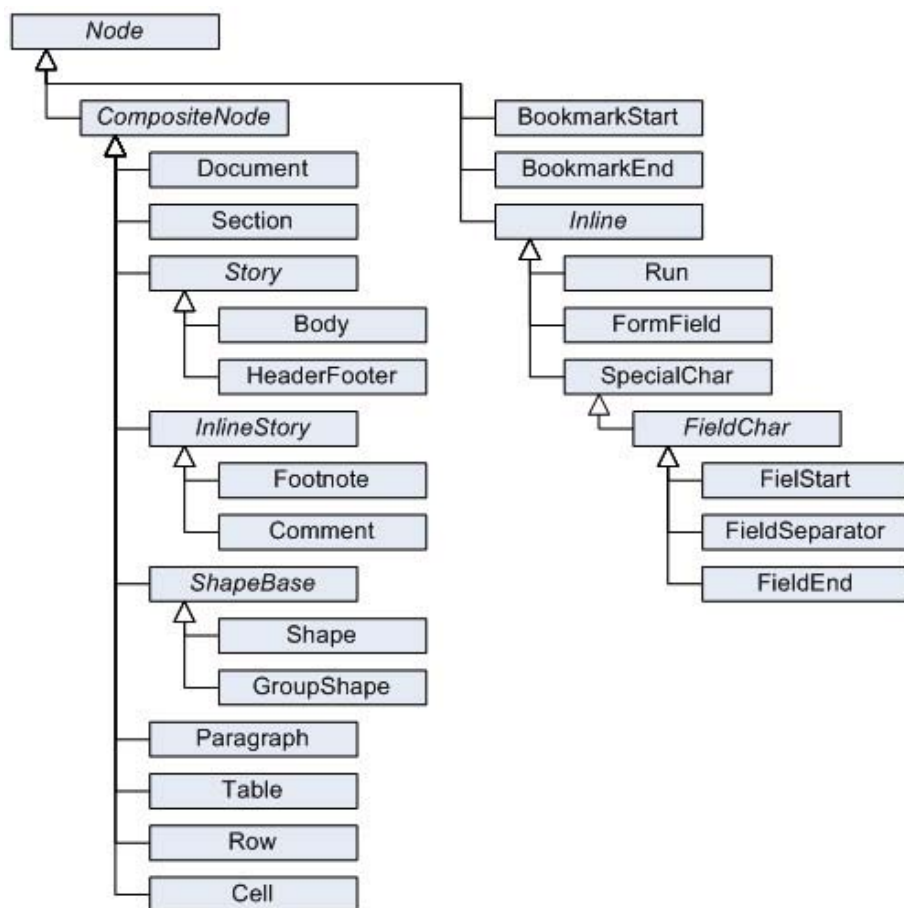
Microsoft Word dokument je složenog formata. Da bi mogla da otvori dokument biblioteka Aspose.Words sadrži klasu *Document* koja je centralna klasa biblioteke. Wordov dokument se sastoji od objekata tipa *Sections*, *Pages*, *Paragraphs*, *Tables*, *Bookmarks*, *Headers and Footers*.



Slika 6.9 Prikaz strukture Wordovog dokumenta sa prethodne slike

Postoji više načina da se pristupi tim elementima dokumenta koristeći Aspose.Words. Ova biblioteka kada učitava Wordov dokument u memoriju, kreira različite objekte da reprezentuje razne elemente dokumenta. Takođe, svako pojavljivanje teksta, pasusa, tabele, sekcije je čvor, pa

čak je čvor i sam dokument. Za svaki tip čvora u dokumentu u ovoj biblioteci postoji posebna klasa. Stoga, ideja je da se koristi najjednostavniji i najefikasniji način pristupa sadržajima preko čvorova (engl. *Nodes*).



Slika 6.10 Aspose. Words elemente dokumenta predstavlja preko čvorova

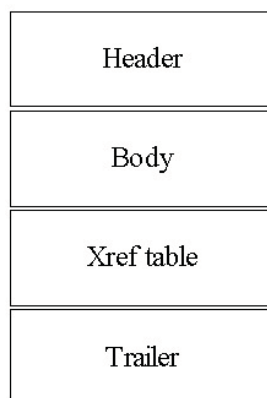
Objekt *Document* omogućava učitavanje dokumenata u više formata. U ovoj aplikaciji se koristi Microsoft Office DOC format.

6.3.2 PDF

Fajlovi formatirani pomoću Adobeovog Portable Document Formata (PDF) popularni su kao i MS Word fajlovi. PDF je razvijen od strane firme Adobe Systems 1993. godine. Radi se o univerzalnom formatu elektronske datoteke nepromenljivog izgleda koji zadržava oblikovanje dokumenta i omogućuje deljenje datoteke.

PDF se koristi za zapis dvodimenzionalnih dokumenata nezavisno od uređaja i rezolucije ispisa. Svaki PDF-dokument sadrži kompletan opis dokumenta, uključujući slike, tekst, vektorsku grafiku, rasterske slike, te može sadržati i fontove potrebne za prikaz teksta.

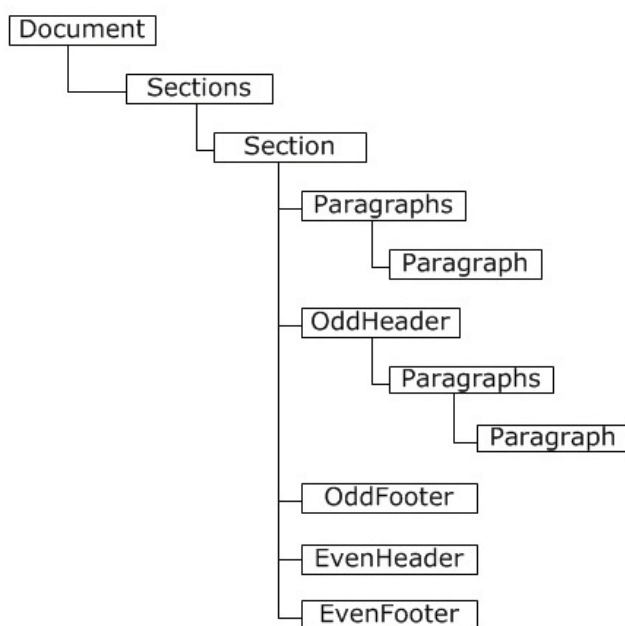
Standardna PDF datoteka se sastoji od četiri glavna dela: zaglavlje - jedan red (engl. *Header*), telo dokumenta (engl. *Body*), tabele indeksa (engl. *Xref table*) i trejlera (poslednji slog).



Slika 6.11 Osnovna struktura PDF dokumenta

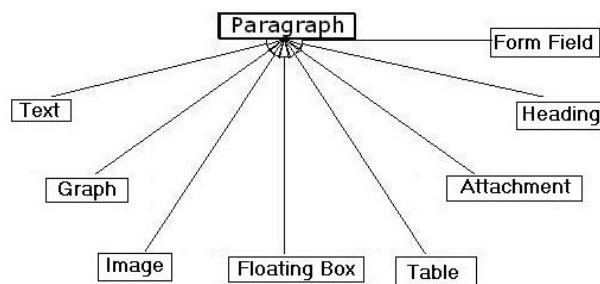
Tekst i slike iz ovih fajlova takođe mogu da se ekstrahuju korišćenjem više komercijalnih i besplatnih softverskih biblioteka. Za razvoj ove aplikacije korišćena je Net verzija biblioteke *Aspose.Pdf*.

Aspose.Pdf takođe koristi DOM da predstavi strukturu PDF dokumenta preko objekata.



Slika 6.12 Aspose.PDF DOM struktura

Na vrhu hijerarhije je objekt *Document* koji sadrži kolekciju sekcija. Svaka sekcija ima zaglavlje, podnožje i pasuse (engl. *Paragraphs*). Zaglavlje i podnožje mogu da sadrže kolekciju pasusa, a pasus može da ima različite tipove podataka koje treba PDF da prezentuje.



Slika 6.13 Struktura apstraktne klase Paragraph

Za ekstrakciju slika iz PDF dokumenata korišćena je Net verzija biblioteke *Apose.Pdf.Kit* [99], koja raspolaže klasom *PDFExtractor* za ekstrahovanje sadržaja PDF dokumenta.

6.3.3 RTF

Rich Text Format (RTF) je vlasnički format dokumenta koji je razvio Microsoft da omogući međuplatformsku razmenu dokumenata. RTF fajlovi se mogu pripremati i prepravljati u više programa za obradu teksta. Specifikacija RTF formata je na raspolaganju svima koji su zainteresovani za primenu tog formata. Obično programi koji podržavaju DOC format mogu da barataju i ovim formatom.

6.4 Procesiranje slika

6.4.1 Boja i atributi boje

Boja se može definisati kao psihofizički fenomen indukovano svetlom ili osećaj koji u mozgu izaziva svetlost emitovana od nekog izvora ili reflektovana s površine nekog tela. Doživljaj boje zavisi od tri faktora:

- spektralnog sastava svetla koje pada na posmatrani predmet,
- molekularne strukture materijala s kog se svetlo reflektuje ili koje ga propušta i
- čovekovog oseća boje, kroz oči i mozak.

Smisao za boje relativno skoro se pojavio u istoriji ljudskog roda. Deca i primitivni narodi razlikuju vrlo malo boja, ili ih barem teško imenuju. Stari narodi bili su vrlo siromašni u označavanju i imenovanju boja, ali s razvojem industrije boja i tkanina, štamparstva, kozmetike itd., razvijao se i taj smisao. Izvorno su boje nazivane prema predmetima za koje su bile karakteristične. Za prosečnog čoveka i danas boja ima svojstvo predmeta: ljubičica je ljubičasta, narandža narandžasta itd.

Boje se mogu definisati svojim uobičajenim imenima, ali i opisnim, literarnim izvedenicama, ali to ne govori ništa o njihovim izražajnim vrednostima. Stoga postoje tri atributa koji uže definišu svaku boju:

- ton boje ili tonalnost boje,
- zasićenost ili saturacija,
- svetlina ili luminancija.

Ton boje označava vrstu boje, boju samu po sebi. Definiše se kao hromatski kvalitet boje, odnosno kvalitet kojim se jedna boja razlikuje od druge. Da bi se objasnili pojmovi zasićenost i svetlina, najpre je potrebno boje podeliti u dve osnovna skupa. U prvom skupu nalaze se prave boje, kao što su crvena, narandžasta, žuta, zelena, plava itd., koje se nazivaju hromatskim bojama, ili jednostavno bojama. Slika 6.14. prikazuje hromatske boje.



Slika 6.14 Hromatske boje

U drugom skupu nalaze se crna, siva i bela, koje se nazivaju ahromatskim bojama, ili jednostavno nebojama. One čine skalu koja seže od crne, preko sive, do bele. Slika 6.15 prikazuje ahromatske boje.



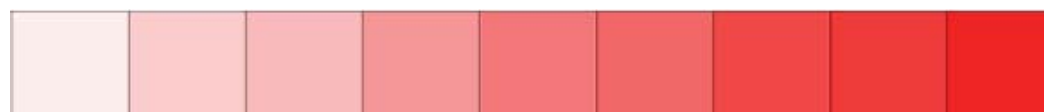
Slika 6.15 Ahromatske boje

Neke od hromatskih boja su tamnije ili svetlije od drugih i moguće je upoređivati svaki stepen njihove svetline sa svetlinom sive ahromatske boje. Ta se osobina naziva svetlina ili luminancija. To je relativna količina svetla (bilo koje talasne dužine) koju boja prividno emituje. To je kvalitet kojim se razlikuje svetla boja od tamne boje. Slika 6.16 prikazuje svetlinu boje.



Slika 6.16 Svetlina boje

Ako se neka hromatska boja meša s ahromatskom bojom jednake svetline, svetlina boje ostaje ista. Nastala promena u kvalitetu, odnosno čistoći boje, zavisi od relativne količine ovih dveju komponenata. Ta se osobina naziva zasićenost ili saturacija. To je stepen do kog se boja čini čistom. Slika 6.17 prikazuje zasićenost boje.



Slika 6.17 Zasićenost boje

6.4.2 Spektar vidljivog svetla

Svetlo je elektromagnetsko zračenje koje se opaža vizuelnim nervnim sistemom. To je energija zračenja, nastala atomskim promenama u fizikalnoj strukturi materije, koja se prostire od svog izvora u svim smerovima i širi u obliku talasa. Slika 6.18 prikazuje vidljivi spektar elektromagnetskog zračenja. Ono na što se obično misli kada se kaže „svetlo” je belo svetlo. Ono što se obično percipira kao belo svetlo nije homogeno – to je mešavina svih talasnih dužina vidljivog spektra od 400 nm do 700 nm u približno jednakim odnosima. Padne li takvo svetlo na komad belog papira, on će reflektovati sve talasne dužine, pa će i svetlo što se od njega reflektovalo izgledati belo. Ako neki predmet apsorbira sve talasne dužine, a ni jednu ne reflektuje, izgledaće crn. Apsorbira li neki predmet sve talasne dužine, osim one koja predstavlja oset crvene boje, te samo nju reflektuje, takav će predmet izazvati u čovekovim osetilima oset crvenog. Koje će talasne dužine svetla biti apsorbirane a koje reflektovane zavisi od molekularne strukture materijala na koje svetlo pada. Klasični spektar razlikuje sedam boja: crvenu, narandžastu, žutu, zelenu, plavozelenu, plavu i ljubičastu. Slika 6.19 prikazuje klasičan spektar od sedam boja. Broj boja i njihovih nijansi u prirodi je neizmeran, budući da veoma mala promena talasne dužine stvara novu i drukčiju boju.



Slika 6.18 Vidljivi spektar elektromagnetskog zračenja



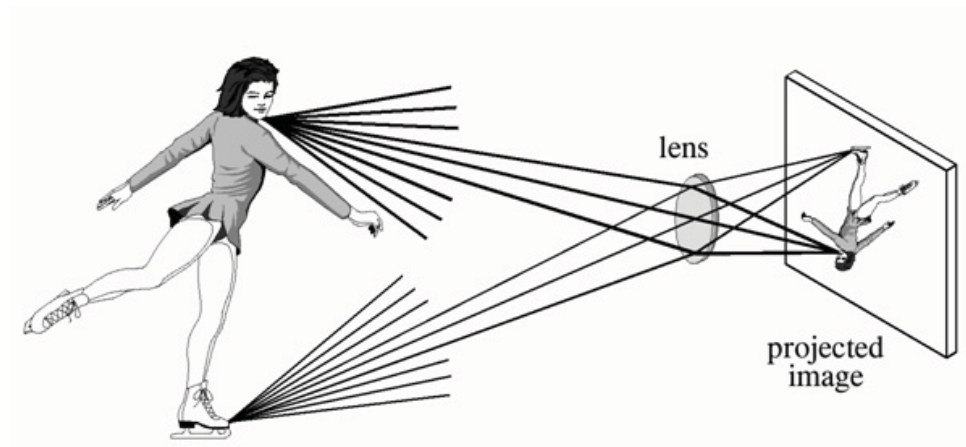
Slika 6. 19 Klasični spektar od sedam boja

6.4.3 Kamere i oči

Struktura i funkcionisanje oka su veoma slični sa elektronskom kamerom, i prirodno je da se razmatraju zajedno. Oboje se oslanjaju na dve glavne komponente: sočivo i senzor slike. Sočivo „hvata” deo svetlosti koji zrači neki objekt i fokusira ga na senzor slike. Senzor slike zatim pretvara svetlosni uzorak u video signal, ili elektronski (kamera), ili nervni (oko).

Na slici 6.20 prikazano je funkcionisanje sočiva. U ovom primeru, slika klizačice je fokusirana na zastor. Usled fokusiranja postoji preslikavanje jedan-na-jedan svake tačke klizačice u odgovarajuću tačku na zastoru. Na primer, posmatrajmo oblast veličine 1 mm x 1 mm na vrhu nožnog prsta. Pri jakom osvetljenju postoji oko 100 triliona fotona svetlosti koji pogađaju ovaj kvadratni milimetar svake sekunde. U zavisnosti od karakteristika površine, između 1 i 99% ovih fotona

upadne svetlosti će se reflektovati u raznim pravcima. Samo mali deo ovih reflektovanih fotona će proći kroz sočivo. Na primer, samo milioniti deo reflektovane svetlosti će proći kroz sočivo prečnika 1 cm, koje se nalazi na udaljenosti od 3 m od objekta.

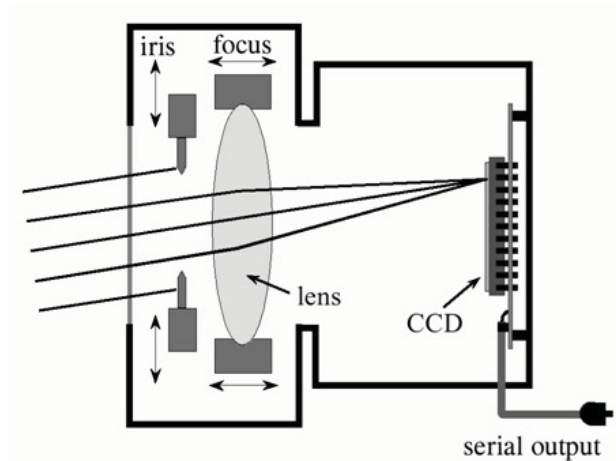


Slika 6.20 Fokusiranje sočivom. Sočivo skuplja svetlost koja se širi od tačkastog izvora i primorava je da se vrati u tačku na drugoj lokaciji. Ovo omogućava sočivu da projektuje sliku na površinu.

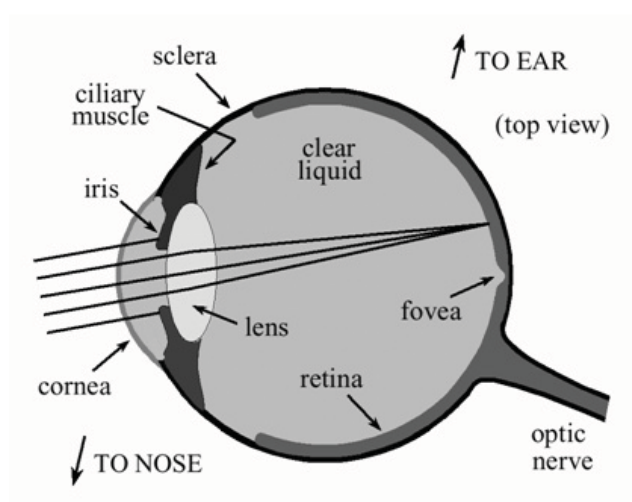
Prelamanje u sočivu menja pravac pojedinačnih fotona, u zavisnosti od lokacije i ugla pod kojim oni pogađaju graničnu površinu staklo/vazduh. Ove promene pravca izazivaju širenje svetlosti od jedne tačke i povratak u jednu tačku na projekcionom zastoru. Svi fotoni koji se reflektuju od prsta i prođu kroz sočivo, vraćaju se zajedno na „prst” u projektovanoj slici. Na sličan način, deo svetlosti koji dolazi od bilo koje tačke sa objekta proći će kroz sočivo i biće fokusiran ka odgovarajućoj tački u projektovanoj slici.

Slike 6.21 i 6.22 ilustruju strukture elektronske kamere i oka, respektivno. Oboje imaju mračnu komoru sa sočivom montiranim na jednoj strani i senzorom slike na suprotnoj strani. Kamera je ispunjena vazduhom, dok je oko ispunjeno tzv. staklastim telom (providna tečnost). Svaki sistem sočiva ima dva podesiva parametra: fokus (žiža) i prečnik irisa (dužice).

Ako sočivo nije odgovarajuće fokusirano, svaka tačka sa objekta projektovaće se kao kružna oblast na senzor slike, usled čega će slika biti zamućena. Kod kamere, fokusiranje se postiže fizičkim pomeranjem sočiva ka senzoru slike i udaljavanjem od senzora slike. Međutim, oko ima dva sočiva, jedno ispupčeno na vrhu očne jabučice koje se zove rožnjača, i podesivo sočivo unutar oka. Rožnjača obavlja većinu prelamanja svetlosti, ali je fiksna po obliku i lokaciji. Podešavanje fokusiranja se postiže unutrašnjim sočivom, fleksibilne strukture koja se može deformisati dejstvom cilijarnih mišića. Kako se ovi mišići skupljaju, sočivo se spljošti da bi dovelo objekt u oštar fokus.



Slika 6.21 Struktura elektronske kamere.



Slika 6.22 Struktura oka.

Kod oba sistema, iris se koristi da upravlja veličinom površine sočiva koja je izložena svetlosti, i stoga jačinom osvetljenosti slike projektovane na senzor slike. Iris kod oka je formirana od neprovidne mišićne materije koja može da se skupi kako bi zenica (pupila) postala veća i time propustila više svetlosti. Iris kod kamere je mehanički sklop koji obavlja istu funkciju.

Parametri u optičkim sistemima interaguju na mnogo neočekivanih načina. Na primer, razmotrimo kako količina raspoložive svetlosti i osetljivost senzora utiču na oštrinu snimljene slike. Ovu vezu razmatramo zato što se prečnik irisa i vreme ekspozicije podešavaju da prenesu odgovarajuću količinu svetlosti od scene na senzor slike. Ako je na raspolaganju više svetlosti nego što je potrebno, prečnik irisa može da se smanji, što će dati veću dubinu polja (zonu oštine; tj. raspon udaljenosti od kamere gde objekt ostaje u fokusu). Veća dubina polja obezbeđuje oštriju sliku kada su objekti na različitim udaljenostima. Osim toga, izobilje svetlosti omogućava da se smanji vreme ekspozicije, što daje manje zamućenje kada se snima kamerom iz ruke (usled podrhtavanja kamere ili tzv. polivanja kamerom) ili kada se objekt kreće.

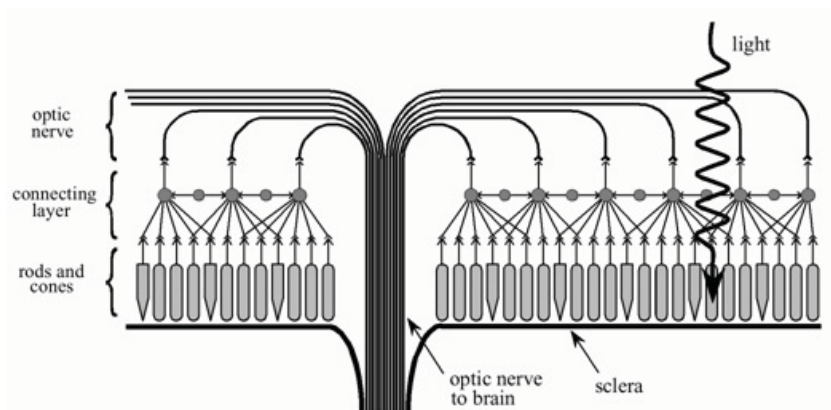
Podesivi iris je potreban i kod kamere i kod oka zato što je raspon svetlosnih intenziteta u okruženju mnogo veći nego što može da direktno podnese sam senzor. Na primer, odnos intenziteta sunčeve i mesečeve svetlosti je oko milion. Kada se još uzme u obzir da koeficijent refleksije može da varira između 1% i 99%, dobija se kao rezultat odnos intenziteta svetlosti od skoro sto miliona.

Dinamički opseg elektronske kamere je obično 300 do 1000, a definisan je kao najveći izmereni signal podeljen sa unutrašnjim šumom uređaja. Razmotrimo to na drugi način, neka je maksimalni proizvedeni signal 1 volt, a efektivni šum u tami je oko 1 milivolt. Sočiva obično imaju iris koji menja otvor za propuštanje svetlosti sa faktorom 300. Usled toga tipične elektronske kamere imaju dinamički opseg od nekoliko stotina hiljada. Naravno, ista kamera i sočivo, koji se koriste za snimanje pri jakoj sunčevoj svetlosti, biće neupotrebljivi za noćna snimanja.

Oko funkcioniše u dinamičkom opsegu, koji skoro da obuhvata celokupne varijacije svetlosti u okruženju. Međutim, iznenađuje činjenica da iris nije glavno sredstvo kojim se ovaj dinamički opseg postiže. Samo veličina zenice se menja sa faktorom 20 pri prelasku iz tame na jaku svetlost, i obrnuto. Svetlost koju detektuju nervne ćelije postepeno podešava njihovu osetljivost da prihvate preostali dinamički opseg. Na primer, oku je potrebno nekoliko minuta da se prilagodi slabom osvetljenju kada zakoračimo u mračnu prostoriju (bioskopsku salu).

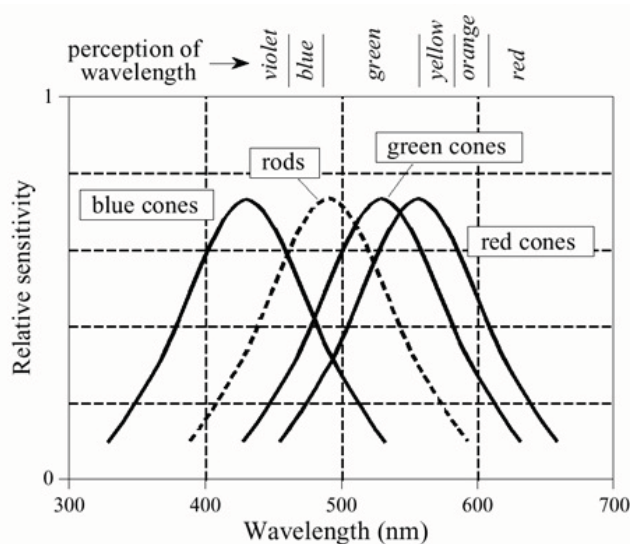
Jedan od načina na koji obrada digitalnih signala može da poboljša slike je da redukuje dinamički opseg koji je potrebno posmatrač da vidi, tj. mi ne želimo veoma svetle i veoma tamne oblasti u istoj slici. Reflektovana slika se obrazuje od dva signala slike: dvodimenzionalnog uzorka koji opisuje kako je scena osvetljena (tzv. iluminaciona komponenta) pomnoženog sa dvodimenzionalnim uzorkom koji opisuje refleksiju scene. Uzorak refleksije ima dinamički opseg manji od 100, zato što svi prirodni materijali reflektuju između 1% i 99% upadne svetlosti. Tu je sadržana glavna informacija slike, kao što su raspored objekata u sceni i karakteristike njihovih površina. Iluminacioni signal zavisi od izvora svetlosti, a ne od samih objekata. On može da ima dinamički opseg vrednosti nekoliko miliona, mada su vrednosti od 10 do 100 tipične unutar jedne slike. Ovaj signal nosi manje interesantnu informaciju, ali može da degradira finalnu sliku povećavanjem njenog dinamičkog opsega. Obrada digitalnih signala može da pomogne u ovoj situaciji potiskivanjem iluminacionog signala, čime omogućava signalu refleksije da dominira slikom.

Površina osetljiva na svetlost koja pokriva dno oka naziva se mrežnjača. Kao što se vidi na slici 6.23, mrežnjača može da se podeli na tri glavna sloja specijalizovanih nervnih ćelija: jedan za pretvaranje svetlosti u nervne signale, jedan za obradu slike i jedan za prenošenje informacija optičkom nervu koji se nalazi u mozgu. Skoro kod svih životinja ovi slojevi su na prvi pogled izokrenuti, tj. ćelije osetljive na svetlost su u poslednjem sloju, usled čega svetlost mora da prođe kroz druge slojeve da bi bila detektovana.



Slika 6.23 Struktura čovekove mrežnjače

Postoje dve vrste ćelija koje detektuju svetlost: štapići i čepići, a nazvane su prema fizičkom obliku koji imaju kada se posmatraju pod mikroskopom. Štapići su specijalizovani za funkcionisanje sa vrlo malo svetlosti, tj. za noćnu viziju. Vizija (vid) je vrlo zašumljena u tami, tj. slika izgleda kao da je popunjena kontinualno promenljivom zrnastom mustrom. Ovo je posledica veoma slabog signala slike, a ne ograničenja oka. U takvim uslovima postoji izuzetno malo svetlosti koja pada na oko, tako da može da se vidi slučajna detekcija pojedinačnih fotona. Ovo se naziva statističkim šumom i pojavljuje se kod obrade slika niskog svetlosnog intenziteta, što je slučaj kod vojnih sistema noćne vizije. Pošto štapići ne mogu da detektuju boju, vizija niskog intenziteta svetlosti je crno-bela.



Slika 6.24 Spektralni odziv oka

Receptori čepići su specijalizovani za razlikovanje boja, ali mogu da funkcionišu samo kada postoji solidna količina svetlosti. Postoje tri vrste čepića u oku: osetljivi na crveno (engl. *Red*), osetljivi na zeleno (engl. *Green*) i osetljivi na plavo (engl. *Blue*). Ova osetljivost na različite boje se javlja usled toga što sadrže različite fotopigmente, tj. hemikalije koje apsorbuju različite talasne dužine (boje) svetlosti. Slika 6.24 prikazuje talasne dužine svetlosti koje okidaju svaku od ove

tri vrste receptora. Ovo se naziva RGB kodiranje i pokazuje kako informacije o boji napuštaju oko kroz optički nerv. Čovekova percepcija boje se usložnjava nervnim procesiranjem u nižim nivoima mozga. RGB kodiranje se pretvara u drugu šemu kodiranja, gde su boje klasifikovane kao: crvena ili zelena, plava ili žuta, i svetlo ili tamno.

RGB kodiranje je važno ograničenje čovekovog vida; talasne dužine koje postoje u okruženju su grupisane u samo tri široke kategorije. Za razliku od oka, specijalizovane kamere mogu da razdvoje optički spektar u stotine ili hiljade pojedinih boja. Na primer, ovo se koristi za klasifikovanje ćelija kao kancerogenih ili zdravih, za tumačenje fizike udaljenih zvezda i slično. Zašto je oko tako ograničeno u detektovanju boja? Očigledno, usled čovekove potrebe za preživljavanjem, zahtevi postavljeni sistemu vizije mogu se grubo svesti na potrebu da pronađe crvenu jabuku između zelenih grana, naspram plavog neba.

Štapići i čepići su oko 3 mikrona široki i gusto su raspoređeni na celokupnoj površini mrežnjače, koja je dimenzija 3 cm x 3 cm. Uz malo matematike dobija se da je mrežnjača sastavljena od oko 100 miliona receptora. Optički nerv ima samo oko milion nervnih vlakana koji su povezani sa ovim ćelijama. U proseku, svako optičko vlakno je preko sloja povezivanja povezano sa 100 svetlosnih receptora. Pored konsolidovanja informacije, sloj povezivanja poboljšava sliku izostravanjem ivica i prigušivanjem iluminacione komponente scene.

U samom centru mrežnjače je mala oblast koja se zove fovea (latinski naziv za jamicu), koja se koristi za viziju visoke rezolucije. Fovea se razlikuje od ostatka mrežnjače u mnogo čemu. Prvo, optički nerv i slojevi za međupovezivanje se nalaze bočno od fovee, što omogućava receptorima da budu direktnije izloženi dolazećoj svetlosti. Usled toga fovea se pojavljuje kao malo udubljenje u mrežnjači. Drugo, samo čepići se nalaze u fovei i oni su gušće raspoređeni od onih koji se nalaze u ostatku mrežnjače. Odsustvo štapića u fovei objašnjava zašto je noćna vizija često bolja kada se objekt gleda bočno nego frontalno. Treće, svako optičko nervno vlakno je pod uticajem samo nekoliko čunjića, što obezbeđuje dobru sposobnost lokalizacije. Fovea je iznenađujuće mala. Pri čitanju teksta na normalnom rastojanju, fovea je oblast prečnika 1 mm, tj. manja je nego što je veličina bilo kog pojedinog slova. Rezolucija je ekvivalentna rešetki od 20 x 20 piksela unutar ove oblasti.

Čovekova vizija kompenzuje malu veličinu fovee naglim pomeranjima oka, koja omogućavaju visokoj rezoluciji fovee da brzo skenira vidno polje i stvori prikladnu informaciju. Pored toga, usled ovih naglih pomeranja oka štapići i čepići su izloženi kontinualno promenljivim uzorcima svetlosti. Ovo je važno zbog prirodne sposobnosti mrežnjače da se prilagodi promenljivim nivoima intenziteta svetlosti. U stvari, ako se oko primora da ostane fiksno na istoj sceni, detalji i boja počinju da se postepeno osvetljavaju u toku nekoliko sekundi.

6.4.4 Primarne boje

Crvena, zelena i plava boja nazivaju se osnovnim ili primarnim bojama jer se one ne mogu dobiti mešanjem drugih boja. Te se boje nazivaju i aditivni primari, jer sabiranje svetlosnih snopova tih boja daje belu i sve ostale boje vidljivog spektra. Takvim se sabiranjem različitih boja svetla, u različitim odnosima, mogu dobiti sve ostale boje, u svim nijansama i svetlinama. Takav se način nastanka boja naziva aditivnim. Tim načinom nastaju boje na ekranu televizora ili monitora. Međusobno mešanje snopova svetala primarnih boja dovodi do pojave nekih novih boja svetla.

Crveni snop pomešan sa zelenim snopom postaje žut. Crveni snop pomešan s plavim snopom postaje magenta (grimizan). Zeleni snop pomešan s plavim snopom postaje cijan (plavozeleni). Boje svetla (žuta, magenta i cijan) nastale su kao rezultat mešanja aditivnih primara, a nazivaju se suptraktivnim primarima. Tim načinom nastaju boje u offset štampi, štampačima sa mlaznicom i drugim uređajima koji koriste suptraktivno mešanje boja. Sledeće slike prikazuju aditivnu i suptraktivnu sintezu boja.



Slika 6.25 Aditivna sinteza boja



Slika 6.26 Suptraktivna sinteza boja

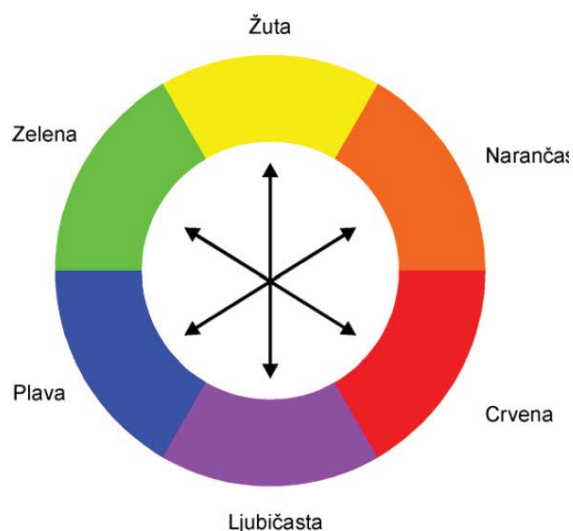
6.4.5 Komplementarne boje

Komplementarna boja je ona koja nastaje kad iz vidljivog spektra elektromagnetskog zračenja oduzmemo jednu od primarnih boja: belo - zeleno = magenta, belo - crveno = cijan, te belo - plavo = žuto. Komplementarna boja zelenoj je magenta, komplementarna crvenoj je cijan, a komplementarna plavoj je žuta. Dve komplementarne boje svetla, dodate jedna drugoj u određenim odnosima, izazivaju oset belog, odnosno sivog. Biti komplementaran znači biti nadopunjujući, dopunski, upotpunjujući. Komplementarne su one boje koje se međusobno nadopunjuju. Ne zna se tačno kome je najpre pala na um zamisao da sliku spektra savije u kružnicu i spoji joj krajeve te na taj način stvori kružnu paletu boja. Slika 6.27. prikazuje traku boja koja savijena u kružnicu obrazuje kružnu paletu boja. Savremena kružna paleta boja, kakvom su se služili i služe svi teoretičari boje od Johana Wolfganga Goethea do Johannesena Ittena, podeljena je na šest osnovnih boja: žutu, narandžastu, crvenu, ljubičastu, plavu i zelenu. Slika 6.28 prikazuje savremenu kružnu paletu sa 6 boja. Komplementarne su one boje koje se u kružnoj paleti boja nalaze jedna nasuprot drugoj. Tri su osnovna komplementarna para: žuta/

ljubičasta, narandžasta/plava i crvena/zelena. Slika 6.29 prikazuje 3 osnovna komplementarna para. Slika 6.30 prikazuje kružnu paletu sa 12 boja koja se koristi uz kružnu paletu sa 6 boja. Na kružnim paletama sa 6 ili 12 boja podela boja je gruba. Ovakva podela nije tačna nego približna, tako da je žutoj komplementarna ljubičasta, narandžastoj plava, a crvenoj zelena. Kada bi se na kružnu paletu postavilo puno više boja, komplementarne boje bi bile malo drugačije. Postojao bi određeni pomak u odnosu na ovde navedene boje.



Slika 6.27 Traka boja



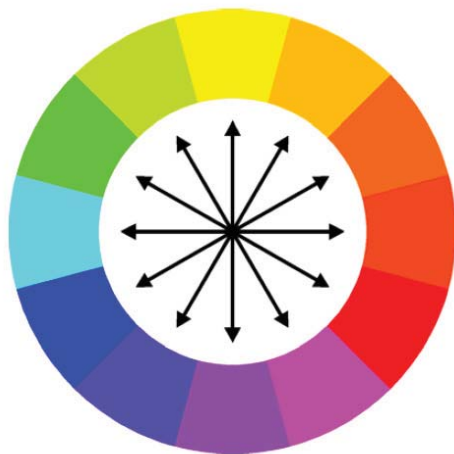
Slika 6.28 Kružna paleta boja (6 boja)



Slika 6.29 Tri osnovna komplementarna para

Godine 1801. Thomas Young, a posle Herman von Helmholtz, ustanovili su kako je moguće prikazati bilo koju boju svetla pomoću mešavine triju primarnih svetlosnih izvora. Postoje različita pomagala za bolje razumevanje i lakše snalaženje u sistemu pravila harmoničnog slaganja boja. Tu je u prvom redu kružna paleta boja kojom se služio još Johann Wolfgang von Goethe. Taj prikaz boja se oslanja više na mešanje pigmenata i bojila. Nedostatak mu je što se njime mogu prikazati samo spektralne boje. Da bi se to ispravilo, postoji još i devetodelna, trouglasta paleta boja, na kojoj se mogu prikazati i tercijari – oker, maslinastozelena i crvenkastosmeđa. Nema boja koje su važnije ili manje važne – postoje samo one koje su uočljivije ili koje prema svom delovanju na čovekova osetila čovek svrstava među one koje lakše zapaža ili čak jednostavnije imenuje. Prema redosledu tvorbe boje se dele na:

- primare, ili boje prvog reda,
- sekundare, ili boje drugog reda,
- tercijske, ili boje trećeg reda.



Slika 6.30 Kružna paleta boja (12 boja)

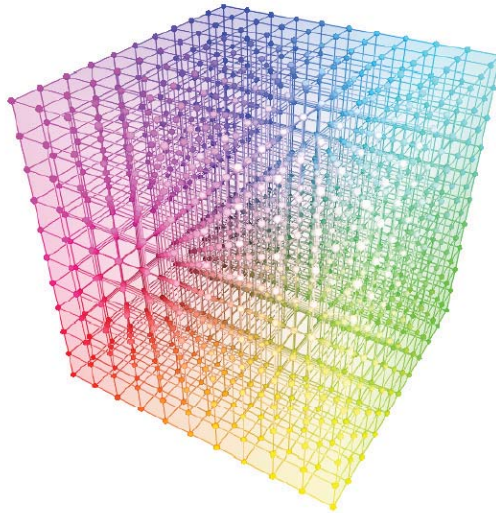
6.5 Prostor boja

Prostor boja ili model boja je način pomoću kog se definišu, stvaraju i vizuelizuju boje. Boja je najčešće definisana pomoću 3 koordinate (parametra). Ti parametri određuju poziciju boje unutar prostora boja koji se koristi. Postoji mnogo prostora boja i svaki se koristi za različite namene. Neki prostori boja su linearni, što znači da će određeni iznos promene podražati izazvati isti iznos promene u percepciji te boje. Mnogi prostori boja nisu linearni. Neki prostori boja su intuitivni za korišćenje, odnosno lako je pomoću parametra u njima izabrati boju. Drugi prostori boja nisu intuitivni, odnosno pomoću parametra je vrlo teško ili gotovo nemoguće u njima izabrati boju. Neki prostori boja vezani su uz uređaj na kom su definisane boje i za njih se kaže da su zavisni od uređaja (engl. *device dependent*). Kada se boje definisane u tim prostorima koriste na nekom drugom uređaju, više nisu iste kao na uređaju gde su bile definisane.

6.5.1 RGB prostor boja

RGB prostor boja definisan je pomoću tri aditivna primara: crvene, zelene i plave boje. Svaka boja u tom prostoru boja nastaje sabiranjem pojedinih komponenata te tri boje. RGB model predstavljen je pomoću kocke, gde crvena boja predstavlja x-osu, zelena boja y-osu, a plava boja z-osu. Slika 6.31 prikazuje model RGB prostora boja. Taj se prostor boja najčešće koristi u računarima. Svaka boja najčešće je predstavljena sa 8-bitima, odnosno vrednostima od 0 do 255 (256 vrednosti). To daje ukupno $256^3 = 16777216$ mogućih boja. U različitim računarskim programima za obradu slike najčešće se navodi podatak o 16 miliona boja ili engleski naziv *true color*. Najčešće se taj prostor boja normira (normalizuje) na vrednosti od 0 do 1. RGB prostor boja jednostavan je za računar, ali nije prikladan za čoveka. Crvena, zelena i plava komponenta međusobno su korelisane tako da je čoveku vrlo teško izborom tih komponenata definisati željenu boju u RGB prostoru boja. Stoga se najčešće koriste neki drugi prostori boja kao što su

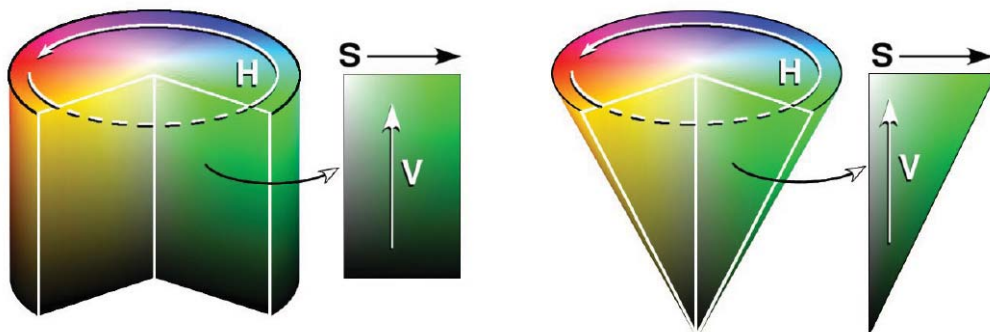
npr. HSV (HSI, HSB) ili HSL prostor boja.



Slika 6.31 Model RGB prostora boja

6.5.2 HSV prostor boja

HSV prostor boja stvorio je A. R. Smith 1978. godine. Taj prostor boja definisan je s tri koordinate: tonom boje (engl. *hue*), zasićenjem boje (engl. *saturation*) i svetlinom boje (engl. *value*, *intensity*, *brightness*). Ton boje predstavljen je uglom od 0° do 360° . Zasićenost boje ima vrednost od 0% do 100%. Svetlina boje ima vrednost od 0% do 100%. HSV prostor boja predstavljen je pomoću valjaka. Često se taj prostor boja prikazuje kao konus ili šestostrana piramida, jer je percipirana promena zasićenja boje od 0% do 100% manja za tamne boje (one koje imaju manju vrednost svetline) nego za svetle boje (one koje imaju veću vrednost svetline). Da bi se nadoknadila ta razlika u percepciji, valjak se izobličuje u konus. Slika 6.32 prikazuje model HSV prostora boja kao valjak i kao konus. Pokazalo se da je u ovom prostoru boja čoveku daleko lakše (intuitivnije) da definiše i izabere boju nego u RGB prostoru boja. Često se vrednosti tona, zasićenja i svetline boje normiraju na vrednosti od 0 do 1.



Slika 6.32 Model HSV prostora boja predstavljen pomoću valjka i konusa

6.6 Slike i stvaranje digitalnih slika

Vizuelna informacija nosi oko 90% ukupnog informacionog sadržaja koje čovek prihvati iz okruženja preko svojih senzorskih sistema. Ovo odslikava činjenicu da se čovek u velikoj meri oslanja na svoj visoko razvijeni vizuelni sistem u poređenju sa drugim senzornim putevima. Spoljašnji optički signal opažaju oči a zatim se pretvara u nervni signal; odgovarajući nervni podsystem specijalizovan za sistem vizije je specijalno organizovan da detektuje suptilne karakteristike slike i obavi procesiranje visokog nivoa, koje se dalje procesira da generiše entitete objekta i koncepte.

Anatomija sistema vizije objašnjava sa aspekta strukture zašto je toliko važna vizuelna informacija za čovekovu kogniciju. Funkcije kognicije koje takav sistem mora da podrži obuhvataju sposobnost da razlikuje objekte, njihov položaj u prostoru, kretanje, veličine, oblike i teksturne površine. Neke od ovih primitiva mogu da se koriste kao deskriptori sadržaja slike.

6.6.1 Taksonomija slika

Postoje dva formata u kojima vizuelne informacije mogu biti zabeležene i prezentirane - statična slika i pokretna slika, odnosno video. Statična slika je ono što nas u ovom radu interesuje. Mada relativno nova oblast istraživanja, kompjuterizovana obrada digitalnih slika privukla je prilično pažnje i bacila svetlo na širok spektar postojećih i potencijalnih aplikacija. To je direktno izazvano naglom akumulacijom slikovnih podataka i posledica eksponencijalnog rasta kapaciteta digitalnih memorijskih uređaja i procesorske snage. Postoji nekoliko glavnih tipova digitalnih slika zavisno od elementarnih konstituenata koji prenose sadržaj slike. Slike mogu biti sledećeg oblika:

- Štampani tekst i rukopis. Primeri ovog tipa su mikro-filmovi starih dokumenata, rukopisa ili fotografija.
- Linijski crteži, uključujući dijagrame, skice ...
- Polutonovi (engl. halftone). Slike su predstavljene mrežom tačaka promenljive veličine i oblika.
- Kontinualni ton. Fotografske slike koje koriste glatke i suptilne tonove.
- Kombinacija svega navedenog.

Fotografske slike i slike kontinualnog tona su najuobičajenije u praksi stvaranja digitalnih slika i od glavnog interesa u pronalaženju slika na osnovu sadržaja. Skeneri i digitalne kamere su danas široko zastupljeni uređaji tako da je lako konvertovati analogne elektronske formate u digitalni format.

Na konceptualnom nivou, slika je reprezentacija svog ciljnog objekta (objekata). Prema trećem izdanju *Webster's New International Dictionary*, slika je „*optički pandan objekta ... mentalna slika, mentalna koncepcija ...*”. Definicija odslikava činjenicu da slika ima svoj sadržaj, koji apsorbuje optička ili mentalna svojstva objekta; sa svojim formatom koji varira kroz različite vrste medijuma. To je određeno time kako su optička svojstva kvantizovana ili stepenom mentalnih apstrakcija koji je zahtevan. Primeri slika prema prethodno navedenoj definiciji:

1. Slika može biti niz pikselskih vrednosti smeštenih u nekompresovani bitmapirani digitalni format slike. Kod ovog formata, svaka vrednost predstavlja intenzitet boje u diskretnim tačkama

ili pikselima. Dobro poznat primer je Microsoftov BMP format. Mada BMP format dozvoljava pakovanje piksela i run-length kodiranje da bi se ostvario određen nivo kompresije, njegova nekompresovana verzija je popularnija.

2. Popularni i standardni formati slika za Internet, GIF i JPEG, zahtevaju ekstenzivnu transformaciju i kompresiju slike. GIF standard definiše proces degeneracije boje, koji mapira boje slike u ne više od 256 novih boja. Zatim se primenjuje kompresovanje korišćenjem LZW (*Lempel Zip Walch*) algoritma. GIF slike smeštaju paletu boje zajedno sa kompresovanim pikselnim vrednostima. Nasuprot tome, JPEG primenjuje transformaciju na intenzitet piksela a ne preslikavanje. Diskretna kosinusna transformacija (DCT) se koristi da transformiše reprezentaciju pikselne vrednosti u prostornom domenu u reprezentaciju u frekventnom domenu. Umesto pikselnih vrednosti, koeficijenti kosinusne transformacije se koriste za reprezentovanje talasa signala. Dvo-dimenzionalna slika se reprezentuje kao kombinacija talasa signala različitih frekvencija. DCT je ortogonalna transformacija koja je kompjuterski reverzibilna. Sposobnost pakovanja informacija od DCT se upotpunjuje potonjim korakom kvantizacije, koji koristi pogodnost karakteristika čovekove vizije da filtrira delove signala koji nisu značajni za čovekov vizuelni sistem. Kodiranje entropije ili aritmetičko kodiranje se primenjuje da se ostvari viši stepen kompresije pre nego što se koeficijenti memorišu [100].

3. Specijalno definisan fajl „potpisa” (engl. signature) koji čuva specijalno izdvojena obeležja slike u numeričkom ili Bool formatu, ukazujući na prisustvo (ili odsustvo) i intenzitet svojstava. To je veoma kompaktna reprezentacija slike koja je namenjena brzom pronalaženju a ne prikazivanju, arhiviranju itd. Samo se suštinska obeležja slike konzerviraju u fajlu potpisa i ona su zavisna od algoritma. Ovo omogućava lako indeksiranje i brzo pretraživanje svojstava. Primer za ovo je metoda kodiranja slike koja koristi vektor kvantizacije (engl. *vector quantization* - VQ), kod koje se blokovi slike kodiraju prema pažljivo izabranoj tabeli kodova. Ako su blokovi slike slični jedni drugima ili slike u skupu nose značajnu sličnost, može se obično postići viši stepen kompresije nego u slučaju opšte-namenskih algoritama kompresije koji se koriste kod GIF i JPEG.

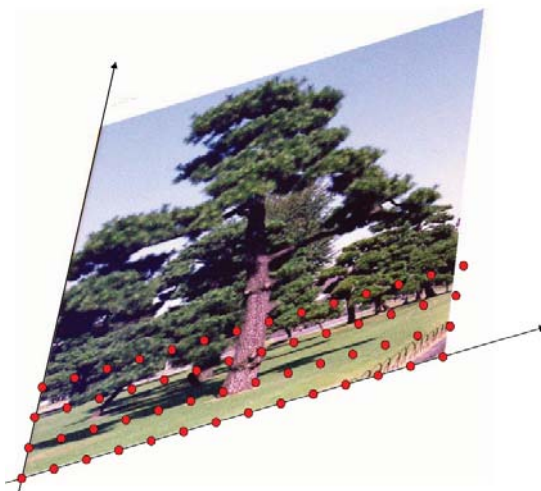
4. Tekstualna anotacija može se posmatrati i kao instanca mentalne slike, a ponekad kao deskriptori koji mogu da se kodiraju prema unapred definisanoj konvenciji ili tezaursu. činjenica da dve vizuelno različite slike mogu da prenesu isti koncept a različiti koncepti mogu postojati u slikama koje dele više sličnih optimalnih svojstava dovodi do procesa između slika pronađenih po sadržaju i pronalaženju po konceptu [101], [102].

6.6.2 Reprezentovanje slike

Sa matematičke tačke gledišta, slike su opisi kako neki parametar varira na površini. Na primer, standardne vizuelne slike su rezultat varijacija intenziteta svetlosti u dvodimenzionalnoj ravni. Međutim, svetlost nije jedini parametar koji se koristi u tehničkoj obradi slike. Na primer, slika može biti formirana od temperature integrisanog kola, brzine krvotoka i slično. Ove egzotične slike se obično pretvaraju u konvencionalne (svetlosne) slike, tako da ih čovekovo oko može procenjivati.

Za razliku od zvuka, koji ima svoje informacije kodirane u vremenskom domenu, slike imaju svoje informacije kodirane u prostornom domenu. Drugim rečima, svojstva slika se predstavljaju ivicama, a ne sinusoidama. Ovo znači da veličina svojstva koje želimo da vidimo na slici određuje razmak i broj piksela, a ne formalno ograničenje tzv. teoreme odabiranja koju ovde nećemo razmatrati.

Slika se u digitalnim sistemima predstavlja matricom tačaka – piksela. Parametri predstavljanja su rezolucija i dinamički raspon. *Rezolucija* je mera preciznosti predstavljanja i razlikujemo *relativnu rezoluciju* koja je broj piksela po jedinici dužine (obično po inču) i *apsolutnu rezoluciju* koja je veličina matrice mereno brojem piksela. Dinamički raspon određuje preciznost predstavljanja pojedinačnih piksela i izražava se brojem različitih podržanih nijansi svake hromatske komponente svetlosti. Dinamički raspon monohromatskog piksela meri se brojem nijansi sive a dinamički raspon piksela u boji meri se brojem nijansi svake od komponenti



Slika 6.33 U digitalnim sistemima slika se predstavlja matricom tačaka - piksela

Ljudsko oko je u stanju da raspozna oko 350000 boja i nešto je osetljivije prema nijansama zelene boje.

Uobičajeni modeli pri RGB prikazivanju su:

12 bita (4096 nijansi) – po 4 bita (16 nijansi) za svaku osnovnu komponentu

15 bita (32768) – po 5 bita (32)

16 bita (65536) – po 5 bita (32) za crvenu i plavu i 6 bita (64) za zelenu

24 bita (16777216) – po 8 bita (256)

Uobičajeni modeli pri RGB obradi su:

30 bita – po 10 bita (1024)

36 bita – po 12 bita (4096)

48 bita – po 16 bita (65536)

Smisao ovih formata je u očuvanju kvaliteta pri obradi slika. Normalno ljudsko oko ne može razlikovati ove zapise od 24-bitnog.

Zapis slike se obično sastoji od:

- zaglavlja – podataka koji opisuju
 - širinu,
 - visinu,

- dinamički raspon i
 - detalje zapisa,
 - sadržaja slike
- sadržaja slike

Veličina zapisa slike bez kompresije je $S \cdot V \cdot B / 8$ bajtova, gde je:

S – širina slike u pikselima

V – visina slike u pikselima

B – broj bitova kojima se opisuje svaki piksel

Pored toga, potreban je i određen prostor za zaglavlje.

Na primer:

$$1024 \times 768 \times 16 / 8 = 1.5 \text{ MB}$$

$$1600 \times 1200 \times 24 / 8 = 5.5 \text{ MB}$$

Tipična digitalna slika je sačinjena od oko 500 redova i 500 kolona. Ovo je kvalitet slike kod televizije i grafičkih aplikacija na personalnim računarima. Slike niže rezolucije, 250 x 250 piksela, smatraju se slikama loše rezolucije. Sa druge strane, slike sa više od 1000 x 1000 piksela smatraju se izuzetno dobrim. To je kvalitet najbolje računarske grafike, televizije visoke definicije i 35-milimetarskog filma.

Najjači motiv za korišćenje slika niže rezolucije je što imaju manje piksela koje treba preneti. Ovo nije trivijalno; jedan od najvećih problema kod obrade slike je upravljanje ogromnim količinama podataka. Na primer, jedna sekunda digitalnog audio zapisa je oko 8 kilobajta podataka, a jedna sekunda televizijskog programa je oko 8 megabajta podataka. Za prenos slike od 500 x 500 piksela, modemom brzine 33,6 kb/s, potreban je skoro minut!

Kod obrade slike uobičajeno je da se koristi skala sa 256 sivih nijansi (kvantizacioni nivoi), što odgovara jednom bajtu po pikselu. Ovaj izbor je napravljen iz više razloga. Prvo, jedan bajt je osnovna memorijska jedinica memorijskog sklopa računara i kao takav pogodan je kod upravljanja podacima. Drugo, veliki broj piksela u slici kompenzuje u izvesnoj meri ograničen broj kvantizacionih nivoa, tj. čovekovo oko susedne piksele koji imaju približne digitalne vrednosti osvetljaja opaža kao oblast sa usrednjenom vrednošću osvetljaja. Drugim rečima, slike su veoma „umekšane”. Treće, i najvažnije, veličina promene osvetljaja od 1/256 (0,39%) je manja od onoga što oko može da zapazi, tako da nema svrhe koristiti više kvantizacionih nivoa. Međutim, neke slike je potrebno memorisati sa više od 8 bitova po pikselu, naročito kada su bitni detalji.

Vrednost svakog piksela digitalne slike predstavlja malu oblast u kontinualnoj (analognoj) slici koja se digitalizuje. Ova oblast kontinualne slike koja učestvuje u vrednosti piksela naziva se blenda odabiranja (engl. *sampling aperture*). Veličina blende odabiranja je često usko povezana sa unutrašnjim svojstvima konkretnog sistema za obradu slike.

Boja se dodaje digitalnim slikama korišćenjem tri broja za svaki piksel, koji predstavljaju intenzitete tri primarne boje: crvene, zelene i plave. Mešanjem ove tri boje stvaraju se sve moguće boje koje čovekovo oko može da opazi. Često se jedan bajt koristi za memorisanje svakog od ovih intenziteta boja, što omogućava da slika „uhvati” ukupno $256 \times 256 \times 256 = 16,8$ miliona različitih boja.

Boja je veoma važna kada je cilj da se gledaocu predstavi što vernija slika realnog sveta, što je slučaj kod televizije i fotografije.

Dizajniranje različitih formata fajlova slika obično je uslovljeno fizičkom implementacijom hardvera koji podržava proces obrade slike, namenom sadržaja slike i kako slika treba da se obrađuje. Na primer, rasterska reprezentacija slike je tesno povezana sa odmeravanjem (uzorkovanjem; engl. *sampling*) digitalne slike i aparaturom prikazivanja. Reprezentacija wavelet koeficijentima se koristi u nekim sistemima koji zahtevaju kontinualni prenos kroz računarsku mrežu uz mogućnost zumiranja.

Jednostavni formati slike takvi kao što je Microsoftov kompresovani Windows BMP format sastoji se od dva vrlo jednostavna dela, zaglavlja (engl. *header*) koje sadrži meta informacije o slici i drugog dela koji čuva sekvencijalno intenzitet svakog piksela. U zaglavlju se nalaze sledeće informacije: veličina zaglavlja, širina slike, visina slike, broj bitova po pikselu, preporučena rezolucija (u pikselima po metru), broj značajnih boja i neka rezervisana polja. Kod nekompresovanog BMP fajla sa 24 bita po pikselu, neposredno posle zaglavlja, smeštene su pikselske vrednosti u redosledu sken linija počevši odozdo na gore sa poslednjom linijom na vrhu. U svakoj sken liniji, pikselske vrednosti su uređene sa leva na desno, sa 3 bajta za svaki piksel koji definišu vrednosti za Red, Green i Blue, tako da svaki kanal boje zahvata 8 bita koji mapiraju raspon od 0 do 255. Zbog efikasnijeg pristupa memoriji, broj bajtova po sken liniji zaokružen je na umnožak od četiri. Ostatak bajtova popunjava se proizvoljnim vrednostima. Za kompresovane BMP fajlove ili kada broj bitova po pikselu nije jednak 24 (broj bitova po pikselu može biti 1, 4, 8 ili 24), potrebni su dodatni koraci da bi se generisao BMP fajl od rasterske reprezentacije.

Obzirom da fajlovi slike zahtevaju enormni prostor za skladištenje i znatan propusni opseg za čist format pikselskih vrednosti, kompresija je poželjna u većini situacija. Algoritmi kompresije slike obično mogu da redukuju veličinu fajla do 10 puta bez znatnog uticaja na vizuelni kvalitet. Sa druge strane, kompresija zahteva opsežno procesiranje i veliku procesorsku snagu. Najčešće korišćene rutine za kompresiju obično uključuju neku vrstu transformacije, kao što je diskretna kosinusna transformacija i diskretna wavelet transformacija (DWT), koju prati opšti algoritam kompresije kao što je kodiranje dužinom serije (engl. *run-length encoding - RLE*) i kodiranje entropije (Huffmanovo kodiranje). Na početku rutine kompresije radi se opciono transformisanje/kvantizacija prostora boje, što se radi kod GIF formata i JPEG standarda.

Pikselske vrednosti moraju da se podvrgnu sofisticiranom procesiranju pre nego što se smeste u JFIF format, koji je više poznat kao JPEG fajl. Obuhvaćena su četiri glavna koraka: transformacija prostora boja, diskretna kosinusna transformacija (DCT), kvantizacija i kompresija entropije. Prvo, pikselske vrednosti treba da se konvertuju iz RGB modela boja u YCbCr model boja, koji ima pogodnost da koncentriše informaciju u Y kanal tako da omogućava bolje kompresovanje za Cb i Cr kanale. Zatim, slika se deli na 8x8 blokova, na koje se primenjuje dvo-dimenzionalna DCT za svaki od tri kanala. DCT je vrsta ortogonalne transformacije koja može da se implemenira kao matrično množenje standardnih 2-dimenzionalnih DCT matrica. Rezultujući koeficijenti se kvantizuju prema svom uticaju na vizuelni kvalitet, sa konzerviranjem više informacija iz nisko-frekventnih komponenti a manje iz visoko-frekventnih prema empirijskom iskustvu. Ovo se postiže preko deljenja koeficijenata matrice sa dve tabele kvantizacije, za Y komponente i Cb-Cr komponente. Huffmanovo kodiranje se koristi za finalno kompresovanje. JPEG standard

omogućava više različitih implementacija a stepen kompresije može se podesiti.

Više hardverski orijentisan format slike je TIFF (*Tag Image File Format*) format. TIFFF služi kao omotač (engl. wrapper) koji drži jedan ili više okvira slike u jednom fajlu, svaki sa meta informacijama smeštenim sa tagovima. Slika u svakom okviru može biti u drugačijem formatu, ili kompresovana ili nekompresovana. TIFF rukuje okvirima slike pojedinačno. Obično se TIFF fajlom rukuje preko biblioteke funkcija koja obezbeđuje konzistentan interfejs za programiranje aplikacija (engl. Application Programming Interface - API) koje manipulišu slikom na različitom nivou granularnosti, kao što je piksel, sken linija, pruga i blok. Jedna takva biblioteka je LIBTIFF (<http://www.libtiff.org/>).

Format slike je važan faktor u pronalaženju slika, pošto on određuje primarne konstituente slike koji prenose sadržaj vizuelne informacije. To mogu biti pikseli, koeficijenti frekvencije, histogrami boja, konceptualni termini, što je slučaj u prethodno navedene 4 vrste slika. Tradicionalno, sve obrade slike bave se prvom vrstom formata slike, dok preostala tri mogu da se izvedu iz njega. Međutim, manipulisanje kompresovanim slikama je od velike važnosti u nekim aplikacijama, kao što je prenos videa visokog kvaliteta: fajl signature je uobičajena stvar u većini sistema pronalaženja; a najviše korišćeni i pouzdani sistemi za pronalaženje slika još koriste tekstualnu anotaciju (koju su eksperti ručno odradili) kao medijator za pronalaženje. Takođe, formira se hijerarhija apstrakcija slike od fizičko optičkih entiteta. Vrste interesantnih obeležja i odgovarajući algoritmi ili metodologija za ekstrakciju najopisnijih obeležja određuju detalje apstrakcije.

6.7 Obeležja slike

Neki istraživači su ustanovili da su neke vrste karakteristika korisne za opisivanje sadržaja slike. One su obično dobro definisane sa skupom rutina koje podržavaju obradu slike. Sa druge strane, one lako mogu da se mapiraju u određene aspekte čovekove vizuelne percepcije. Proces čovekove vizuelne percepcije i konstrukcije mentalne slike su izneti u Marrovoj teoriji prvobitne skice (engl. *primal sketch*). Čovekov vizuelni sistem može da razlikuje objekte, identifikuje njihovu lokaciju u prostoru, njihovo kretanje, veličine, oblike i njihove teksturne površine [103]. Shodno tome, sistemi mašinske vizije takođe izvode algoritme za prepoznavanje objekata, detekciju kretanja, detekciju ivica i konture, opis oblika i identifikaciju mustre (paterna) teksture. Za pronalaženje statičnih slika relevantni su oni koji se odnose na prepoznavanje objekata, ivica i kontura, oblika i tekstura a vrlo često se koriste i obeležja boje da pripomognu u svakom aspektu ovih karakteristika ili se koriste samostalno kao globalno obeležje cele slike.

Na osnovu prethodne analize može se definisati koncept obeležja slike. Obeležje slike može se definisati kao rezultat izračunavanja ili ekspertovog kriterijuma evaluacije, obavljenog na ciljnoj slici. Praktično, iz perspektive mašinske vizije i kompjuterizovanog pronalaženja slika na osnovu sadržaja, možemo definisati sadržaj slike kao skup svih mogućih obeležja ili kombinaciju osnovnih obeležja ciljne slike. Priroda sadržaja slike kojom treba da se bavimo, informacije koje su korisniku potrebne, kao i način na koji ljudi interpretiraju sadržaj slike variraju od slučaja do slučaja. Kao istraživački pristup razumevanju sadržaja slike pomoću mašinske inteligencije, nije neuobičajeno da se priibegne korišćenju obeležja slike manje intuitivnog za čovekovu percepciju, ili čak razviju novi algoritmi za definisanje novih obeležja slike u cilju razmatranja konkretnog

sadržaja slike i korisnikovih potreba za informacijom. Ove činjenice idu u prilog otvorenoj definiciji obeležja slike koja je proširiva da se prilagodi konkretnim slučajevima aplikacija. Nije zgoreg napomenuti da je obeležje slike tesno povezano sa algoritmom koji ga definiše.

Prema različitim interesovanjima, sadržaj slike može se opažati u tri nivoa sa rastućom kompleksnošću: mašinska vizija koja je dobra pri mudrom obuhvatanju primitivnih obeležja, ali nije ograničena na boju, teksturu i oblik [104]; ljudi kao konzumenti informacija slike su više zainteresovani za kompoziciju osnovnih obeležja za formiranje objekata i semantičkih relacija između više objekata u istom kontekstu slike. Na najvišem nivou, sadržaj slike može da se simbolizuje konceptima koje on prenosi [105], [106]. Shodno tome, postoje upiti koje korisnici mogu da generišu fokusirajući se na različite nivoe sadržaja. Neki od njih mogu se zadovoljiti fokusiranjem na samo jedno ili dva obeležja slike, dok drugi zahtevaju razumevanje semantike slike.

6.8 Presek stanja CBIR sistema

Uopšteno, pretraživanje slika i konkretno, pretraživanje slika na osnovu sadržaja su dobro poznata polja istraživanja u oblasti upravljanja informacijama u kojima je predlagan i istraživani veliki broj metoda ali još uvek nema zadovoljavajućih opštih metoda.

Kod CBIR, grubo govoreći, postoje dva glavna pristupa: diskretni i kontinualni [107]. Diskretni pristup je inspirisan pretraživanjem tekstualnih informacija i koriste se tehnike kao što su invertovani fajlovi i metrike pretraživanja teksta. Ovaj pristup zahteva da sva obeležja budu preslikana u binarna obeležja; prisustvo izvesnog obeležja slike se tretira kao prisustvo reči u dokumentu. Kontinualni pristup je sličan klasifikaciji najbližih suseda. Svaka slika se predstavlja vektorom obeležja i ta obeležja se porede koristeći različite metrike rastojanja. Slika sa najmanjim rastojanjem se najviše rangira u procesu pretraživanja.

Prednost diskretnog pristupa je što se metode iz pretraživanja tekstualnih informacija mogu lako transferisati kao korisnikova interakcija i rukovanje smeštanjem. Pored svega toga, većina sistema za pretraživanje slika sledi kontinualni pristup koristeći često neku optimizaciju, na primer prefiltriranje i pre-klasifikaciju, da ostvari bolje performanse izvršavanja.

Odnedavno, lokalna obeležja slike zadobila su veću pažnju unutar zajednice koja se bavi računarskom vizijom. Osnovna ideja je da se objekti u slikama sastoje od delova koji se mogu modelovati sa različitim stepenima nezavisnosti. Ti pristupi se uspešno koriste za prepoznavanje i detekciju objekata i CBIR. Za predstavljanje lokalnih delova slike uobičajeno se koriste SIFT obeležja i tzv. *raw image patches* i prihvata se *bag-of-features* pristup, sličan sa *bag-of-words* pristupom koji se koristi u procesiranju prirodnih jezika. Nasuprot prethodno opisanim metodama, slika se ne modeluje kao celina već se delovi slike modeluju pojedinačno. Većina pristupa opisana u literaturi o prepoznavanju objekata na osnovu delova uči (često složene) modele iz velikog skupa obučavajućih podataka. Taj pristup je nepraktičan za CBIR aplikacije zato što zahteva enormnu količinu obučavajućih podataka sa jedne strane i vodi do enormnih vremena za kreiranje tih modela, sa druge strane. Međutim, neki od ovih pristupa su primenjivi za pretraživanje u ograničenom domenu, kao što je slučaj kod IRMA baze podataka.

Grubo govoreći, obeležja mogu biti grupisana u sledeće tipove:

- a) predstavljanje boje,
- b) predstavljanje teksture,
- c) lokalna obeležja i
- d) predstavljanje oblika

Funkcija rastojanja koja se koristi za poređenje obeležja koja predstavljaju sliku takođe ima veliki uticaj na performanse sistema.

6.9 Histogrami digitalnih slika

Šta je histogram? U statistici, histogram je grafički prikaz tabeliranih frekvencija. Histogram je grafička verzija tabele koja pokazuje koja proporcija slučajeva pripada svakoj od više ili većem broju specificiranih kategorija. Kategorije se obično specificiraju kao nepreklapajući intervali neke promenljive. Kategorije (stupci; trake) moraju biti susedni.

Uopšteno, histogram je definisan kao način prikazivanja podataka raspoređenih u određene kategorije (klase) ili grupe. Prvi korak u kreiranju histograma je skupljanje podataka i razvrstavanje prikupljenih podataka u kategorije. Nadalje moramo odrediti koje su promenljive zavisne, a koje nezavisne. Karakteristika po kojoj smo grupisali podatke u kategorije predstavlja nezavisnu promenljivu, a broj prikupljenih podataka koji upadaju u određenu kategoriju predstavlja zavisnu promenljivu.

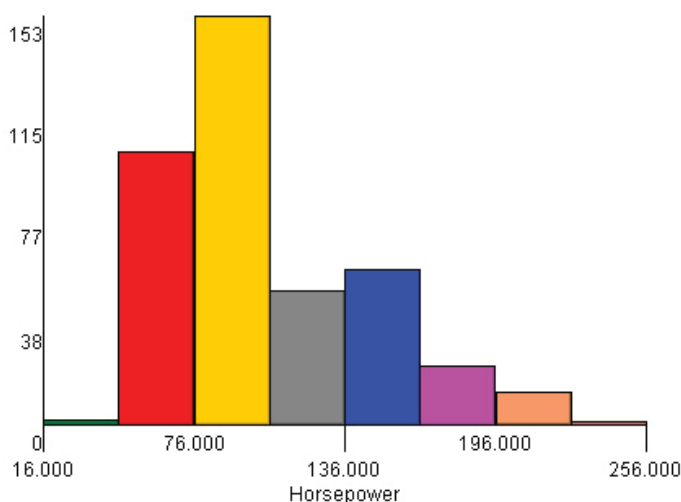
Histogram je zapravo stubičasti graf, koji na apscisnoj osi ima vrednosti nezavisne promenljive, a na ordinatnoj osi vrednosti zavisne promenljive. Oznake na osama treba da budu linearno raspoređene. Graf se crta tako da se prvo na apscisu nanese vrednosti svih kategorija, čime dobijamo apscisu podeljenu na intervale. Zatim se broj podataka koji odgovaraju toj kategoriji crta kao horizontalna linija iznad odgovarajućeg intervala. To je razlog zbog kojeg dobijamo stubičasti graf.

Da bi bilo jasnije ono što je prethodno rečeno, navodim jedan jednostavan primer. Uzmimo da posmatramo snage motora automobila. Zanima nas u kom rasponu se kreću snage automobila, te kako su te snage raspoređene. Broj uzoraka koje posmatramo je 400 automobila, a snaga je izražena u konjskim snagama. Kao rezultat ispitivanja dobijamo određenu količinu podataka koja je data u tabeli 1. Kao što vidimo iz tabele kategorija po kojoj razvrstavamo automobile je snaga motora u konjskim snagama. Imamo 8 kategorija i one su: 16-46 konjskih snaga, 46-76, 76-106 itd. U desnoj koloni dat je broj automobila čija snaga motora ulazi u zadati raspon. Histogram crtamo na bazi tabele 6.2 i on je prikazan na slici 6.34.

Histogram ima široku primenu, a posebno je pogodan za prikaz rezultata ispitivanja izvršenog na velikom broju uzoraka, tj. kada nije pogodno prikazivati vrednost svakog pojedinog ispitanog uzorka. Kao primer možemo navesti brojna statistička ispitivanja, koja se mogu odnositi na bilo šta.

Tabela 6.2 Raspodela automobila po kategorijama snage

Raspon snaga (KS)	Broj automobila
16-46	2
46-76	103
76-106	153
106-136	50
136-166	58
166-196	22
196-226	11
226-256	1



Slika 6.34 Histogram

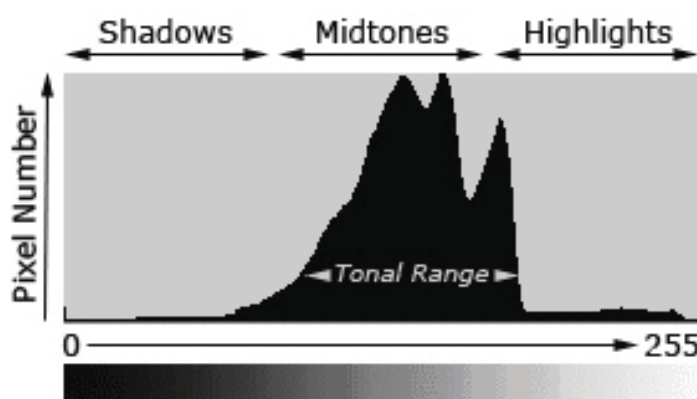
Jedna od vrlo važnih primena histograma je u digitalnoj fotografiji. To se ne odnosi isključivo na digitalne fotoaparate, nego npr. i na skenere. Međutim, budući da digitalne kamere danas poseduje veliki broj ljudi upotrebu histograma ćemo opisati na primeru snimanja digitalne fotografije. Kako možemo proceniti snimljenu fotografiju? Jednostavne faktore poput fokusa ili kadra možemo odmah videti gledajući snimljenu fotografiju na LCD displeju. Međutim, neke složenije stvari poput balansa boja i tonova ne možemo dobro odrediti isključivo gledanjem snimljene fotografije. Razumevanje histograma je verovatno najvažnija stvar za ispravan rad i potpuno razumevanje digitalne fotografije. Histogram nam može reći da li je ekspozicija prilikom snimanja bila dobra, kakvo je osvetljenje, te njegovom primenom možemo naknadno popraviti već snimljene fotografije.

Prilikom snimanja digitalnom kamerom zavisno od rezolucije CCD čipa (čip sa spregnutim naelektrisanjem), snimamo određeni broj piksela. Informacija o boji čuva se u RGB formatu, tj.

informacija o boji za svaki piksel je kombinacija triju primarnih boja: crvene, zelene i plave. Za svaku boju koristimo 8 bita, a to znači 24 bita po pikselu. To zapravo znači da možemo sačuvati $2^8=256$ nijansi crvene, zelene ili plave boje. U današnje vreme sasvim je normalno da digitalne kamere imaju rezoluciju 5 megapiksela ($5 \cdot 10^6$ piksela). Očito je da snimanjem digitalnih fotografija čuvamo ogromnu količinu podataka. Tako veliku količinu podataka je izuzetno teško proceniti, ako gledamo svaki piksel zasebno. Kao što je navedeno u uvodu, histogram je pogodan za korišćenje kada imamo jako veliki broj uzoraka, koje je sasvim nepraktično posmatrati pojedinačno. Zbog toga je primena histograma u digitalnoj fotografiji bitna.

Jedna od stvari koja nas zanima na digitalnoj fotografiji je koliko su snimljeni pikseli svetli ili tamni, odnosno osvetljenje fotografije. Osvetljenje pojedinog piksela je predstavljeno celobrojn timer, pozitivnom vrednošću između 0 i 255. 0 predstavlja potpuno tamno, a 255 potpuno svetlo. Ako ne govorimo o fotografiji u boji nego samo o sivoj skali (engl. *grayscale*) 0 predstavlja crnu boju, a 255 belu. Kada se radi o fotografijama u boji tada je logično najtamnija nijansa određene boje označena sa 0, a najsvetlija nijansa sa 255.

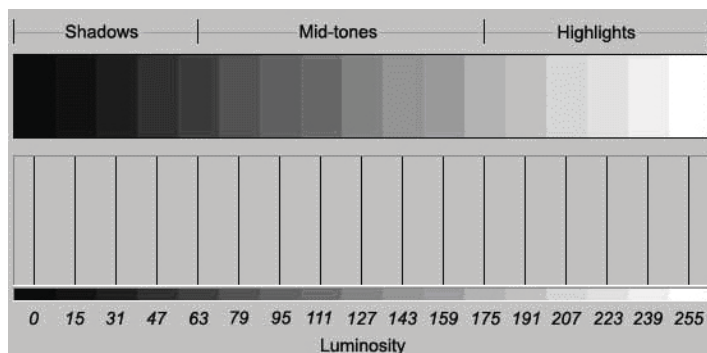
Histogram zapravo posmatra fotografiju u celini i određuje koliki broj piksela ima određeno (isto) osvetljenje. To znači da će na apscisi biti vrednosti osvetljenja od 0 do 255, a na ordinati broj piksela koji imaju odgovarajuće osvetljenje. Valja приметiti da je i ovde prisutna, za histograme karakteristična podela na kategorije, s tom razlikom da sada kategorija nije raspon između dva broja nego kvantovana vrednost između 0 i 255. Na slici 6.35. prikazan je primer histograma digitalne fotografije. Deo histograma gde je smešten najveći broj vrednosti osvetljenja naziva se raspon tonova (engl. *tonal range*). Raspon tonova jako varira za različite fotografije i ne postoji neki idealni histogram koji bi trebalo primenjivati na sve fotografije.



Slika 6.35 Primer histograma

6.9.1 Primeri histograma

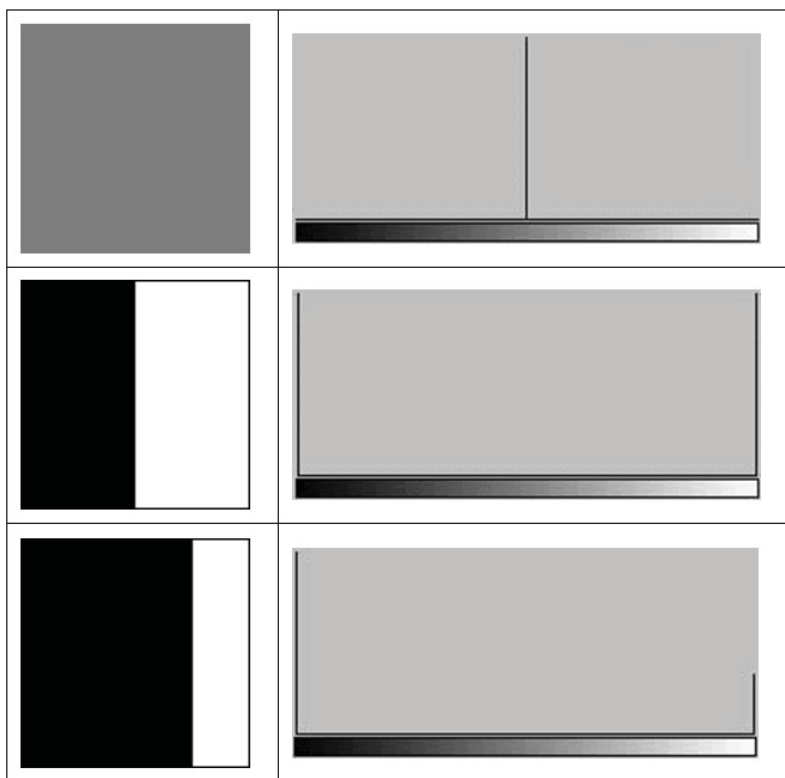
U ovom delu navodim nekoliko jednostavnih histograma radi boljeg razumevanja kako se od fotografije dobija histogram. Radi jednostavnosti posmatramo slike koje se sastoje od različitih nijansi sive boje (*grayscale*). Na slici 6.36. prikazana je slika koja se sastoji od stepeničasto raspoređenih nijansi sive boje, a odmah ispod nje prikazan je njezin histogram.



Slika 6.36 Primer histograma sive skale

Broj piksela svakog odsečka („stepenice”) slike je 2048, tako da su svi stupci histograma jednake visine i predstavljaju vrednost od 2048 piksela. Inače se na ordinati histograma ne označava kvantitativna vrednost broja piksela, već postoji fiksna visina kojoj se dodeljuje najveći broj piksela koji imaju isto osvetljenje. Visine ostalih stubaca se podešavaju relativno prema najvišem stupcu.

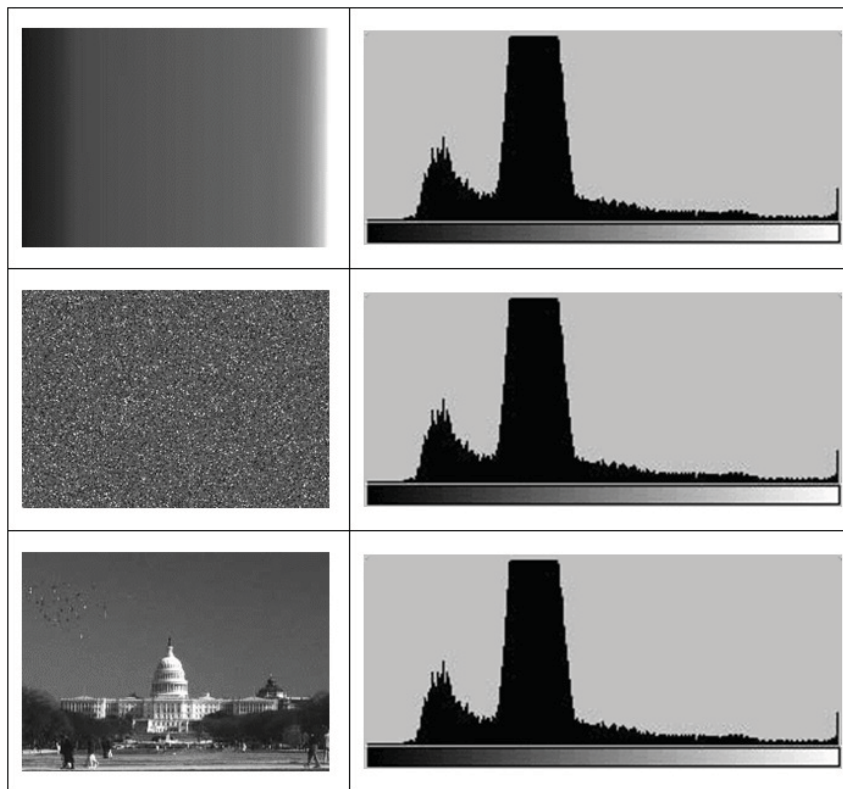
Na slici 6.37. prikazano je nekoliko jednostavnih slika na levoj strani, dok su na desnoj strani odgovarajući histogrami.



Slika 6.37 Primeri histograma jednostavnih slika

Bitno je naglasiti da izgled histograma ne zavisi od rasporeda piksela na slici, niti od orijentacije slike, već samo od toga koliki broj piksela ima isto osvetljenje. To znači da bi histogram sa slike

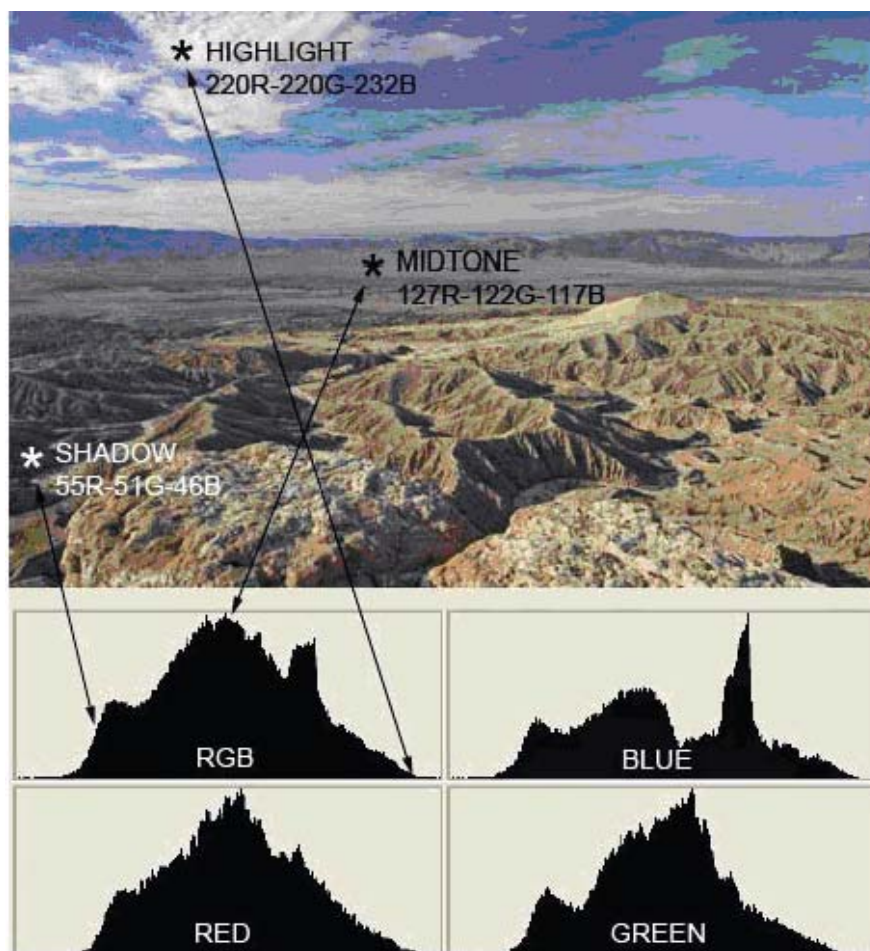
6.36 izgledao isto bez obzira ako bi npr. izvornu sliku zarotirali za 180 stepeni. Dodatni primer nezavisnosti histograma od rasporeda piksela prikazan je na slici 6.38. Prikazane su potpuno 3 različite slike. Ove slike se sastoje od istih piksela, a razlikuju se po razmeštaju piksela. Za sve 3 slike dobijamo identične histograme.



Slika 6.38 Nezavisnost histograma od rasporeda piksela

6.9.2 RGB histogram i histogrami po bojama

U prethodnom odeljku prikazani su primeri histograma jednostavnih slika u sivoj skali. U tim slikama svaki piksel je opisan jednom nijansom sive boje, gde 0 znači crnu, a 255 belu. Međutim siva boja se u digitalnom obliku čuva kao kombinacija crvene, zelene i plave, ali tako da sve 3 boje uvek imaju istu vrednost osvetljenja. Tako kombinacija $0R+0G+0B$ predstavlja crnu, $127R+127G+127B$ sivu u sredini skale, a $255R+255G+255B$ belu boju. To zapravo znači da su svi histogrami prikazani u prethodnom delu takođe RGB histogrami, kao i oni koje dobijemo iz fotografija za koje kažemo da su „u boji”. RGB histogram je najčešće korišćeni histogram. Histogram po boji (engl. *color histogram*) može prikazati 3 odvojena histograma za svaku boju, a RGB histogram će biti kompozicija ova 3 histograma. RGB histogram nastaje tako da prvo odredimo tri nezavisna histograma po bojama, koje zatim spajamo u jedan nezavisno od toga koja je boja došla od kog piksela. Ovo će biti puno jasnije ako pogledamo sledeću sliku, na kojoj je prikazana originalna fotografija, te odgovarajući histogrami (R, G, B i RGB).

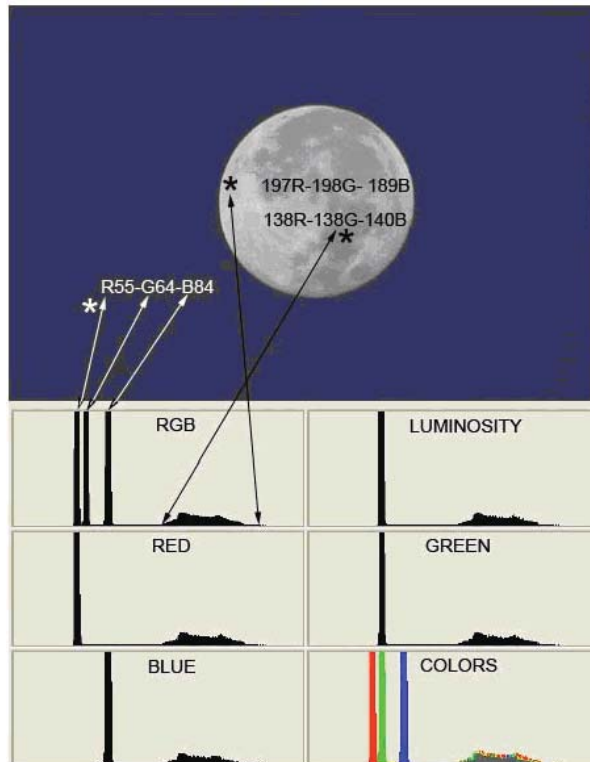


Slika 6.39 RGB histogram i histogrami po bojama

Histogram osvetljenja je takođe kompozicijski RGB histogram, međutim on uzima u obzir činjenicu da je ljudsko oko najosetljivije na promenu zelene boje, zatim crvene i na kraju plave. Histogram osvetljenja još se naziva i histogram sjaja (engl. *brightness*) zato što prikazuje kompenzirani sjaj fotografije, koji ljudsko oko percipira.

Kako nastaje histogram osvetljenja? Prvo se svaki piksel konvertuje tako da predstavlja ponderisano osvetljenje svake od 3 osnovne boje. Svetlost zelene boje uzima se sa težinskim faktorom od 59%, crvene boje sa 30% i plave sa 11%. Nakon što su svi pikseli konvertovani histogram osvetljenja nastaje prebrojavanjem i određivanjem koliki broj piksela ima koje osvetljenje, identično kao što i nastaju histogrami za svaku pojedinu boju.

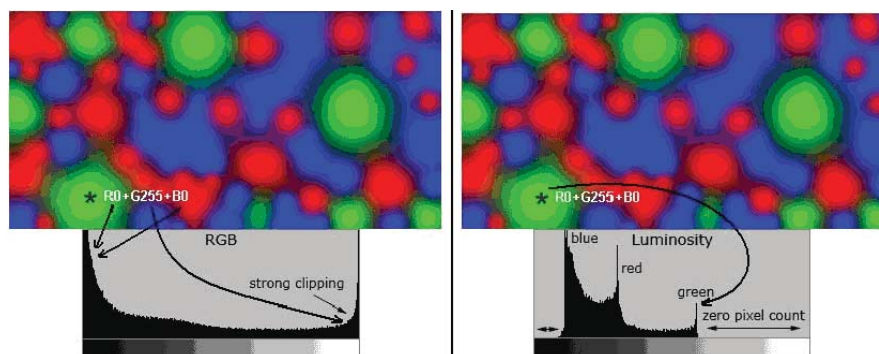
Na slici 6.40 prikazana je jedna fotografija sa svim pripadajućim histogramima. Oblik histograma je atipičan, međutim dobro ilustruje kako nastaje histogram osvetljenja. Histogram osvetljenja označen je engleskom rečju *luminosity* (osvetljenje). Vidimo da je histogram osvetljenja najsljedniji histogramu zelene boje, a to je zato što se prilikom konvertovanja piksela vrednost zelene boje uzima sa težinskim faktorom od 59%.



Slika 6.40 Pored ostalih histograma prikazan je i histogram osvetljenja

6.9.3 Histogram osvetljenja

Osnovna razlika između RGB histograma i histograma osvetljenja je ta da histogram osvetljenja zadržava informaciju o lokaciji boja svakog pojedinog piksela, dok RGB ovu informaciju gubi, zato što on nastaje kompozicijom tri histograma komponentnih boja. Ovo se najbolje može ilustrovati primerom sa slike 6.41. Posmatrana tačka (piksel) se sastoji od sledećih komponenti osnovnih boja: $R0+G255+B0$. Na RGB histogramu crvena i plava komponenta ove tačke smeštene su krajnje levo, a zelena komponenta krajnje desno. Kod histograma osvetljenja sve tri komponente tog piksela posmatraju se kao celina, koja je zbog toga smeštena u sredini histograma.



Slika 6.41 RGB histogram i histogram osvetljenja.

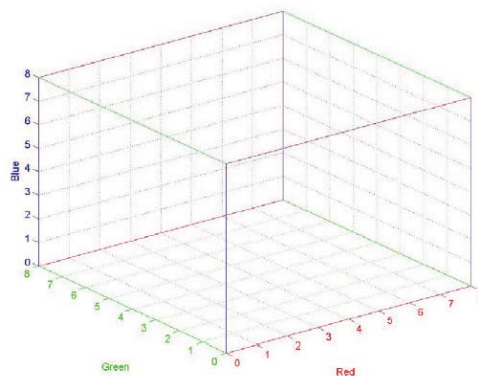
Vidimo da se histogrami iste slike značajno razlikuju. Valja primetiti da su vrhovi na istogramu osvetljenja raspoređeni u skladu sa težinskim faktorima, prema kojima se određena boja uzima u obzir, a oni iznose: 59%, 30% i 11%.

6.9.4 Globalni histogrami boje

Jedno od najpopularnijih obeležja za indeksiranje i pronalaženje slika je boja. Poređenje raspodele boja dve slike često može da nagovesti stepen njihove sličnosti.

Međutim, poređenje svih boja dve slike može da bude vremenski zahtevno i složeno, stoga se mora pribegavati redukovanju količine informacija. Jedan od načina da se to uradi je kvantizovanje raspodele boja u histograme boja. To su uveli Swan i Balard [108], a kasnije su to primenili i mnogi drugi. Danas je to možda jedan od najpopularnijih pristupa.

Kod izračunavanja histograma boje za sliku, različite ose boja se dele na određen broj kategorija (intervala). Tro-dimenzionalni 8x8x8 RGB histogram (slika 6.42) stoga sadrži ukupno 512 takvih kategorija. Kod indeksiranja slike, utvrđuje se boja svakog piksela i inkrementira se brojač odgovarajuće kategorije.



Slika 6.42 Prazan 3-dimenzionalni 8x8x8 RGB-histogram

6.9.5 Poređenje histograma

Kada se slike kvantizuju u histograme, potrebna je metoda za njihovo poređenje. Među najpopularnijim metrikama za poređenje histograma su $L1$ i $L2$ norma definisane formulama:

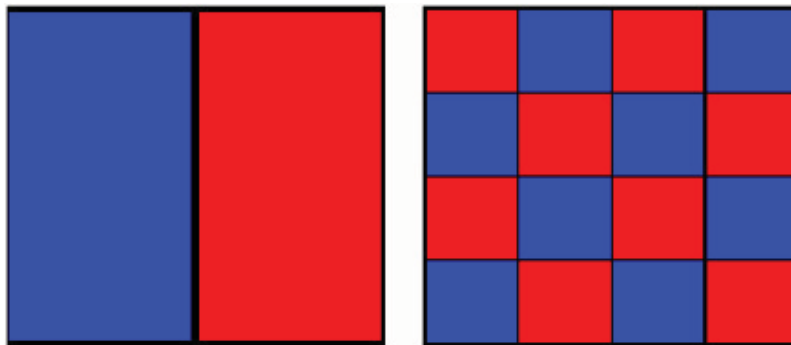
$$L1 = \sum_{i=1}^n |Q_i - I_i| \quad (6.1)$$

$$L2 = \sqrt{\sum_{i=1}^n (Q_i - I_i)^2} \quad (6.2)$$

Gde je Q_i vrednost kategorije i slike upita a I_i je vrednost odgovarajuće kategorije u bazi slika.

6.9.6 Lokalni histogrami boja

Jedna osobina histograma boje koja je ujedno njegova i pogodnost i mana je nedostatak prostorne informacije. To može biti pogodnost pošto globalni histogram date slike će ostati isti ako se ona rotira ili simetrično preslika. Međutim, to znači i da će dve perceptualno vrlo različite slike sa sličnim raspodelama boje biti „proglašene” sličnim u sistemu pretraživanja slika koji je zasnovan na histogramu boja. Primer koji to ilustruje prikazan je na slici 6.43.



Slika 6.43 Dve perceptualno različite slike sa jednakom raspodelom boje

Najprostiji i najočigledniji način za uvođenje prostorne informacije je metoda koju su koristili Gong i drugi [109]. Oni su podelili slike na devet jednakih delova i računali histograme za svaki od njih. Ovo daje izvesnu prostornu osetljivost, ali povećava obim izračunavanja i zahteve za smeštanjem podataka. Takođe, gubi se neosetljivost na rotaciju koja postoji kod globalnih histograma boje.

Shengjui Wang [110] je predložio varijantu lokalnih histograma boja koju je nazvao Harbinov pristup. Ova metoda takođe deli sliku na neki broj regiona iste veličine i izračunava njihove lokalne histograme boje. Razlika u odnosu na Gongov pristup ogleda se u metodi poređenja slika. Harbin pristup koristi sistem ponderisanih bipartitnih grafova za izračunavanje minimalne distance između dve slike. Kod ove metode svaki deo jedne slike poredi se sa svim delovima druge slike. Usled toga metoda je manje osetljiva na rotaciju u poređenju sa Gongovom metodom i nekim drugim.

Liu, Phillips i Hartman [111] predložili su pristup koji ne koristi regione iste veličine za izračunavanje lokalnih histograma boje. Oni su predložili sistem koji kreira dva histograma, jedan za prednji plan a drugi za pozadinu. Ovo treba da prevaziđe problem tzv. maskirajuće pozadine. Maskirajuća pozadina se javlja kada slika, na primer, prikazuje mali automobil na zelenom travnjaku. Zeleni travnjak zahvata veći deo slike i zelena boja će biti znatno izražena u globalnom histogramu boje. Ovo predstavlja problem u slučaju kada korisnik želi da pronađe više slika koje prikazuju mali automobil u prednjem planu. Definisan kao minimalni pravougaonik koji sadrži najvažnije objekte na slici, prednji plan bi bio pronađen automatski određivanjem vertikalnih i horizontalnih prelaza pikselских vrednosti ili postavljen ručno od strane korisnika.

6.9.7 Problemi kod histograma boja

Kao što je već rečeno, osnovni nedostatak globalnog histograma boje je gubljenje prostorne informacije. Međutim, to nije jedini problem. Drugi opšti problem je međukategorijska (međuintervalska) sličnost.

Kod kvantizovanja slika, svaki piksel se smešta u određenu kategoriju, prema svojoj boji. Neminovno, to će dovesti do situacije da se perceptualno veoma slične boje postave u susedne kategorije. To može zatim da dovede do razlike između dva histograma koja je mnogo veća nego perceptualna razlika između aktuelnih slika. Da bi prevazišli taj problem, Flickner, Sawhney i drugi [112] predložili su ponderisanu $L2$ metriku za IBM-ov QBIC sistem.

Sledeći problem kod histograma boja je definisanje broja kategorija koje će se koristiti. Mali broj kategorija smanjuje zahteve za smeštajnim prostorom i vremenom potrebnim za indeksiranje i pronalaženje, ali redukuje i efektivnost. U svojoj doktorskoj tezi Jeff Berens [113] je utvrdio da je idealan broj kategorija kod RGB prostora boja oko $8 \times 8 \times 8$ (512). Veći broj kategorija od ovog obezbeđuje samo vrlo mali porast efektivnosti na račun velikog opadanja brzine rada.

6.10 Obeležja Tamura

Smith i Chang [114] su predložili šest obeležja teksture koji odgovaraju čovekovoj vizuelnoj percepciji: grubost (engl. *coarseness*), kontrast (engl. *contrast*), usmerenost (engl. *directionality*), linijaturnost (engl. *line-likeness*), pravilnost (engl. *regularity*) i hrapavost (engl. *roughness*). Iz eksperimentalnog testiranja značajnosti ovih obeležja s obzirom na čovekovu percepciju zaključeno je da su prva tri obeležja važna. Stoga su Deselaers i saradnici [115] u svojim eksperimentima koristili grubost, kontrast i usmerenost da kreiraju histogram koji opisuje teksturu i poredili te histograme koristeći Jeffreyovu divergenciju. QBIC sistem takođe koristi histograme tih obeležja.

6.11 Obeležje globalne teksture

U [116] opisano je obeležje teksture koje se sastoji od nekoliko delova: fraktal dimenzija meri hrapavost površine. Fraktal dimenzija se izračunava koristeći metod brojanja retikularne ćelije (engl. *reticular cell counting method*). Grubost karakteriše veličinu zrna slike. Izračunava se zavisno od varijanse slike. Entropija vrednosti piksela se koristi kao mera neuređenosti (engl. *disorderedness*) u slici. Statistika prostorne razlike nivoa sivog opisuje relacije sjajnosti (engl. *brightness*) piksela unutar susedstva. Takođe, to je poznato kao analiza matrice istovremenog pojavljivanja (engl. *co-occurrence matrix analysis*). Kružna Moranova autokorelaciona funkcija meri hrapavost teksture. Za izračunavanje se koristi skup autokorelacionih funkcija. Iz toga se dobija 43-dimenzioni vektor koji se sadrži od jedne vrednosti za fraktalnu dimenziju, jednu vrednost za grubost, jednu vrednost za entropiju i 32 vrednosti za statistike razlike i 8 vrednosti za kružnu Moranovu autokorelacionu funkciju. Ovo obeležje se uspešno koristi kod medicinskih slika.

6.12 Gabor obeležje

Gabor obeležja se široko koriste za analizu teksture. Deselaers i saradnici su koristili dva različita obeležja izvedena iz Gabor obeležja:

- Srednja vrednost i standardna devijacija: Gabor obeležja su izdvojena pri različitim skalama i pravcima iz slika a izračunate su srednja vrednost i standardna devijacija filtarskih odziva. Izdvojena Gabor obeležja u 5 različitih orijentacija i 5 različitih skala dala su 50 dimenzioni vektor.
- Banka od 12 različitih kružno simetričnih Gabor filtara primenjena je na sliku, energija za svaki filter banke kvantovana je u 10 opsega a histogram od srednjih vrednosti izlaza filtra na regionima slike je izračunat da bi se dobila globalna mera obeležja teksture slike. Ovi histogrami su upoređivani pomoću JSD.

6.13 Histogrami invarijantnog obeležja

Obeležje se naziva invarijantnim s obzirom na izvesnu transformaciju ako se ne menja kada se te transformacije primene na sliku. Ovde razmatrane transformacije su translacija, rotacija i skaliranje.

Ova obeležja su zasnovana na ideji konstruisanja invarijantnih obeležja integracijom, tj. određena funkcija obeležja se integriše na skupu svih navedenih transformacija. Funkcije obeležja koje su Deselaers i saradnici razmatrali su nomijalne i relacione funkcije nad intenzitetima piksela. Umesto sumiranja nad translacijom i rotacijom, oni su sumirali samo nad rotacijom i kreirali histogram nad translacijom. Taj histogram je još uvek invarijantan s obzirom na rotaciju i translaciju. Dobijeni histogrami se porede koristeći JSD. Prethodni eksperimenti su pokazali da su karakteristike histograma invarijantnog obeležja i histograma boje vrlo slični i da ti histogrami invarijantnog obeležja mogu ponekad zameniti histograme boje [117].

6.14 Lokalna obeležja slike

Parčad slike ili obeležja izvedena od toga su trenutno veoma obećavajući pristup u prepoznavanju objekata [118], [119], [120]. Jasno, prepoznavanje objekata i CBIR su tesno povezana polja i za neke jasno definisane zadatke pretraživanja, metode prepoznavanja objekata mogu biti jedino moguće rešenje, na primer traženje svih slika koje pokazuju neku osobu, jasno sistemi za prepoznavanje i detekciju lica daju najbolje rezultate [121], [122].

Tabela 6.3.a CBIR sistemi – korišćena obeležja

SISTEM	Država	Ekstrakcija boje	Ekstrakcija teksture	Ekstrakcija oblika
CALIPH&EMIR	Austria	Da	Da	-
CIRES	USA	Da	Da	Ne
COMPASS	Italy	Da	Da	Ne
CORTINA	USA	Da	Da	Ne
FIDS	USA	Da	Da	Ne
FIRE	Germany	Da	Da	Da
GIFT/VIPER	Switzerland	Da	Da	Ne
Img(Anaktisi)	Greece	Da	Da	Ne
Img(Rummager)	Greece	Da	Da	Ne
IMAGEROVER		Da	Da	Ne
IMAGESCAPE	Netherland	Da	Da	Da
IMAGE-SEEKER	France	Da	Ne	Da
IMEDIA	France	Da	Da	Da
IMGSEEK	Brasil	Da	Ne	Ne
IRMA	Germany	Da	Da	Da
LIRE	Austria	Da	Da	Ne
LPCD	Netherland	Ne	Da	Ne
MARS	USA	Da	Da	Da
PIC2SEEK	Netherland	Da	Ne	Da
PICSOM	Finland	Da	Da	Da
QBIC	USA	Da	Da	Da
QUICKLOOK	Italy	Da	Da	Da
RETRIEVR	Austria	-	Ne	Ne
SHAPE SIMILARITY	USA	Da	Da	Da
SIMPLICITY	USA	Da	Da	Da
SQUID	United Kingdom	Ne	Ne	Da
TILTOMO	United Kingdom	Da	Da	Ne
VIRAGE		Da	Da	Da
WEBSEEK	USA	Da	Ne	Ne

Tabela 6.3.b CBIR – upit i relevantnost

SISTEM	Upit (primerom)	Povratna informacija o relevantnosti	Formati slike	Softverska aplikacija
CALIPH&EMIR	No	No		Web based
CIRES	No	No	jpg, gif	
COMPASS	No	No		
CORTINA	No	Ne		
FIDS	No	Ne	jpg	
FIRE	No	No	jpg	
GIFT/VIPER	Ne	No	jpg	
Img(Anaktisi)	No	No	jpg	Web based
Img(Rummager)	No	No	jpg	Desktop
IMAGEROVER				
IMAGESCAPE	Ne	Ne		
IMAGE-SEEKER	No	Ne	jpg, gif, bmp, ...	
IMEDIA	No	No		
IMGSEEK	No	Ne	jpg, gif, ...	
IRMA	No	No	radiology	
LIRE	No	No		Web based
LPCD	No	No		
MARS	No	No		Web based
PIC2SEEK	No	Ne		
PICSOM	No	No		
QBIC	No	Ne		
QUICKLOOK	Ne	No		
RETRIEVR	Ne	Ne	jpg, gif, png	
SHAPE SIMILARITY	No	Ne		
SIMPLICITY	No	Ne	jpg	
SQUID	No	Ne		
TILTOMO	No	Ne		
VIRAGE	No	No	jpg, ...	Web based/desktop
WEBSEEK	No	No		Web based

6.15 MPEG-7

MPEG-7 je razvijen od strane *Moving Pictures Expert Group* [123]. Za razliku od prethodnih MPEG standarda (MPEG-1, MPEG-2, MPEG-4) koji su opisivali kodiranje audio-vizuelnih sadržaja, MPEG-7 se fokusira na davanje informacija o sadržaju. Cilj MPEG-7 standarda je davanje opširnog skupa standardizovanih alata za opis multimedijalnog sadržaja.

Standardom MPEG-7 se standardizuje:

- Skup tzv. opisivača (deskriptora; engl. *descriptors*) koji služe za opisivanje raznih obeležja multimedijalnog sadržaja,
- predefinisane strukture opisivača i relacije među njima, tzv. opisne šeme (engl. *Description Schemes - DSs*),
- jezik za definisanje opisivača i opisnih šema (engl. *Description Definition Language - DDL*),
- kodirana reprezentacija opisivača radi smanjenja veličine datoteka i bržeg pristupa.

Svaki deskriptor/opisivač definisan je pomoću tzv. normativnog i nenormativnog dela. Normativni deo sastoji se od sintakse, semantike i binarne reprezentacije opisivača. Nenormativni deo je opcionalan i sastoji se od preporučenih metoda ekstrakcije i tzv. metoda prepoznavanja sličnosti. Obeležja niskog nivoa (engl. *low-level*) sadržaja mogu se izdvojiti iz sadržaja automatski. Preporučeni algoritmi za njihovo izdvajanje nalaze se u nenormativnom delu nekih opisivača. Isto se odnosi i na metode prepoznavanja sličnosti parametara opisivača. Za ovaj rad bitni su samo deskriptori slike

Različiti prostori boje se koriste u MPEG-7 uključujući dobro poznate monohromatski, RGB, HSV, YCrCb i novi HMMD (engl. *Hue-Min-Max-Difference*). Monohromatski (samo intenzitet) prostor je takođe podržan. On odgovara Y komponenti u YCrCb prostoru. Moguće je da se definiše RGB prema referenci hrome primara, ako je na raspolaganju od procesa apstrahovanja. Konverzija iz normalizovanog RGB (kod koga su vrednosti svake spektralne komponente u opsegu od 0 do 1) u druge prostore boja prikazana je na slici 6.44.

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ Cb &= -0.169 \cdot R - 0.331 \cdot G + 0.500 \cdot B \\ Cr &= 0.500 \cdot R - 0.419 \cdot G - 0.081 \cdot B \end{aligned}$$

(a)

```
Max = max(R, G, B); Min = min(R, G, B);
Value = max(R, G, B);
if( Max == 0 ) then
    Saturation = 0; else
    Saturation = (Max-Min)/Max;
if( Max == Min ) Hue is undefined (achromatic color);
otherwise:
if( Max == R && G > B ) Hue = 60*(G-B)/(Max-Min)
else if( Max == R && G < B ) Hue = 360 + 60*(G-B)/(Max-Min)
else if( G == Max ) Hue = 60*(2.0 + (B-R)/(Max-Min))
else Hue = 60*(4.0 + (R-G)/(Max-Min))
```

(b)

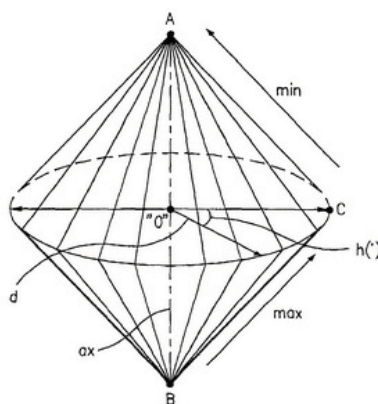
```
Diff=Max-Min
Sum=(max+min)/2
Hue as defined for the HSV.
```

(c)

Slika 6.44 Prostori boje korišćeni u MPEG-7. (a) Konverzija RGB u YCbCr. (b) Konverzija RGB u HSV. (c) Konverzija RGB u HMMD.

HSV prostor boje je popularan izbor za manipulisanje bojom. On je razvijen da obezbedi intuitivno predstavljanje boje i da aproksimira način na koji ljudi opažaju i manipulišu bojama. Pretvaranje RGB u HSV je nelinearna, ali reverzibilna transformacija. Hue (H - nijansa) predstavlja dominantnu spektralnu komponentu – boju u svom čistom obliku, kao zeleno, crveno ili žuto. Dodavanje bele čistoj boji menja boju: manjebele, boja je zasićenija. Ovo korespondira sa zasićenjem (engl. *Saturation* – *S*). Vrednost (engl. *Value* – *V*) korespondira sa svetlinom boje. Koordinatni sistem je cilindričan i često se predstavlja podprostorom definisanim šesto-stranom invertovanom piramidom. Vrh piramide odgovara vrednosti $V = 1$, sa „belom” u centru. Hue se meri uglom oko vertikalne ose, sa crvenom koja odgovara uglu od 0 stepeni. Zasićenje je u opsegu od 0 (centar) do 1 (na površini piramide). Invertovani konus se takođe koristi da označi potprostor umesto piramide.

Novi prostor boje, HMMD prostor boje, takođe je podržan u MPEG-7. *Hue* ima isto značenje kao u HSV prostoru, a *max* i *min* su maksimum i minimum između R, G i B vrednosti respektivno. Komponenta *diff* je definisana kao razlika između max i min. Samo tri od četiri komponente su dovoljne da opišu HMMD prostor. Ovaj prostor boje može se opisati pomoću duple konusne strukture prikazane na slici 6.45. U MPEG-7 baznim eksperimentima pretraživanja slika opaženo je da je HMMD prostor boje veoma efektivan. Treba imati u vidu da je HMMD prostor boje neznatno uvijen u odnosu na HSI prostor boje, kod koga je komponenta *diff* skalirana sa vrednošću intenziteta. HMMD prostor boje se koristi kod deskriptora strukture boje (engl. *Color Structure Descriptor* – *CSD*).



Slika 6.45 HMMD prostor boje

Da bi se obezbedila interoperabilnost, dozvoljeni prostori boja za različite deskriptore boje su ograničeni standardom. Deskriptor dominantne boje (engl. *Dominant Color Descriptor* – *DCD*) dozvoljava specifikaciju boje u bilo kom prostoru boje podržanom od strane MPEG-7. RGB prostor boje nije naročito efikasan u zadacima traženja i pronalaženja i ne koristi se eksplicitno ni u jednom deskriptoru boje. SCD koristi HSV prostor a histogram strukture boje koristi HMMD prostor. CLD je definisan za YcrCb prostor. Ovi deskriptori prostora boje se takođe koriste izvan vizuelnih deskriptora, na primer, kod specifikiranja „svojstava medija” u odgovarajućim šemama opisa.

6.15.1 Izbor obeležja/deksriptora

Pronalaženje slika na osnovu sadržaja preduzima ekstrakciju obeležja iz slika, koja se potom koriste u proceduri pronalaženja. Vizuelni sadržaj slika mapira se u novi prostor, koji se naziva prostor obeležja. Obeležja moraju da budu diskriminativna i dovoljna za opisivanje slike. U suštini, ključ za postizanje uspešnog sistema za pronalaženje je u izboru pravih obeležja (deskriptora) koja predstavljaju slike što je moguće „jače” i jedinstvenije.

Međutim, pitanje koje se u poslednjih par godina sve češće postavlja je kako postići da ova obeležja budu što kompaktnija. Opisivanje slike sa vektorom velike dimenzije može da ima vrlo dobre rezultate pronalaženja ali znatno usporava proceduru pronalaženja i povećava zahteve za smeštajnim prostorom.

MPEG-7 deskriptori koji su predloženi od strane MPEG grupe za indeksiranje i pronalaženje, održavaju balans između veličine obeležja i kvaliteta rezultata. Ovi deskriptori su relativno kompaktni i prilično dobro opisuju vizuelni sadržaj slike.

U ovom radu za opis sadržaja slika odlučeno je da se koriste SCD i EHD MPEG-7 deskriptori.

Osvrt na neke matematičke pojmove

Šta su funkcije baze?

Jednostavnije je objasniti funkcije baze ako se preselimo iz analognog sveta (funkcije) u digitalni svet (vektori). Svaki dvo-dimenzioni vektor (x, y) je kombinacija vektora $(1, 0)$ i $(0, 1)$. Ta dva vektora su vektori baze za (x, y) . Zašto? Zato što x pomnoženo sa $(1, 0)$ daje vektor $(x, 0)$ a y pomnoženo sa $(0, 1)$ daje vektor $(0, y)$. Suma je (x, y) .

Najbolji vektori baze imaju dodatno važno svojstvo, a to je da su perpendikularni (ili ortogonalni) jedan u odnosu na drugi. Kod baze $(1, 0)$ i $(0, 1)$ taj uslov je zadovoljen.

Vratimo se sada nazad u analogni svet i pogledajmo kako tu stvari funkcionišu. Umesto vektora (x, y) , imamo funkciju $f(x)$. Zamislimo da je $f(x)$ muzički ton, recimo nota A u određenoj oktavi. Možemo da konstruišemo A dodavanjem sinusa i kosinusa koristeći kombinacije amplituda i frekvencija. Sinusi i kosinusi su funkcije baze u ovom primeru i elementi Furijeove sinteze. Za izabrane sinuse i kosinuse možemo postaviti dodatni zahtev da budu ortogonalni. Kako? Birajući odgovarajuću kombinaciju funkcija sinusnih i kosinusnih izraza čiji unutrašnji proizvod je nula. Konkretni skup funkcija koje su ortogonalne i konstruišu $f(x)$ su naše ortogonalne funkcije baze za ovaj problem.

Šta su funkcije baze varirajuće skale?

Funkcije baze vriraju po skali seckajući istu funkciju ili prostor podataka koristeći skale različite veličine. Na primer, zamislimo da imamo signal u domenu od 0 do 1. Signal možemo podeliti sa dve „step” funkcije koje su u opsegu od 0 do $\frac{1}{2}$ i od $\frac{1}{2}$ do 1. Zatim možemo opet podeliti orogonalni signal koristeći četiri „step” funkcije od 0 do $\frac{1}{4}$, $\frac{1}{4}$ do $\frac{1}{2}$, $\frac{1}{2}$ do $\frac{3}{4}$ i $\frac{3}{4}$ do 1. I tako redom. Svaki skup reprezentacija kodira ortogonalni signal sa konkretnom rezolucijom ili skalom.

Šta su talasići?

Talasići su skup nelinearnih baza (osnova; engl. *basis*). Kod projektovanja (tj. razlaganja ili aproksimacije) funkcije na talasiće, funkcije baze talasića biraju se prema funkciji koja se aproksimira. Stoga, za razliku od familija linearnih osnova gde se isti, statični skup funkcija osnove koristi za svaku ulaznu funkciju, talasići koriste dinamički skup funkcija osnove koji predstavlja ulaznu funkciju na najefikasniji način. Stoga talasići mogu da obezbede veliki stepen kompresije i zato su popularni u oblasti obrade slike i signala.

Razmotrimo detaljnije kako se ovaj dinamički skup funkcija osnove bira i kako se ulazna funkcija transformiše u talasiće.

Pretpostavimo da imamo string brojeva (2, 2, 2, 2) i da želimo da ga prenesemo preko mreže. Svakako, želimo to da uradimo na najefikasniji mogući način, što implicira da želimo da pošaljemo najmanju moguću količinu podataka. Razmotrimo sada koje opcije su na raspolaganju. Trivijalno, jedna opcija je da pošaljemo sva četiri broja, tj. pošaljemo prvu dvojku, zatim drugu i tako redom. Radeći na takav način, implicitno bismo sledeće baze/osnove:

$\langle 1, 0, 0, 0 \rangle$

$\langle 0, 1, 0, 0 \rangle$

$\langle 0, 0, 1, 0 \rangle$

$\langle 0, 0, 0, 1 \rangle$

Međutim, to nije najbolji način da se obavi postavljeni zadatak. Može li to da se uradi bolje? Trik je u izboru osnove koja predstavlja naše podatke efikasno i na što kompaktniji način. Posmatrajući navedene podatke odmah uočavamo njihovu uniformnost; ustvari to je konstantan signal veličine 2. Poželjno je iskoristiti tu uniformnost. Ako izaberemo vektor osnove $\langle 1, 1, 1, 1 \rangle$, možemo navedene podatke predstaviti samo jednim brojem! Treba preko mreže da šaljemo samo jedan broj 2, a celokupni string podataka može da se rekonstruiše množenjem (ili ponderisanjem) sa vektorom osnove $\langle 1, 1, 1, 1 \rangle$. To je sjajno, ali nama su potrebna još tri vektora osnove da kompletiramo našu osnovu pošto se u našem primeru radi o prostoru sa 4 dimenzije. Poznato je da svi vektori osnove moraju da budu ortogonalni (ili perpendikularni). To znači da ako skalarno pomnožimo bilo koja dva vektora osnove rezultat će biti nula. Prema tome, naš zadatak je da izračunamo vektor koji je ortogonalan sa $\langle 1, 1, 1, 1 \rangle$. Jedan takav vektor je $\langle 1, 1, -1, -1 \rangle$. Ako izračunamo skalarni (engl. *dot product*) proizvod ta dva vektora, rezultat je zaista nula. Ta dva vektora možemo grafički predstaviti u sledećem obliku:



Grafički prikaz ovih vektora baze podseća na „talasiće”, otuda i naziv talasići (engl. *wavelets*). Sada imamo dva vektora baze, znači potrebna su nam još dva. Haar je konstruisao preostale vektore osnove dilatacijom i translacijom. Dilatacija u suštini znači skupljanje; dakle preostali

vektori baze su konstruisani skupljanjem i transliranjem. Ako skupimo vektor $\langle 1, 1, -1, -1 \rangle$ dobijamo $\langle 1, -1, 0, 0 \rangle$. Par 1, 1 se skupio u jedno 1, a na sličan način par -1, -1 postaje jedno -1. Zatim, transliranjem novodobijenog vektora baze dobija se $\langle 0, 0, 1, -1 \rangle$, koji je četvrti vektor baze našeg prostora. Ta dva vektora imaju sledeći grafički prikaza:



Sada imamo kompletnu bazu za naš 4-dimenzionalni prostor, koju sačinjavaju sledeći vektori baze ili talasići:

$\langle 1, 1, 1, 1 \rangle$
 $\langle 1, 1, -1, -1 \rangle$
 $\langle 1, -1, 0, 0 \rangle$
 $\langle 0, 0, 1, -1 \rangle$

Skalarnim množenjem može se proveriti da su ovi vektori međusobno perpendikularni. Mada su ovi vektori baze ortogonalni, oni nisu ortonormalni. Međutim, možemo ih lako normalizovati izračunavanjem veličine svakog od ovih vektora a zatim deljenjem njihovih komponenti sa tom veličinom.

$\langle 1, 1, 1, 1 \rangle$	--> magnitude = $\sqrt{(1)^2 + (1)^2 + (1)^2 + (1)^2} = 2$	--> $\langle 1/2, 1/2, 1/2, 1/2 \rangle$
$\langle 1, 1, -1, -1 \rangle$	--> magnitude = $\sqrt{(1)^2 + (1)^2 + (-1)^2 + (-1)^2} = 2$	--> $\langle 1/2, 1/2, -1/2, -1/2 \rangle$
$\langle 1, -1, 0, 0 \rangle$	--> magnitude = $\sqrt{(1)^2 + (-1)^2 + (0)^2 + (0)^2} = \sqrt{2}$	--> $\langle 1/\sqrt{2}, -1/\sqrt{2}, 0, 0 \rangle$
$\langle 0, 0, 1, -1 \rangle$	--> magnitude = $\sqrt{(0)^2 + (0)^2 + (1)^2 + (-1)^2} = \sqrt{2}$	--> $\langle 0, 0, 1/\sqrt{2}, -1/\sqrt{2} \rangle$

Sad kada imamo na raspolaganju kompletnu bazu, možemo razmotriti kako se projektuje ulazni vektor na talasiće. Ovo je u literaturi poznato kao Haarova 1D transformacija.

Haarova 1D transformacija

Neka je naš ulazni vektor oblika $\langle 4, 2, 5, 5 \rangle$. Da bismo ga razložili na talasiće jednostavno skalarno ga množimo sa svakim vektorom baze.

$\text{dot} (\langle 4, 2, 5, 5 \rangle , \langle 1/2, 1/2, 1/2, 1/2 \rangle) = 8$
$\text{dot} (\langle 4, 2, 5, 5 \rangle , \langle 1/2, 1/2, -1/2, -1/2 \rangle) = -2$
$\text{dot} (\langle 4, 2, 5, 5 \rangle , \langle 1/\sqrt{2}, -1/\sqrt{2}, 0, 0 \rangle) = 2/\sqrt{2}$
$\text{dot} (\langle 4, 2, 5, 5 \rangle , \langle 0, 0, 1/\sqrt{2}, -1/\sqrt{2} \rangle) = 0$

Na taj način ulazni vektor je transformisan u $\langle 8, -2, 2/2, 0 \rangle$. Uočavamo da je četvrta komponenta 0!. To znači da nam nije potreban četvrti vektor baze, tj. možemo da rekonstruišemo naš originalni

ulazni vektor samo sa prva tri vektora baze. Drugim rečima, **dinamički** biramo tri vektora baze od četiri moguća prema našem ulazu.

$$\begin{aligned}
 8 * \langle 1/2, 1/2, 1/2, 1/2 \rangle &= \langle 4, 4, 4, 4 \rangle \\
 -2 * \langle 1/2, 1/2, -1/2, -1/2 \rangle &= \langle -1, -1, 1, 1 \rangle \\
 2/\sqrt{2} * \langle 1/\sqrt{2}, -1/\sqrt{2}, 0, 0 \rangle &= \langle 1, -1, 0, 0 \rangle \\
 \text{add the vectors} &= \langle 4, 2, 5, 5 \rangle
 \end{aligned}$$

Do sad imali smo 4-komponentni ulazni vektor i odgovarajući skup od četiri 4-komponentna vektora baze. Šta se dešava ako imamo veći ulazni vektor, recimo sa 8 komponentata? Da li je potrebno da nađemo nove vektore baze? Odgovor je ne. Možemo čak da koristimo manju bazu od one koju već imamo. Ustvari, možemo da koristimo najprostiju bazu talasića koja se sastoji od: $\langle 1/\sqrt{2}, 1/\sqrt{2} \rangle$ i $\langle 1/\sqrt{2}, -1/\sqrt{2} \rangle$. To su najmanji talasići i oni se ne mogu više skupiti. Međutim, biranjem ovih manjih vektora baze za veći ulaz, nije više moguće primeniti Haarovu transformaciju talasića u jednom prolazu kao što smo prethodno radili. U ovom slučaju moraćemo rekurzivno da transformišemo ulazni vektor sve dok ne dobijemo finalni rezultat. Za ilustraciju uzmimo jednostavnu 2-komponentnu bazu da transformišemo 4-komponentni ulazni vektor koji smo koristili u prethodnom primeru. Algoritam je izložen u sledećoj tabeli.

Tabela 6.4 Haarov algoritam

Algoritam	Primer
Ulazni vektor:	$\langle 1/\sqrt{2}, 1/\sqrt{2} \rangle$ i $\langle 1/\sqrt{2}, -1/\sqrt{2} \rangle$
Transformisani vektor (koeficijenti baze)	(To treba da se izračuna)
Vektori baze (talasići)	
1. Ako je veličina (dimenzija) vektora manja od veličine vektora baze, stavi ulazni vektor u transformisani vektor i napusti proceduru	Veličina ulaznog vektora = 4 Veličina vektora baze = 2 $4 > 2$, nastavi dalje
2. Podeli ulazni vektor u delove koji po veličini (dimenziji) odgovaraju talasićima	$\langle 4, 2 \rangle, \langle 5, 5 \rangle$
3. Izračunaj skalarni proizvod prve baze i svakog dela ulaznog vektora	$\text{dot}(\langle 4, 2 \rangle, \langle 1/\sqrt{2}, 1/\sqrt{2} \rangle) = 6/\sqrt{2}$ $\text{dot}(\langle 5, 5 \rangle, \langle 1/\sqrt{2}, 1/\sqrt{2} \rangle) = 10/\sqrt{2}$
4. Izračunaj skalarni proizvod druge baze i svakog dela ulaznog vektora	$\text{dot}(\langle 4, 2 \rangle, \langle 1/\sqrt{2}, -1/\sqrt{2} \rangle) = 2/\sqrt{2}$ $\text{dot}(\langle 5, 5 \rangle, \langle 1/\sqrt{2}, -1/\sqrt{2} \rangle) = 0$
Rezultat 3-eg koraka je novi ulazni vektor:	$\langle 6/\sqrt{2}, 10/\sqrt{2} \rangle$
Rezultat 4-og koraka popunjava delove transformisanog vektora:	$\langle ?, ?, 2/\sqrt{2}, 0 \rangle$

5. Skok na korak 1.	Veličina ulaznog vektora = 2 Veličina vektora baze = 2 $2 = 2$, nastavi dalje
	$\text{dot}(<6/\sqrt{2}, 10/\sqrt{2}>, <1/\sqrt{2}, 1/\sqrt{2}>) = 8$
	$\text{dot}(<6/\sqrt{2}, 10/\sqrt{2}>, <1/\sqrt{2}, -1/\sqrt{2}>) = -2$
Ulazni vektor:	$<8>$
Ažurirani transformisani vektor:	$<?, -2, 2/\sqrt{2}, 0>$
	Veličina ulaznog vektora = 1 Veličina vektora baze = 2 $1 < 2$, postavi ulazni vektor u transformisani i napusti proceduru
Transformisani vektor (finalni rezultat):	$<8, -2, 2/\sqrt{2}, 0>$

Ovo je vrlo jednostavan algoritam. Sve što on radi je sabiranje i oduzimanje svakog para brojeva u ulaznom vektoru i njihovo deljenje sa kvadratnim korenom iz 2. Zatim se postupak ponavlja na ulaznom vektoru dobijenom od sumiranih članova.

Haarova 2D transformacija

Haarova 1D transformacija može se lako proširiti na 2D. U 2D slučaju radi se sa ulaznom matricom a ne sa ulaznim vektorom. Da bi transformisali ulaznu matricu, prvo primenjujemo Haarovu 1D transformaciju na svaki red (vrstu). Dobijamo rezultatnu matricu i zatim primenjujemo Haarovu 1D transformaciju na svaku kolonu. Kao rezultat dobijamo konačnu transformisanu matricu. Haarova 2D transformacija se masovno koristi kod komprimovanja slika. Sada možemo da se vratimo na opis nekih MPEG-7 deskriptora.

6.15.2 MPEG-7 deskriptor SCD

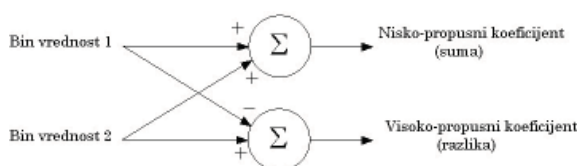
Generički *Color Histogram Descriptor (SCD)* definisan još u ranim MPEG-7 eksperimentima je složeni deskriptor koji se sastoji od deskriptora prostora boje, kvantizacije boje i histograma. Ovo treba da omogući specifikaciju histograma boje sa varirajućim brojem kategorija (engl. *Bins*) i neuniformnu kvantizaciju različitih prostora boje. Međutim, nije bilo poželjno da se obezbedi isuviše fleksibilnosti u takvoj specifikaciji, jer to može da ograniči interoperabilnost između različitih opisa baziranih na MPEG-7. SCD adresira problem interoperabilnosti fiksirajući prostor boje na HSV, sa uniformnom kvantizacijom HSV prostora u 256 kategorija. Vrednosti kategorija se neuniformno kvantizuju 11-bitnom vrednošću.

Ova metoda ostvaruje punu interoperabilnost između različitih rezolucija predstavljanja boje, počevši od 16 bitova/histogram na donjem kraju do aproksimativno 1000 bitova/histogram na gornjem kraju. Naravno, preciznost opisa obeležja je veoma zavisna od broja korišćenih bitova. Međutim, bazni eksperimenti su pokazali da su dobri rezultati pronalaženja ostvarivi koristeći samo 64 bita, dok se odlični rezultati mogu dobiti koristeći srednju ili punu rezoluciju deskriptora.

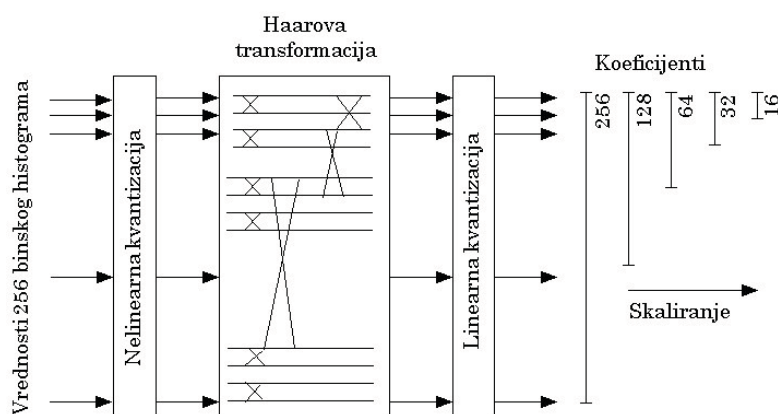
HSV prostor se uniformno kvantizuje u ukupno 256 kategorija. To obuhvata 16 nivoa u H, 4 nivoa u S i 4 nivoa u V. Vrednosti histograma se odsecaju u 11-bitnom celobrojnom predstavljanju. Da bi se postiglo efikasnije kodiranje, 11-bitne celobrojne vrednosti se prvo mapiraju (preslikavaju) u „nelineranu” 4-bitnu predstavu, dajući viši značaj malim vrednostima sa višom verovatnoćom.

Ovo 4-bitno predstavljanje 256-razrednog HSV histograma zahteva 1024 bitova/histogram, što je isuviše veliki broj u kontekstu većine MPEG-7 aplikacija. Da bi se umanjio ovaj broj i učinila aplikacija skalabilnom, histogrami se kodiraju pomoću Haarove transformacije.

Osnovna jedinica Haarove transformacije sastoji se od operacije sumiranja i operacije razlike (vidi sliku 6.46), koje su povezane sa primitivnim nisko- i visoko-propusnim filtrima. Sumiranje parova susednih linija histograma je ekvivalentno sa izračunavanjem histograma sa upola manje kategorija. Ako se proces obavlja iterativno, korišćenje podskupova koeficijenata u Haarovom predstavljanju je ekvivalentno sa histogramima od 128, 64, 32, ... kategorija, koji se svi izračunavaju iz izvornog histograma.



Slika 6.46 Osnovna jedinica Haarove transformacije



Slika 6.47 Šematski dijagram generisanja SCD deskriptora

Visoko-propusni (razlika) koeficijenti Haarove transformacije izražavaju informaciju sadržanu u nivoima finije rezolucije (sa višim brojem kategorija) histograma. Signali obično pokazuju visoku redundansu između susednih linija histograma. Ovo se može objasniti sa „nečistoćom”, tj. neznatnim varijacijama boja izazvanih promenljivom iluminacijom i efektima senke. Stoga, može se očekivati da visoko-propusni koeficijenti izražavajući razlike između susednih kategorija histograma obično imaju samo male vrednosti. Koristeći ovo svojstvo moguće je odseći/svesti visoko-propusne koeficijente na celobrojnu predstavu sa manjim brojem bitova.

Slika 6.47 prikazuje blok dijagram kompletnog sistema. Izlaz je skalabilan u smislu broja kategorija variranjem broja korišćenih koeficijenata. Interoperabilnost između različitih nivoa rezolucije je zadržana zbog svojstva skaliranja Haarove transformacije. Prema tome, poklapanje bazirano na informacijama iz podskupova koeficijenata garantuje aproksimaciju. Tabela 6.5 pokazuje relaciju između brojeva Haarovih koeficijenata kako je specificirano u SCD i particija u komponentama odgovarajućeg HSV histograma koji može da se rekonstruiše iz koeficijenata.

Tabela 6.5. Ekvivalentno particionisanje HSV prostora boja za različite konfiguracije SCD deskriptora

16	4	2	2
32	8	2	2
64	8	2	4
128	8	4	4
256	16	4	4

Drugačiji tip skalabilnosti je ostvaren skaliranjem kvantizovane (celobrojne) predstave koeficijenata na različite brojeve bitova. Koeficijenti „razlike” u Haarovoj transformaciji mogu imati ili pozitivne ili negativne vrednosti. Deo za znak se uvek zadržava pošto deo za magnitudu može da se skalira preskačući najmanje značajne bitove. Korišćenje samo bit znaka (1 bit/koeficijent) dovodi do izuzetno kompaktnog predstavljanja, dok se zadržava dobra efikasnost pronalaženja. Na najvišem nivou preciznosti, 1 – 8 bitovi se definišu za celobrojna predstavljanja dela za magnitudu, u zavisnosti od važnosti respektivnih koeficijenata. Između tih ekstrema moguće je skalirati na različite nivoe rezolucije. Na primer, razmotrimo skupa od pet koeficijenata čije magnitude su kodirane koristeći 8, 4, 7, 3 i 7 bitova, respektivno. Ako su najniža 3 bita odbačena kod skalabilnog bit predstavljanja, samo 5, 1, 4, 0 i 4 bita ostaje da kodira apsolutnu vrednost.

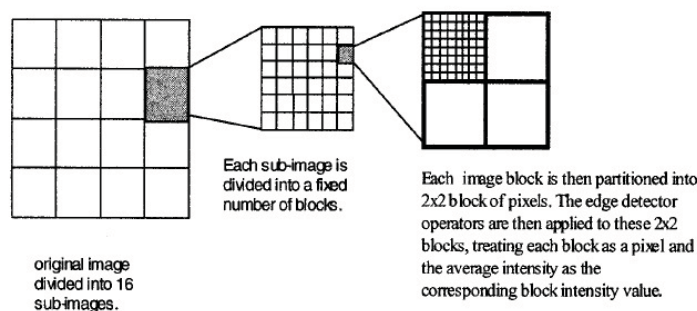
Kod poklapanja sličnosti histograma, norma $L1$ (suma apsolutnih razlika) obično daje dobru preciznost pronalaženja. Isto tako poklapanje bazirano na $L1$ normi može se primeniti na domen Haarove transformacije; međutim, rezultati nisu identični (osim za slučaj kada „visoko-propusni” koeficijenti imaju identične znake kod dva deskriptora koja se porede), kao direktno poklapanje u domenu histograma. U svakom slučaju kada samo bit znaka se koristi (sve bitske ravni koje predstavljaju apsolutne vrednosti su odbačene), norma $L1$ degeneriše u Hammingovu distancu, dozvoljavajući čak manju kompleksnost u traženju.

Za izračunavanje preciznosti pronalaženja kod SCD, koristi se $ANMMR$ mera.

6.15.3 MPEG-7 deskriptor EHD

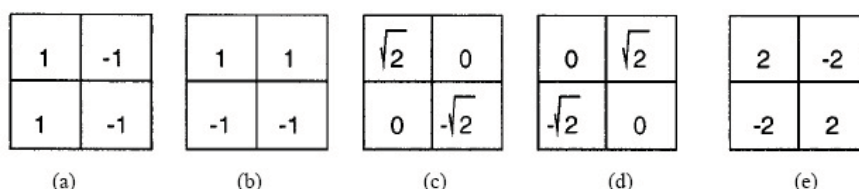
Deskriptor histograma ivica apsorbuje prostornu raspodelu ivica. Raspodela ivica je dobra oznaka tekture koja je korisna za poklapanje slika čak i kada analizirana tekstura nije homogena. Izračunavanje ovog deskriptora je direktno (slika 6.48). Data slika se prvo podeli na 4×4 podslike (regiona) a zatim se izračunavaju lokalni ivični histogrami za svaku od podslika. Ivice su široko grupisane u pet kategorija: vertikalne, horizontalne, dijagonalne pod 45 stepeni, dijagonalne

pod 135 stepeni i izotropne (bez specifične orijentacije). Prema tome, svaki lokalni histogram ima pet kategorija koje odgovaraju navedenim kategorijama ivica. Obzirom da je slika podeljena na 16 podslika dobija se 80 kategorija. Ove kategorije se neuniformno kvantizuju koristeći 3 bita/kategorija, što daje deskriptor veličine 240 bitova.



Slika 6.48 Izračunavanje deskriptora histograma ivica

Da bi se izračunali histogrami ivica, svaka od 16 podslika se dalje deli u blokove. Veličina ovih blokova skalira se sa veličinom slike i podrazumeva se da je stepen broja 2. Broj blokova po slici drži se konstantnim, nezavisno od dimenzija originalne slike, skalirajući njihovu veličinu na odgovarajući način. Zatim se primenjuje jednostavni detektor ivice na svaki makro-blok, tretirajući makro-blok kao sliku 2x2 piksela. Intenziteti piksela za 2x2 particije bloka se izračunavaju usrednjavanjem vrednosti intenziteta odgovarajućih piksela. Operatori detekcije ivica obuhvataju četiri detektora selektivnog pravca i jedan izotropni operator (slika 6.49). Oni blokovi slike čije jačine ivice premašuju određeni minimalni prag se koriste u izračunavanju histograma.



Slika 6.49 Filtri za detekciju ivica

Prema tome, za jedan blok slike, možemo da izračunamo pet jačina ivica, po jedna za svaki od pet filtara sa slike 6.49. Ako maksimum ovih jačina premaši određeni unapred postavljeni prag, tada se odgovarajući blok slike smatra ivičnim blokom. Ivični blok učestvuje u kategorijama ivičnog histograma. Metoda izračunavanja ivica je sasvim jednostavna i može se direktno primeniti na MPEG-2 kompresovane bit strimove.

Svaki od blokova slike označen kao ivični blok „upada” u odgovarajuću kategoriju histograma deskriptora. Ove vrednosti se normalizuju na opseg [0, 1]. Nelinearna kvantizacija vrednosti kategorija daje predstavu 3 bita/kategorija.

6.15.4 Povezivanje deskriptora SCD i EHD sa Lucene bibliotekom

Lucene biblioteka se koristi za indeksiranje tekstualnog sadržaja poslovnih dokumenata. Slike sa prvih strana dokumenata, ukoliko ih ima, se ekstrahuju u slikovne objekte. Korišćenjem softverske biblioteke za MPEG-7 deskriptore koja je portovana na C# moguće je ekstrahovati SCD i EHD deskriptore tih slika. Ti deskriptori su numerički vektori ali se mogu posmatrati kao niska karaktera (string, tj. tekst). Svaka slika praktično postaje entitet koji je u relaciji sa pripadajućim poslovnim dokumentom iz koga je ekstrahovana i njeni deskriptori predstavljeni kao string mogu se indeksirati korišćenjem biblioteke Lucene i uključiti kao polje u zapis Lucene dokumenta koji sadrži i ostale attribute tog poslovnog dokumenta.

6.16 Projektni zahtevi

Realizovati desktop aplikaciju sa grafičkim korisničkim interfejsom za *ad-hoc* pretraživanje elektronskih poslovnih dokumenata koja treba:

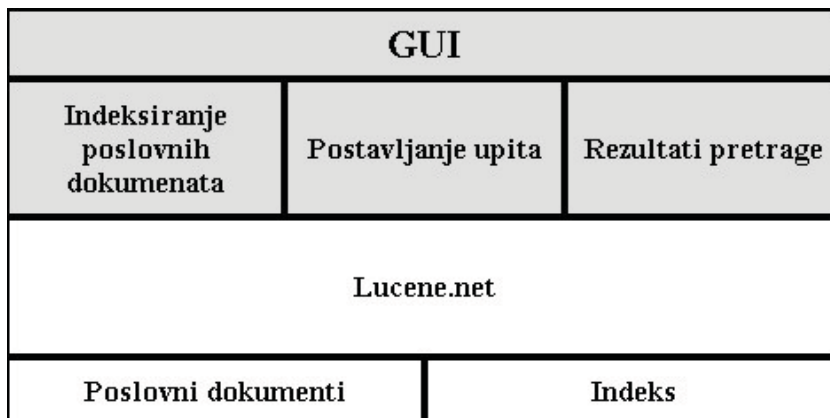
- da se pokreće na zahtev korisnika;
- da omogući korisniku biranje lokacije koja se indeksira;
- da obavesti korisnika ukoliko ima problema prilikom indeksiranja;
- da omogući korisniku da bira vrstu upita koji će postaviti, tekstualni, slikovni ili kombinovani;
- da vizuelno prikaže rezultate pretraživanja u obliku liste.

7. Implementacija, eksperimenti i rezultati

7.1 Implementacija

Analiziranjem projektnih zahteva i razmatranjem raspoloživih razvojnih alata, odlučeno je da se aplikacija implementira na programskom jeziku *C#* u razvojnom okruženju *Microsoft Visual Studio 2005*. Programski jezik *C#* je dobro dokumentovan, na raspolaganju je dosta stručne literature, zatim poseduje veliku kolekciju gotovih grafičkih kontrola za realizaciju grafičkog korisničkog interfejsa. Integrirano razvojno okruženje *Microsoft Visual Studio* je pogodno za razvoj aplikacija obzirom da raspolaže tzv. *intellisense* interfejsom i dobrim debagerom.

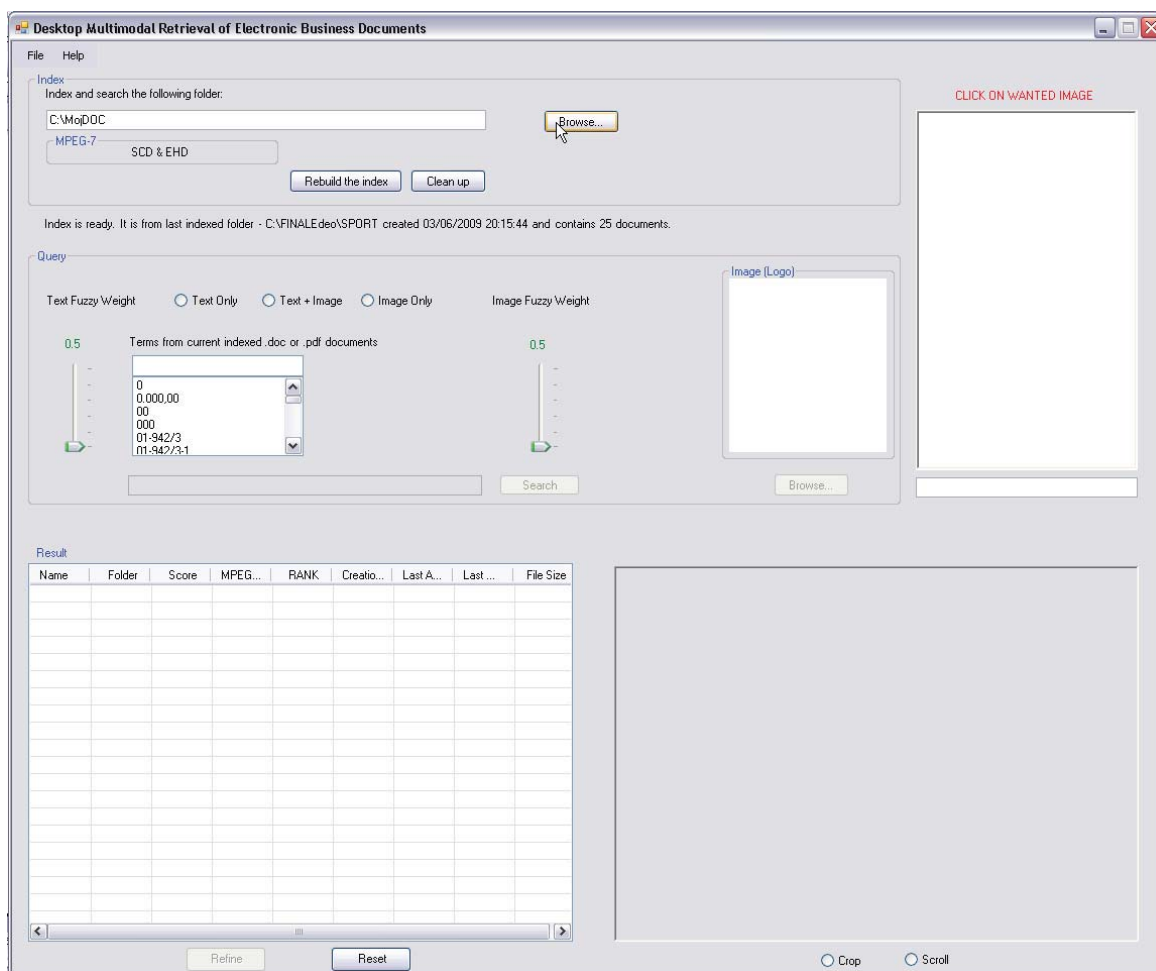
Aplikacija se sastoji od tri modula: grafičkog korisničkog interfejsa (GUI), modula poslovne logike i mašine za pretraživanje (Lucene.net).



Slika 7.1 Globalna struktura sistema

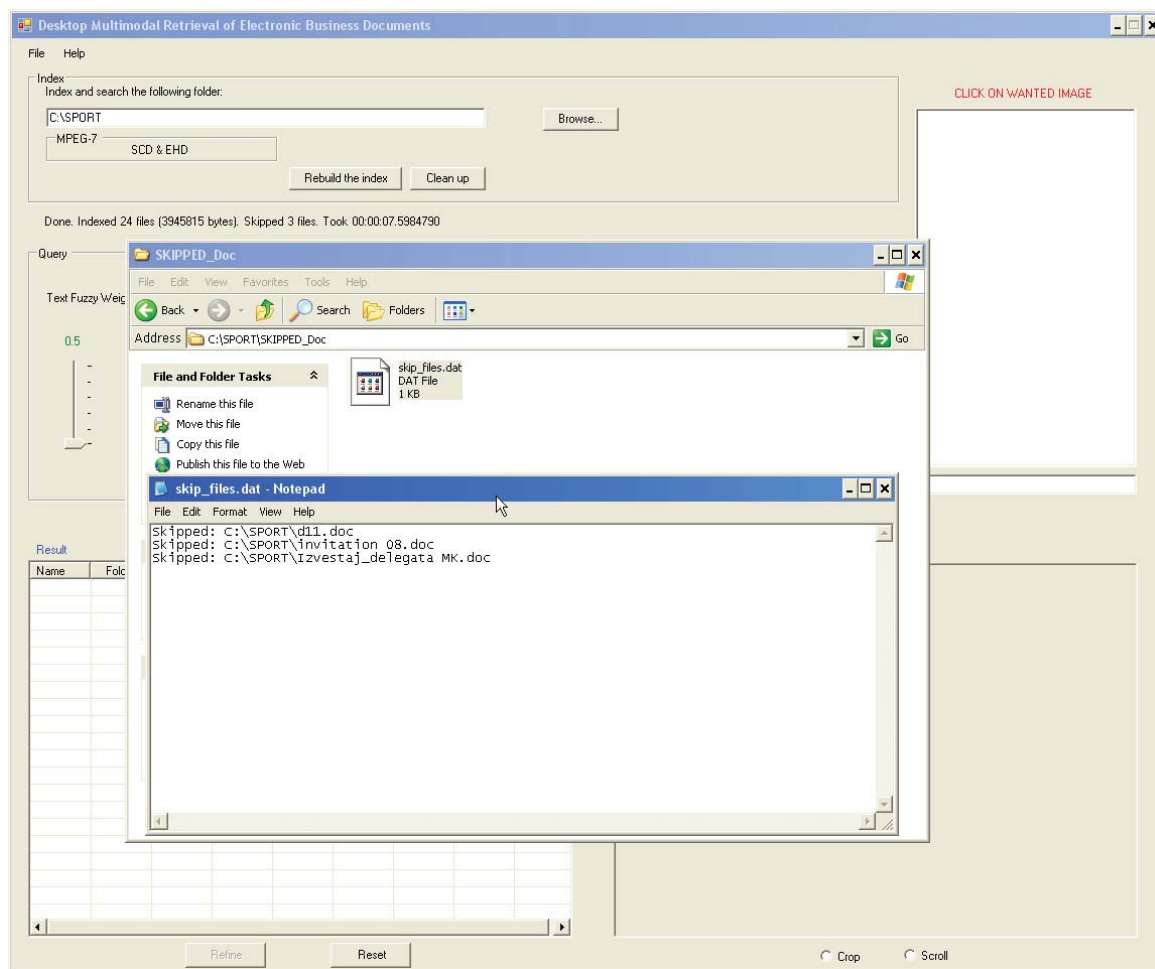
Grafički korisnički interfejs

Aplikacija ima grafički korisnički interfejs prikazan na slici 7.2 i raspolaže uobičajenom funkcionalnošću za taj tip aplikacija.



Slika 7.2 Grafički korisnički interfejs

Korisnik ima mogućnost da definiše folder čiji sadržaj će da indeksira. Program proverava da li već postoji indeksni fajl, kada je kreiran, koliko dokumenata je indeksirano i iz kog foldera. Ukoliko je indeksiran folder koji ne odgovara aktuelnom zahtevu korisnika, korisnik može da pokrene indeksiranje sadržaja (PDF, DOC i RTF dokumenti) željenog foldera, usled čega se stari indeksni fajl uništava i kreira novi. Po obavljenom indeksiranju program obaveštava korisnika o obavljenom indeksiranju, tj. koliko je vremenski indeksiranje trajalo, koliko dokumenata je indeksirano i da li su neki dokumenti prilikom indeksiranja preskočeni (bilo da su nekorektnog formata ili su u vreme indeksiranja bili u nadležnosti nekih drugih procesa na računaru). U tom slučaju on beleži putanju i ime fajla u tekstualnu datoteku *Skipped.dat* koja se smešta u podfolder *SKIPPED_Doc* koji kreira program u folderu koji se indeksira. Korisnik naknadnim pregledom tog fajla može da utvrdi o kojim dokumentima se radi i da ih ručno pregleda.



Slika 7.3 Prikaz sadržaja datoteke Skipped.dat

Tokom indeksiranja foldera aplikacija generiše listu termina bez ponavljanja iz obrađenih dokumenata. Ti termini su u formi padajuće liste i stavljaju se na raspolaganje korisniku da lakše formuliše svoje upite. Takođe, tokom procesa indeksiranja aplikacija vrši izdvajanje rezimea (sumar) iz pojedinačnih dokumenata (pojedinačno sumiranje dokumenata). Upotrebljena je verzija klasifikatorske biblioteke *Classifier4J* koja je portovana na C# i adaptirana za potrebe ove aplikacije. Rezimei su kao meta-podatak uključeni u pripadajuće dokumente.

Ukoliko dokumenti na prvoj stranici sadrže slike, program vrši izdvajanje tih slika u poseban folder *ImgFromDoc* koji aplikacija kreira kao podfolder foldera koji se indeksira. Te slike korisnik može kasnije da koristi za definisanje slikovnog dela upita.

Korisnik može da bira da li će da pretražuje indeksirane dokumente po:

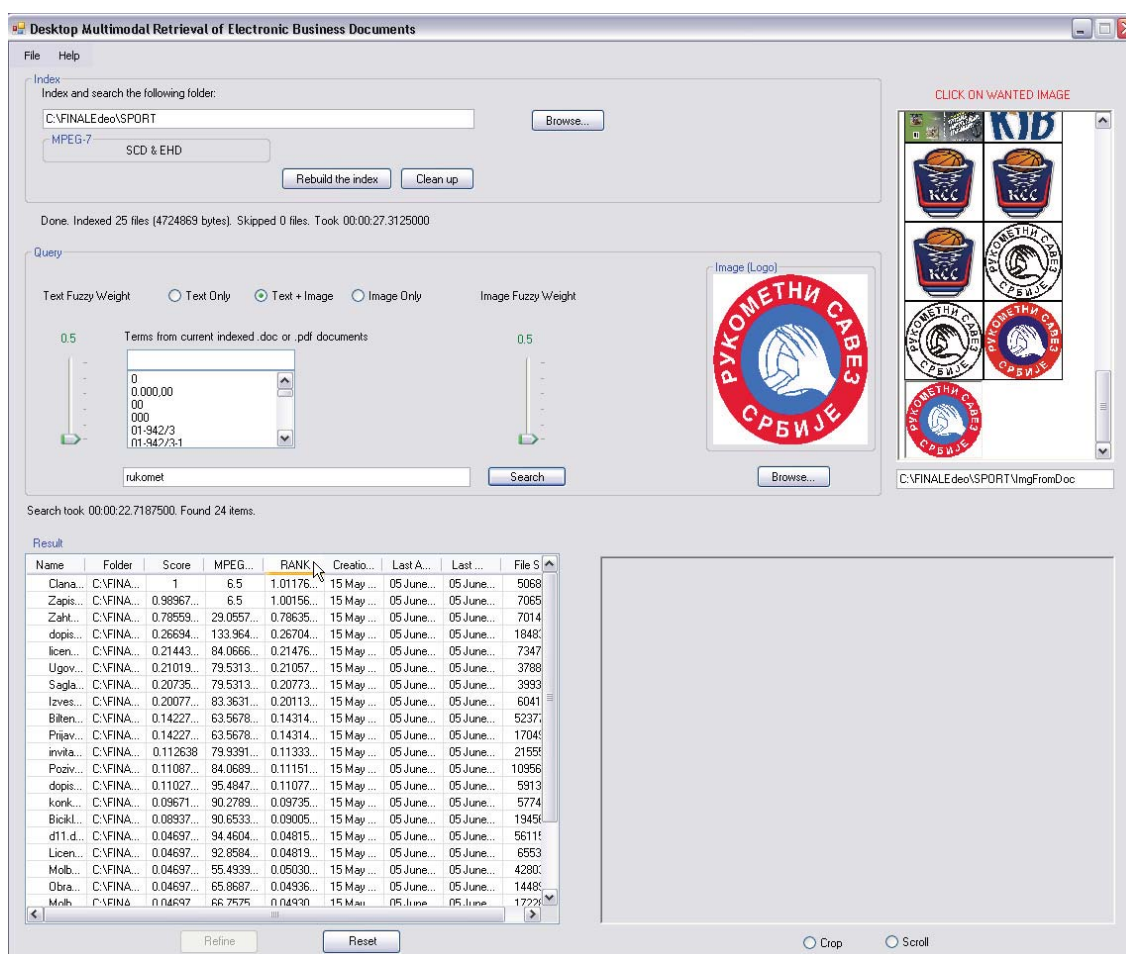
- tekstualnom,
- slikovnom ili
- kombinovanom (tekstualnom i slikovnom) sadržaju.

Tekstualni upit može da se definiše koristeći tekstualno polje, a što se tiče slikovnog upita, na raspolaganju je dugme za prelistavanje foldera u koji je aplikacija izdvojila slike sa prvih stranica

dokumenata ukoliko su ih imali. Na taj način korisnik je rasterećen problema „nulte slike” kod postavljanja ove vrste upita.

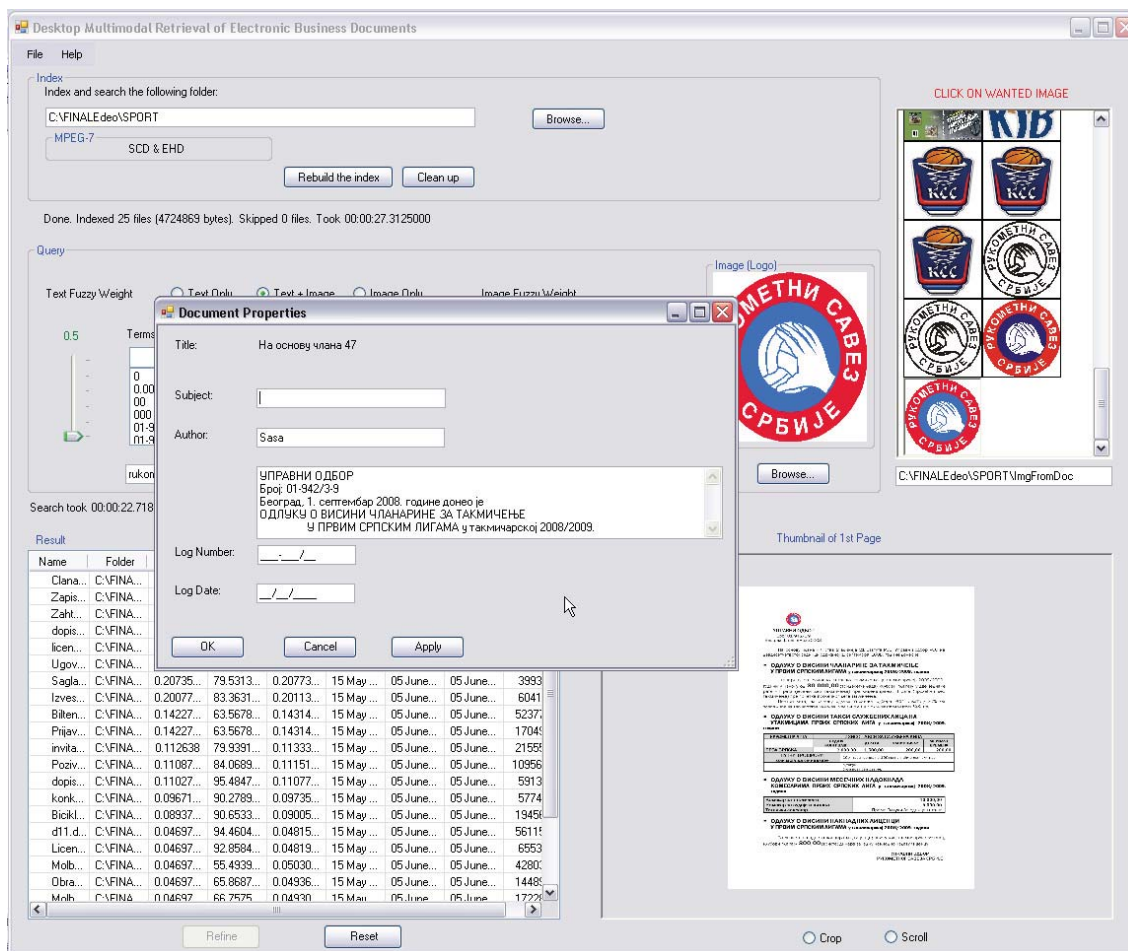
Kod kombinovanog pretraživanja, korisnik definiše tekstualni upit i bira sliku iz skupa slika izdvojenih iz obrađenih dokumenata. Korišćenjem grafičke kontrole *Slider* može da definiše vrednost fazi pondera za izabranu vrstu upita. Time definiše mašini za pretraživanje koji „stepen tolerancije” da primeni kod pretraživanja.

Po definisanju upita, korisnik klikom na dugme pokreće pretraživanje. Po obavljenom pretraživanju program generiše listu rezultata prema zadatom upitu. Svaka stavka liste sadrži putanju do dokumenta, naziv dokumenta, meru sličnosti u odnosu na postavljeni upit, datum nastanka dokumenta, veličinu fajla itd.



Slika 7.4 Definisane kombinovanog upita

Korisnik klikom miša na željenu stavku liste rezultata pretraživanja dobija na uvid preko grafičke forme meta-podatke selektovanog dokumenta (Author, Subject, Summary), kao i minijturni grafički prikaz slike cele prve stranice dokumenta, koji može da uveća i skroluje. Na osnovu ovih informacija korisnik može da proceni da li je to dokument koji mu odgovara ili ne.



Slika 7.5 Prikaz meta-podataka selektovanog dokumenta iz spiska

U ovoj fazi razvoja aplikacije nije implementiran modul koji bi omogućio reformulisanje upita uzimajući korisnikovu procenu relevantnosti dokumenta čime bi se omogućilo sužavanje liste pretrage.

Po pregledanju rezultata pretrage, korisnik može da obriše listu i postavi novi (ili preformulisani) upit koji se odnosi na indeksirani skup dokumenata.

Ukoliko želi da pretražuje na nekoj drugoj lokaciji, korisnik mora da navede novu putanju i zada aplikaciji da indeksira sadržaj te lokacije i po već navedenom scenariju obavi pretraživanje.

Modul poslovne logike

Kreiranje indeksa

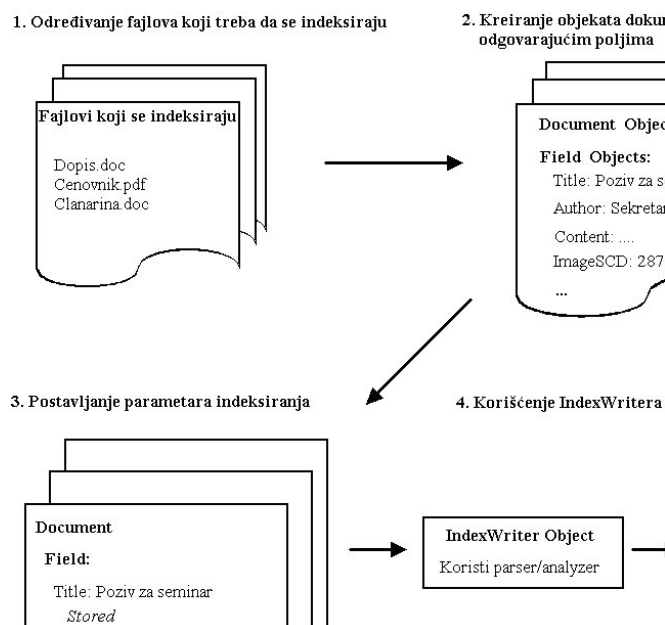
Indeksiranje je proces identifikovanja i analiziranja „sirovih“ podataka i njihovog konvertovanja u format koji će omogućiti biblioteci *Lucene.net* da te podatke brzo pretraži. Indeks koji kreira ova biblioteka je optimizovan za brz direktan pristup svim terminima smeštenim u indeks. Kada se kreira indeks na raspolaganju je opcija da se kreira više polja i smeste različiti podaci u svako polje. Na primer, ako se indeksiraju Microsoft Office (Word, Excel, ...) fajlovi, može se kreirati

polje za ime fajla, polje za datum i polje za telo dokumenta. Na taj način, prilikom pretrage može se suziti upit samo na imena fajlova, datume fajlova ili tela dokumenata ili se može kombinovati dva ili više polja u istom upitu i dobiti rezultati pretrage.

Lucene.net može da indeksira samo običan tekst a kako aplikacija treba da obrađuje formatirane poslovne dokumente (DOC, RTF i PDF format), potrebno je parsirati te dokumente i ekstrahovati tekst iz njih. Čitanje binarne strukture tih dokumenata nije tako jednostavan posao. Za tu namenu iskorišćen je interfejs *IFilter* koji je inače deo *Windows Indexing* servisa i instaliran je na Windows platformama. *IFilter* je skup *API* (*Application Programming Interface* – interfejs za programiranje aplikacija) funkcija koji je razvio Microsoft za svoje proizvode *Windows Desktop Search* i *Lookout*. Ova aplikacija komunicira sa *IFilter* interfejsom preko tzv. *COM interop* poziva. Ovaj interfejs može da parsira i druge tipove dokumenata ukoliko je na sistemu instaliran odgovarajući filter.

Poslovni dokumenti pored glavnog sadržaja mogu da sadrže meta-podatke koji se odnose na dokument, kao što su tema (*Subject*), autor (*Author*), rezime (*Summary*) itd. Ti podaci, ukoliko postoje, mogu da budu korisni kod pretraživanja informacija pošto upućuju na izvor dokumenta. Za pristupanje tim podacima kod DOC i RTF dokumenata, aplikacija koristi funkcije komercijalne softverske biblioteke *Aspose.Words* a u slučaju PDF dokumenata funkcije komercijalne softverske biblioteke *Aspose.Pdf.Kit*.

Da bi se indeksirao slikovni sadržaj poslovnih dokumenata, potrebno je iz tih dokumenata ekstrahovati slike. Kod poslovnih dokumenata ustaljena je praksa da se logo znaci nalaze u gornjem levom uglu prve stranice. Međutim, postoje izuzeci i zato je odlučeno da se ekstrahovanje slika ograniči na celu prvu stranicu. Za ekstrahovanje slika kod DOC i RTF dokumenata koristi se ranije pomenuti komercijalni proizvod *Aspose.Words* a u slučaju PDF dokumenata *Aspose.Pdf.Kit*.



Slika 7.6 Tok obrade poslovnih dokumenata pri indeksiranju

Sledeći segment koda pokazuje kako se kreira indeks i popunjava podacima.

```
private void Form1_Load(object sender, System.EventArgs e)
{
    // define folder that will be indexed
    this.textBoxPath.Text = Environment.GetFolderPath(Environment.SpecialFolder.
MyComputer); //Personal);

    // define folder for storing index
    this.pathIndex = Path.Combine(Environment.GetFolderPath(Environment.
SpecialFolder.LocalApplicationData), "DeskMMIR");

    // check if index exist
    checkIndex();
}

private void buttonIndex_Click(object sender, System.EventArgs e)
{
    // checking if user enter path to folder for indexing
    if (this.textBoxPath.Text.Trim(new char[] { ' ' }) == String.Empty)
    {
        string cautionindex = String.Format("You must BROWSE FOLDER!");
        status(cautionindex);
        return;
    }
    //
    // creating index
    //
    indexWriter = new IndexWriter(this.pathIndex, new StandardAnalyzer(), true);

    bytesTotal = 0;
    countTotal = 0;
    countSkipped = 0;

    // disable controls while app indexing documents
    enableControls(false);

    DirectoryInfo di = new DirectoryInfo(this.textBoxPath.Text);

    this.displayComboBox.Items.Clear();

    DateTime start = DateTime.Now;

    addFolder(di);

    string summary = String.Format("Done. Indexed {0} files ({1} bytes). Skipped {2}
files.", countTotal, bytesTotal, countSkipped);
    summary += String.Format(" Took {0}", (DateTime.Now - start));
    status(summary);
    enableControls(true);

    indexWriter.Optimize();
    indexWriter.Close();
    ...
    ...
}
```

Analizator, kombinovan sa stemerom, igra važnu ulogu u biblioteci *Lucene.net*. Za vreme indeksiranja, analizator i stemer prihvataju strim (tok) sirovog teksta i raščlanjuju ga u pretražive termine. Pored toga, oni uklanjaju svaki „šum” iz teksta (zapete, dvotačke, upitnike itd.), kao i uobičajene reči (veznike, uzrečice itd.). Uklanjanje šuma i uobičajenih reči znatno ubrzava pretragu.

Za indeksiranje ne-engleskih podataka, može se napraviti sopstveni analizator i stemer. Međutim, za nekoliko jezika je to već urađeno (danski, holandski, finski, francuski, nemački, italijanski, norveški, portugalski, ruski, španski i švedski). Za izradu sopstvenog analizatora, može se koristiti jedan od raspoloživih analizatora i stemera kao model.

Kako ne postoji implementiran analizator za srpski jezik, ova aplikacija koristi standardni analizator biblioteke *Lucene.net* koji je projektovan za rad sa engleskim tekstovima.

Uloga IndexWritera

Sledeći segment koda kreira ili otvara indeks. To se radi preko objekta *IndexWriter*. On se koristi kadgod treba nešto da se doda indeksu ili da se izbriše iz indeksa. Prvi parametar je putanja do indeksa. Drugi parametar je analizator. Poslednji parametar saopštava *IndexWriteru* da kreira novi indeks (vrednost *true*) ili otvori postojeći (vrednost *false*).

```
//  
// creating index  
//  
indexWriter = new IndexWriter(this.pathIndex, new StandardAnalyzer(), true);  
...  
...  
addFolder(di);
```

Kada je indeks kreiran ili otvoren, spreman je za izmene. Ova aplikacija indeksira sistem fajlova, što znači da se čitaju specificirani folder i podfolderi koje on eventualno sadrži. Kako se iterativno prolazi kroz sistem fajlova, svaki fajl poslovnog dokumenta u formatu DOC, RTF ili PDF otvara se metodom *addFolder()* kao tekst i indeksira.

```
private void addFolder(DirectoryInfo directory)  
{  
    ...  
    // find all matching files  
    //  
    foreach (string pattern in patterns)  
    {  
        foreach (FileInfo fi in directory.GetFiles(pattern))  
        {  
            // skip temporary office files  
            if (fi.Name.StartsWith("~"))  
                continue;  
  
            try  
            {  
                addBusinessDocument(fi.FullName);  
  
                // update statistics  
                this.countTotal++;  
                this.bytesTotal += fi.Length;  
  
                // show added file  
                status(fi.FullName);  
            }  
            ...  
        }  
    }  
}
```

addFolder() otvara fajlove i prosleđuje fajl hendlere metodi *addBusinessDocument()*. Ova metoda konstruiše ono što se u *Lucene.net* terminologiji naziva *Document*.

```
private void addBusinessDocument(string path)
{
    Lucene.Net.Documents.Document doc = new Lucene.Net.Documents.
Document();
    string filename = Path.GetFileName(path);
```

Document se može posmatrati kao dokument koji pored ostalog sadrži i meta-podatke: naslov, autor, ... Za svaki fajl, indeks kreira poseban *Document*.

Dodavanje podataka Lucene dokumentu

Kada je Lucene dokument kreiran mogu mu se dodati podaci. To se radi kreiranjem jednog ili više polja za svako „parče” podatka u procesiranom fajlu. U ovom slučaju kreirana su sledeća polja: *text*, *path*, *title_doc*, *abstract*, *docSubject*, ... , *SCD*, *EHD*.

Kada se kreira polje, tada se specificira i tip polja.

Segment koda kojim se dodaju polja u *Document* ima sledeći oblik:

```
doc.Add(new Lucene.Net.Documents.Field("text", DefaultParser.Extract(path), Lucene.Net.
Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.TOKENIZED));
    doc.Add(new Lucene.Net.Documents.Field("path", path, Lucene.Net.
Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.UN_TOKENIZED));
    doc.Add(new Lucene.Net.Documents.Field("title_doc", filename,
Lucene.Net.Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.UN_TOKENIZED));
    //
    doc.Add(new Lucene.Net.Documents.Field("abstract", strSumm, Lucene.
Net.Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.UN_TOKENIZED));
    //
    doc.Add(new Lucene.Net.Documents.Field("docSubject", docA.
BuiltInDocumentProperties.Subject, Lucene.Net.Documents.Field.Store.YES, Lucene.Net.
Documents.Field.Index.UN_TOKENIZED));
    Aspose.Words.Properties.CustomDocumentProperties props = docA.
CustomDocumentProperties;
    ...
    doc.Add(new Lucene.Net.Documents.Field("SCD", SCDindex(folder + "\\\" + name), Lucene.Net.
Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.UN_TOKENIZED));
    doc.Add(new Lucene.Net.Documents.Field("EHD", EHDindex(folder +
"\\\" + name), Lucene.Net.Documents.Field.Store.YES, Lucene.Net.Documents.Field.Index.UN_
TOKENIZED));
```

Pošto je objekt *Document* popunjen sa objektima *Field*, *Document* može da se doda u indeks:

```
...
// add Document to index
indexWriter.AddDocument(doc);
}
```

Pretraživanje indeksa

Pretraživanje je slično sa indeksiranjem i nudi izuzetnu funkcionalnost. Postoji više načina za pretraživanje indeksa. Lucene.net se može koristiti da pretražuje jedan indeks a može se koristiti i *MultiSearcher* da se pretražuje više indeksa. Pretraživanje dva ili više indeksa distribuira podatke

kroz više indeksa za brže pretraživanje, bolje podešavanje i veću kontrolu. Ova aplikacija koristi samo jedan indeks.

Lucene.net omogućava da se pretraživanje vrši po jednom ili više polja. U slučaju više polja mogu se primenjivati Boolovi kriterijumi upita kao što su AND, OR, NOT, NEAR i DATE RANGE a svako pojedinačno polje može se vrednovati posebnim težinskim faktorom.

Kod ove aplikacije korisniku je omogućeno da bira da li će da vrši pretraživanje samo tekstualnog, samo slikovnog ili oba tipa sadržaja. Takođe, korisnik može preko grafičke kontrole *Slider* da definiše vrednost fazi pondera za postavljeni upit. Aplikacija ima implementirano tzv. preslovljavanje, tj. korisnik može prebacivanjem tastature na Windows TaskBaru na odgovarajuće kodiranje (Serbian Latin/Serbian Cyrillic) da postavi upit latiničnim, odnosno ćiriličnim ispisom.

Sledeći segment koda pokazuje kako se pretražuje indeks samo po tekstualnom sadržaju.

```
private void search()
{
    ...
    ...
    try
    {
        searcher = new IndexSearcher(this.pathIndex);
    }
    catch (IOException ex)
    {
        MessageBox.Show("The index doesn't exist or is damaged. Please rebuild the
index.\r\n\r\nDetails:\r\n" + ex.Message);
        return;
    }

    this.listViewResults.Items.Clear();
    //*****
    //
    // ONLY TEXT
    //
    //*****
    if (radioButton8.Checked == true)
    {

        // Parse the query, "text" is the default field to search
        if (this.textBoxQuery.Text.Trim(new char[] { ' ' }) == String.Empty)
        {
            string searchT = String.Format("You must select/write term!");
            statusSearch(searchT);
            return;
        }
        QueryParser qT = new QueryParser("text", new StandardAnalyzer());
        BooleanQuery boolQuery = new BooleanQuery();
        //
        //
        Query queryT = qT.Parse(this.textBoxQuery.Text + "~" + tfw.
ToString("0.00"));
        //
        keyboardID = LanguageSelector.KeyboardLayout.getName();
        //
        if (keyboardID == "00000809" || keyboardID == "00000409")
        {
            boolQuery.Add(queryT, BooleanClause.Occur.SHOULD);
        }
        else
        {

```

```

        Query queryTalt = qT.Parse(altQueryText + "~" + tfw.ToString("0.00"));

        boolQuery.Add(queryT, BooleanClause.Occur.SHOULD);
        boolQuery.Add(queryTalt, BooleanClause.Occur.SHOULD);
    }
    //
    Hits hits = searcher.Search(boolQuery);
    ...
    ...

```

Iz koda se vidi da *Lucene.net* referencira klase *StandardAnalyzer*, *Document*, *QueryParser*, *Hits*, *IndexSearcher*, *Query* i *Searcher*.

Razumevanje pretraživanja

Searcher je „ulaz” u indeks. Preko njega pretražuje se jedan ili više indeksa lociranih lokalno na hard disku ili na različitim udaljenim mašinama. Sledeći segment koda kreira objekat *Searcher* instanciranjem klase *IndexSearcher*. Parametar prosleđen klasi *IndexSearcher* je ime foldera koji sadrži indeks, izraženo kao puna ili relativna putanja.

```
searcher = new IndexSearcher(this.pathIndex);
```

Raščlanjavanje korisnikovog upita QueryParserom

QueryParser radi ruku-pod-ruku sa analizatorom. Analizator je korišćen kada je kreiran indeks. Zašto nam je opet potreban kod pretraživanja? Za vreme indeksiranja, analizator je korišćen za „čišćenje” sirovog teksta. Ista pravila moraju se primeniti na tekst koji je korisnik upisao kao upit. Pored toga, isti tip analizatora mora da se koristi i za pretraživanje i za indeksiranje, inače rezultati pretraživanja neće biti korektni, ili još gore, neće se dobiti ništa kao rezultat pretraživanja. Zadatak klase *QueryParser* je da prihvati korisnikov upit, primeni ista pravila kao i analizator i „zaključiti” šta korisnik traži.

Na primer, ako je korisnik postavio upit *+pas +kanarinac*, *QueryParser* će znati da korisnik traži obe reči, *pas* i *kanarinac* i da one moraju biti u istom polju.

Opcija *+* označava da je term koji sledi obavezan deo upita.

Lucene.net podržava nekoliko moćnih opcija za pretraživanje. Može se postaviti Boolov upit koristeći operatore *OR*, *AND* i *NOT* a upit se može ograničiti na konkretno polje eksplicitnim navođenjem naziva polja.

U ovoj aplikaciji *QueryParser* je kreiran na sledeći način:

```
QueryParser qT = new QueryParser("text", new StandardAnalyzer());
```

Rad sa rezultatima pretrage

Kolekcija *Hits* je ono što se dobija kao rezultat izvršavanja upita. Ako postavljeni upit vraća „pogotke” (rezultate), koristi se objekat *Hits* da se iterira kroz listu objekata *Documents*.

U aplikaciji referenca na objekat *Hits* vraća se kao:

```
...
Hits hits = searcher.Search(boolQuery);
//
for (int i = 0; i < hits.Length(); i++)
{
    // get the document from index
    Lucene.Net.Documents.Document doc = hits.Doc(i);
    // create a new row with the result data
    string filename = doc.Get("title_doc");
    string path = doc.Get("path");
    string folder = Path.GetDirectoryName(path);
    DirectoryInfo di = new DirectoryInfo(folder);
    //
    di.CreateSubdirectory("Thumbnails");
    ...
    ...
}
```

Kod pretraživanja koristimo objekat *Document* da drži informaciju o „pogotku” koji je dobijen kao rezultat za postavljeni upit. U aplikaciji, referenca na objekat *Document* je realizovana naredbom:

```
Lucene.Net.Documents.Document doc = hits.Doc(i);
```

Objekat *Searcher* je instanciran i usmeren na folder sa indeksom. Zatim se prosleđuje referenca na objekat *Query*. Ovakav način apstrakcije čini *Lucene.net* fleksibilnim i moćnim a rad sa indeksom je konzistentan. Pored toga, pretraga je konzistentna i u slučaju jednog upita ili kombinacije upita.

Aplikacija rezultate pretrage prikazuje korisniku u obliku liste.

```
...
ListViewItem item = new ListViewItem(new string[] {filename,
    folder,
    hits.Score(i).ToString(),
    MPEG7_Dis.ToString(),
    rank,
    TheFile.CreationTime.ToLongDateString(),
    TheFile.LastAccessTime.ToLongDateString(),
    TheFile.LastWriteTime.ToLongDateString(),
    TheFile.Length.ToString()});
//
//
this.listViewResults.Items.Add(item);
...
```

Izvorni kôd aplikacije nalazi se na pratećem CD-ROM-u.

7.2 Eksperimenti i rezultati

Aplikacija je testirana na elektronskim poslovnim dokumentima prikupljenim preko Interneta sa sajtova više sportskih saveza Republike Srbije, poslovnih banaka i firmi koje se bave trgovinom.

Dokumenti su izmešani i smešteni u tri foldera sa različitim brojem dokumenata. Eksperimenti su obavljani na personalnom računaru sa procesorom Sempron na 1.6 GHz i radnom memorijom od 768 MB. Za indeksiranje sadržaja tih foldera dobijena su vremena indeksiranja koja su data u tabeli 7.1.

Tabela 7.1 Vremena indeksiranja u zavisnosti od broja dokumenata

Broj dokumenata	20	50	100
Prosečno vreme indeksiranje	18 sekundi	44 sekunde	98 sekundi

Testirana su vremena pretraživanja u zavisnosti od broja dokumenata u folderu, kao i od vrednosti primenjenog fazi pondera na upit. Zadat je kombinovani upit koji se sastojao od latiničnog terma „košarka” i logo znaka *Košarkaškog saveza Srbije*. Dobijeni rezultati prikazani su tabelarno.

Fazi ponder tekst 0,99 Fazi ponder slike 0,99	
Broj dokumenata	Vreme pretraživanja u sekundama
20	1,1
50	1,11
100	1,12

Fazi ponder tekst 0,5 Fazi ponder slike 0,99	
Broj dokumenata	Vreme pretraživanja u sekundama
20	5
50	25
100	34

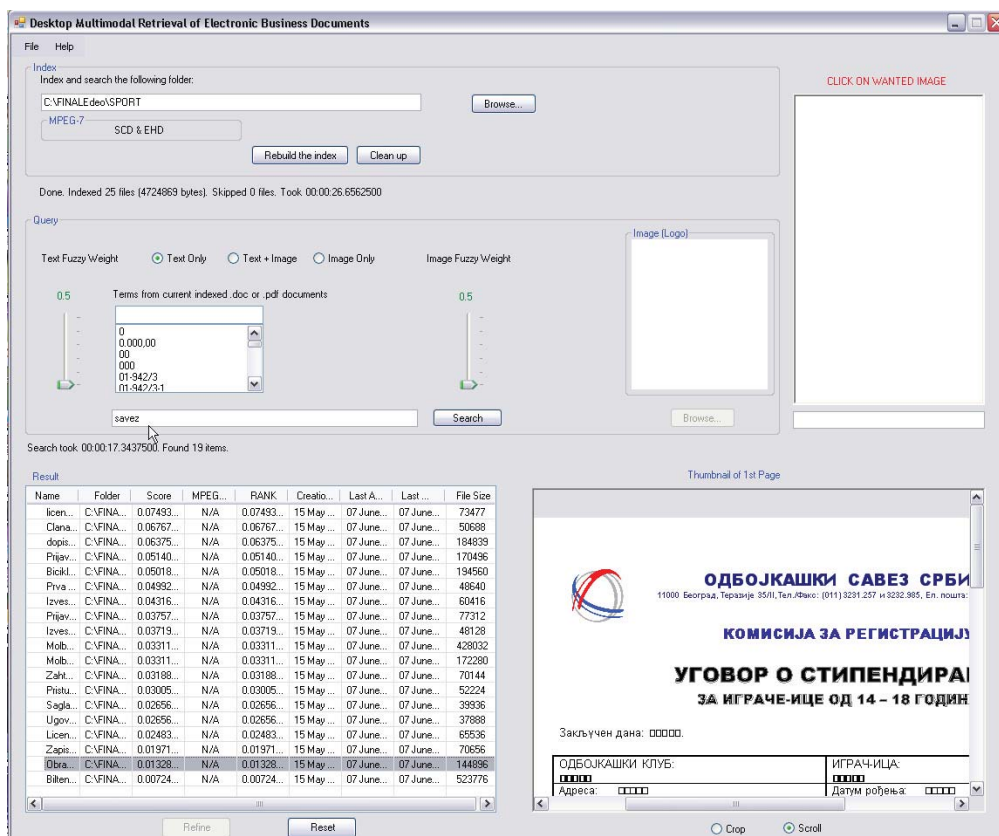
Fazi ponder tekst 0,5 Fazi ponder slike 0,5	
Broj dokumenata	Vreme pretraživanja
100	> veće od 5 minuta

Uvidom u rezultate može se zaključiti da broj dokumenata koji se pretražuje nije kritičan ako su fazi ponderi upita visoki (mali stepen tolerancije). Međutim, kod relativno niskih vrednosti fazi pondera upita (veliki stepen tolerancije) vreme pretraživanja puno zavisi od broja dokumenata koji treba da se pretraže.

Izvršeni su brojni eksperimenti pronalaženja dokumenata a ovde se navode karakteristična zapažanja.

Primer 1:

Za primer samo tekstualnog upita sa jednim termom „savez” u latiničnom ispisu (tastatura postavljena na *Serbian Latin*) i fazi ponderom vrednosti 0,5 aplikacija je dala rezultate prikazane na slici 7.7.



Slika 7.7 Tekstualni upit postavljen sa velikim stepenom tolerancije

Vizuelnom inspekcijom rezultata utvrđeno je da je aplikacija vratila sve dokumente iz indeksiranog foldera koji sadrže navedeni term. U rezultat su uključeni i ćirilični i latinični dokumenti, ukupno 19 dokumenata.

Primer 2:

Ponovljen je prethodni upit, samo sa fazi ponderom vrednosti 0,99. U rezultat je uključeno 16 dokumenata. Sva rangirana dokumenta su latinična.

Primer 3:

Postavljen je ponovo isti tekstualni upit sa ćiriličnim ispisom (tastatura postavljena na *Serbian Cyrillic*) i fazi ponderom vrednosti 0,5. Rezultat je isti kao kod upita sa latiničnim ispisom i istom vrednošću fazi pondera.

Primer 4:

Četvrta varijanta ovog eksperimenta je sa istim tekstualnim upitom sa ćiriličnim ispisom i fazi ponderom vrednosti 0,99. Aplikacija je u rezultat uključila samo dokumenta sa ćiriličnim ispisom.

Primer 5:

Sledeća varijanta eksperimenta je sa latiničnim ispisom istog terma ali sa engleskom tastaturom. Fazi ponder je prvo postavljen na vrednost 0,5 a zatim na 0,99. U oba slučaja rezultat je bio isti, vraćeno je 6 latiničnih dokumenata.

Dobijeni rezultati se mogu objasniti time da aplikacija za indeksiranje koristi standardni analizator biblioteke Lucene.net koji korektno analizira oblike reči engleskog jezika ali ne i srpske reči. Druga specifičnost je latinično/ćirilično kodiranje srpskih slova na Windows platformi u okviru aplikacija za obradu tekstova.

Primer 6:

Izvršen je eksperiment sa tekstualnim upitom „košarka” (tastatura postavljena na *Serbian Latin*) i fazi ponderom vrednosti 0,99. Rezultat postavljenog upita je obuhvatio 24 dokumenta. Po rangui se izdvaja jedan dokument i taj dokument je zaista relevantan. Međutim, vizuelnom inspekcijom je ustanovljeno da se tek na osmom mestu nalazi sledeći dokument koji je takođe relevantan za postavljeni upit.

Primer 7:

U sledećem eksperimentu proširen je prethodni upit uključivanjem logoa *Košarkaškog saveza Srbije* sa fazi ponderom za sliku vrednosti 0,99. U rezultatu se na prva četiri mesta nalaze dokumenti koji potiču iz tog Saveza, što znači da su relevantni za postavljeni upit.

U ovom slučaju se vidi doprinos uključivanja slike u upit u cilju dobijanja relevantnih dokumenata za postavljeni upit.

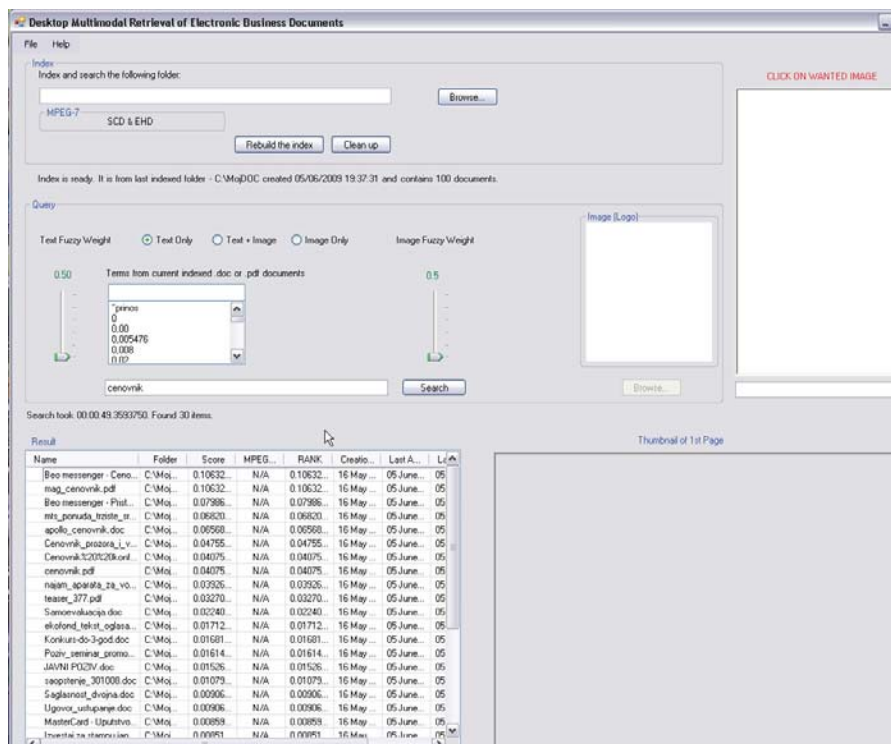
U nizu eksperimenata uočen je jedan eksperiment sa interesantnim rezultatima. Pregledanjem prikupljenih dokumenata *Rukometnog saveza Srbije* uočeno je da su na nekim dokumentima korišćeni crno-beli logo znaci a na nekima u boji, koji nisu istih nijansi.

Primer 8:

Izveden je eksperiment samo sa slikovnim upitom. Uzet je crno-beli logo *Rukometnog saveza Srbije* sa fazi ponderom vrednosti 0,99. U vraćenim rezultatima prva dva mesta zauzela su dva dokumenta sa crno-belim znakom a sledeća tri dokumenta su sa znakom u boji. Posle njih slede dokumenti koji nisu relevantni za upit.

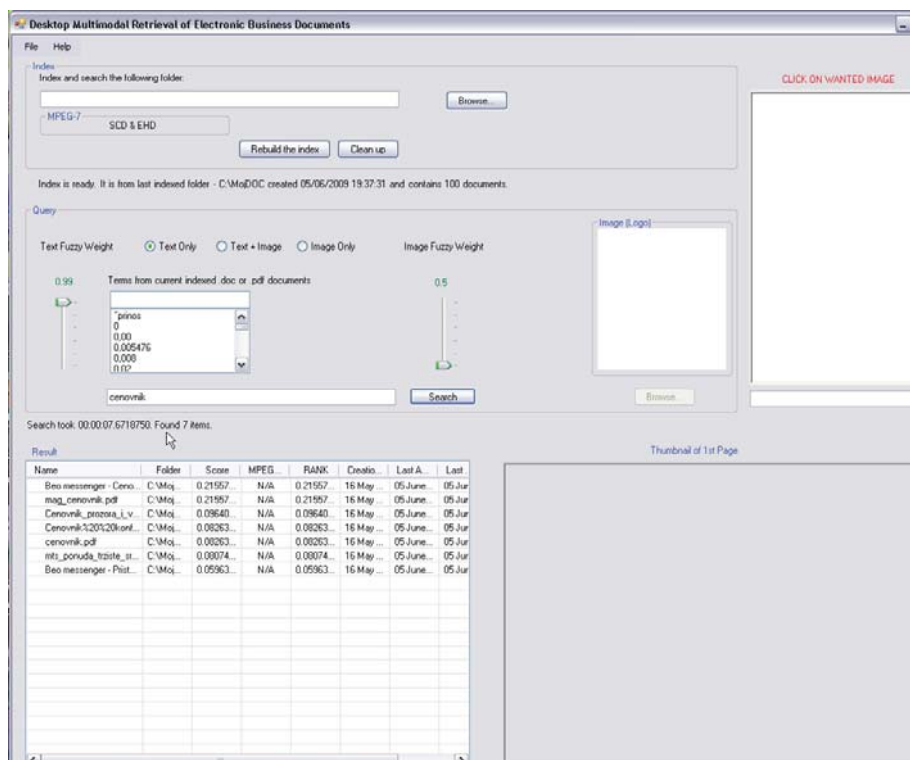
Primer 9:

Eksperiment je izvršen na kolekciji dokumenata (100 dokumenata) koja je sadržavala pored sportskih dokumenata i dokumente komercijalnih firmi i banaka. Prvo je postavljen samo tekstualni upit sa termom „cenovnik” i fazi ponderom vrednosti 0,5. U rezultat je bilo uključeno 30 dokumenata.



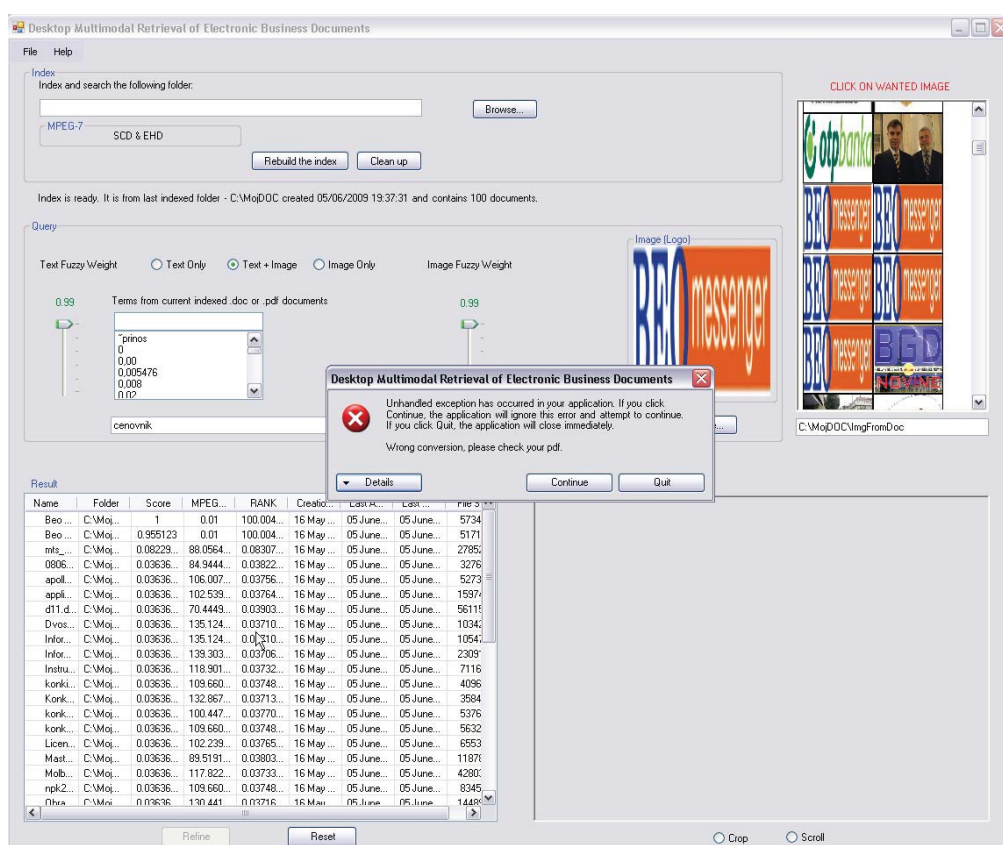
Slika 7.8 Tekstualni upit sa velikim stepenom tolerancije

Zatim je promenjena vrednost fazi pondera na 0,99 i rezultat je obuhvatio 7 dokumenata.



Slika 7.9 Tekstualni upit sa malim stepenom tolerancije

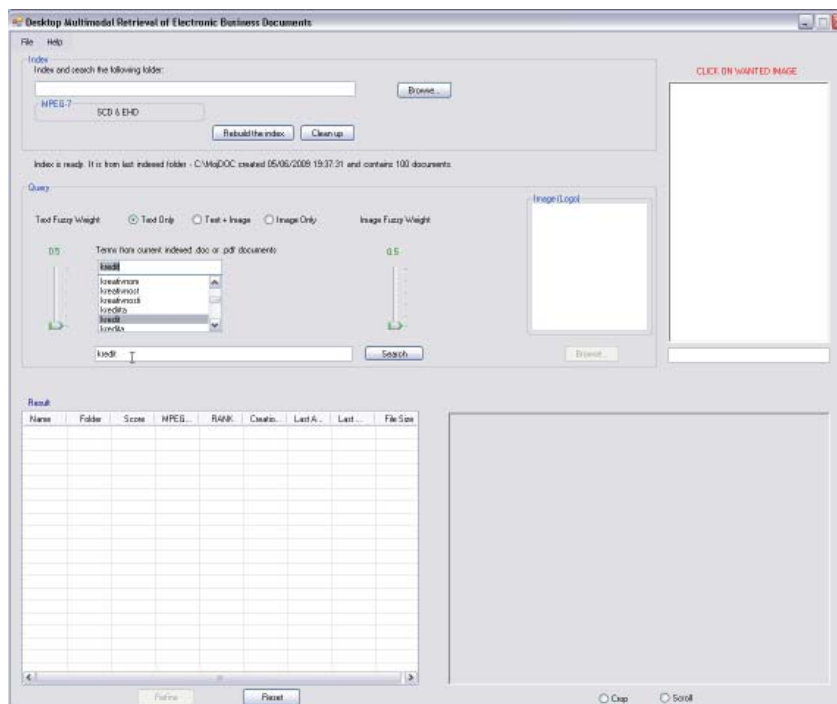
Usledio je eksperiment sa kombinovanim upitom koji se sastojao od istog tekstualnog terma sa fazi ponderom 0,99 i slikovnog upita (logo znak firme *Beo Messenger*) sa fazi ponderom vrednosti 0,99. Vreme pretraživanja je premašilo 4 minuta i došlo je do kraha aplikacije zbog problema u pristupu PDF dokumentu prilikom izlistavanja rezultata pretrage, tj. zbog problema sa memorijom.



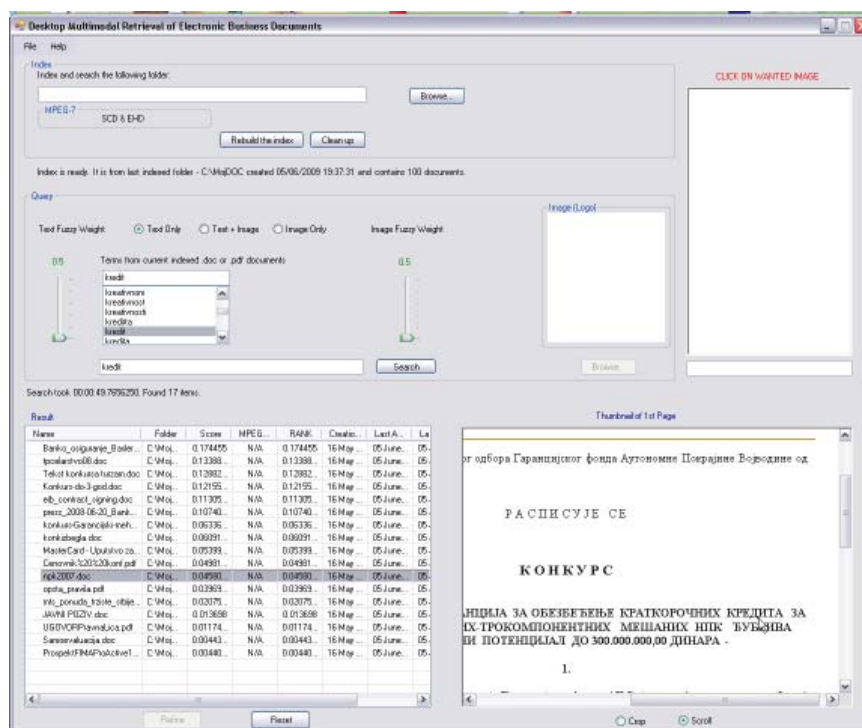
Slika 7.10 Kod kombinovanog upita sa malim stepenom tolerancije na nešto većoj kolekciji dokumenata aplikacija je nestabilna

Primer 10:

Postavljen je tekstualni upit sa termom „kredit” koji je izabran iz padajuće liste ekstrahovanih reči iz kolekcije dokumenata i fazi ponderom vrednosti 0,5. Vreme pretraživanja na kolekciji od 100 dokumenata bilo je 50 sekundi a rezultat je obuhvatio 17 dokumenata.



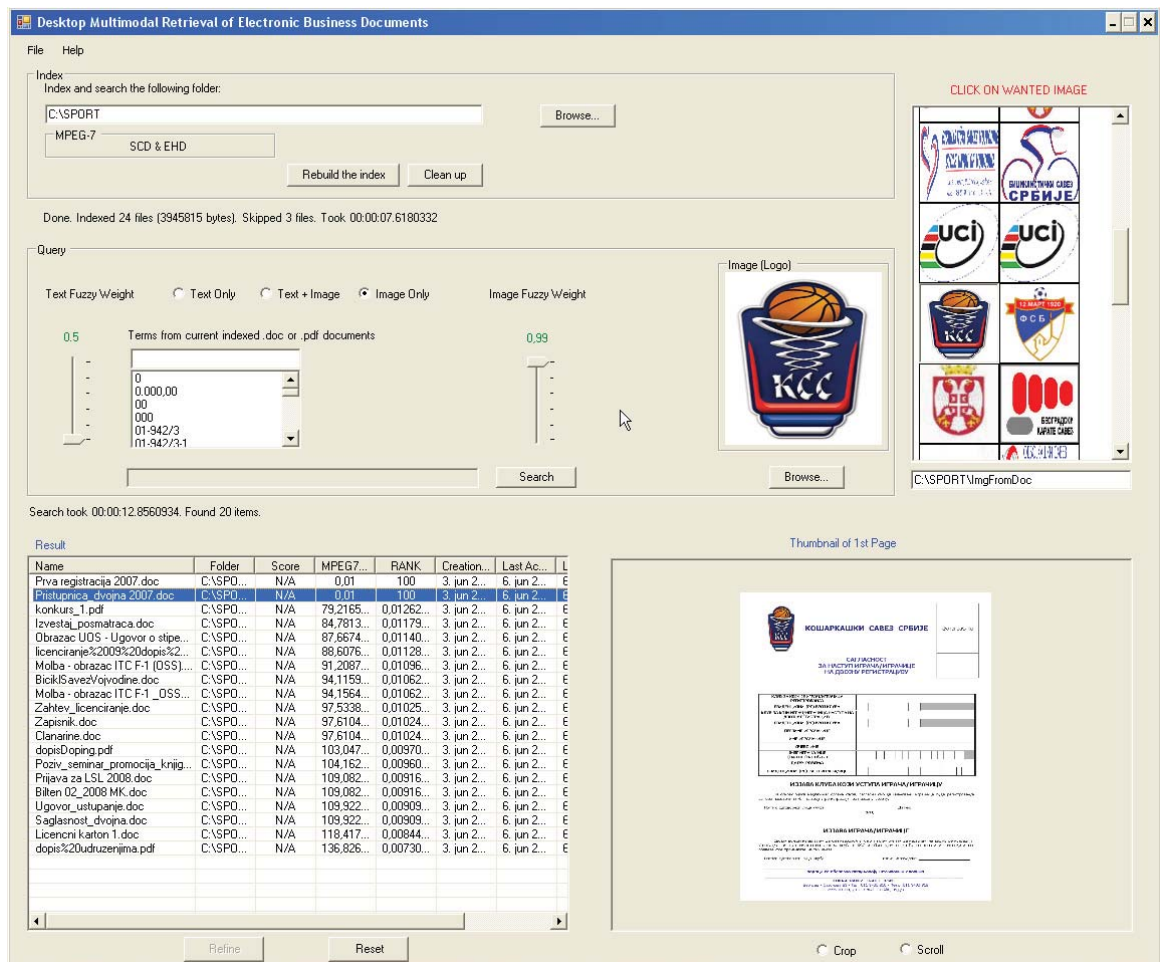
Slika 7.11 Korišćenje za upit tekstualnog terma iz rečnika koji je program izdvojio iz kolekcije dokumenata



Slika 7.12 Prikaz rezultata prethodnog upita

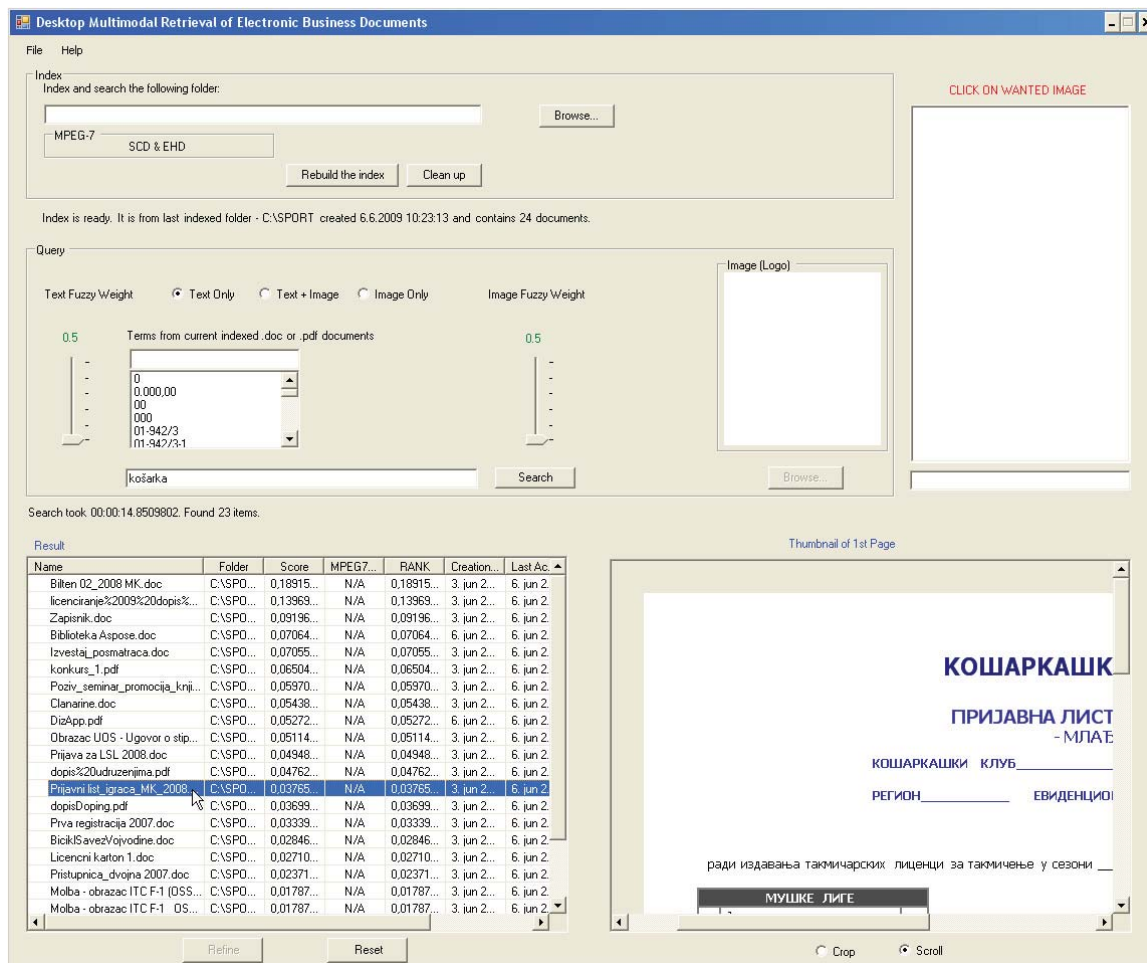
Primer 11:

Za potrebe ovog eksperimenta iz dokumenta Prijavni list_igraca_MK_2008.doc *Košarkaškog saveza Srbije* uklonjen je logo znak i dokument snimljen sa novim stanjem pod istim imenom. Postavljen je samo slikovni upit sa logo znakom *Košarkaškog saveza Srbije* i postavljen fazi ponder na vrednost 0,99. Aplikacija je vratila rezultate u koje nije bio uključen taj dokument.



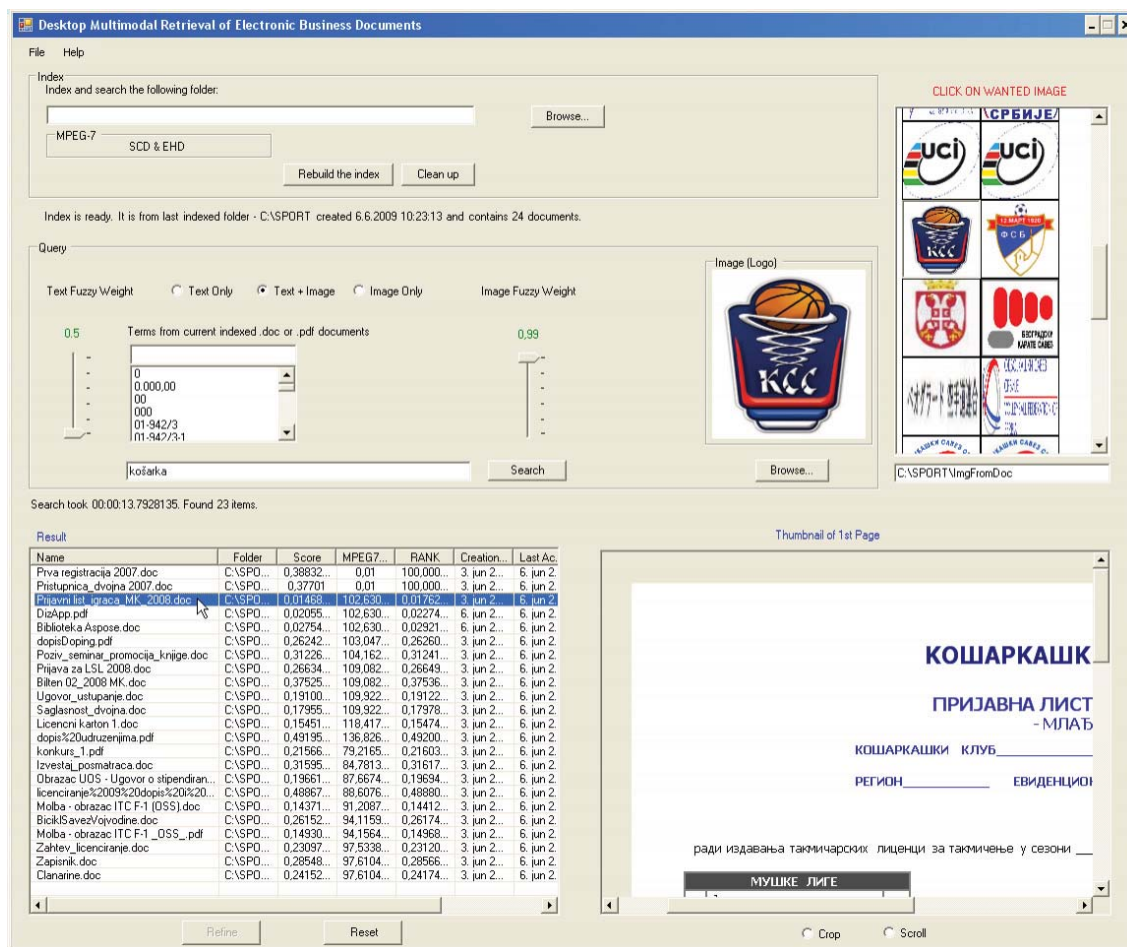
Slika 7.13 Kod samo slikovnog upita program neće uključiti u rezultate dokumente koji nemaju traženi logo

Zatim je postavljen samo tekstualni upit sa termom „košarka” i postavljena vrednost fazi pondera na 0,5. U rezultate je uključen pomenuti dokument ali je po rangui 13-ti.



Slika 7.14 Kod postavljenog samo tekstualnog upita program u rezultate uključuje dokumente koji sadrže postavljene term prema definisanom stepenu tolerancije

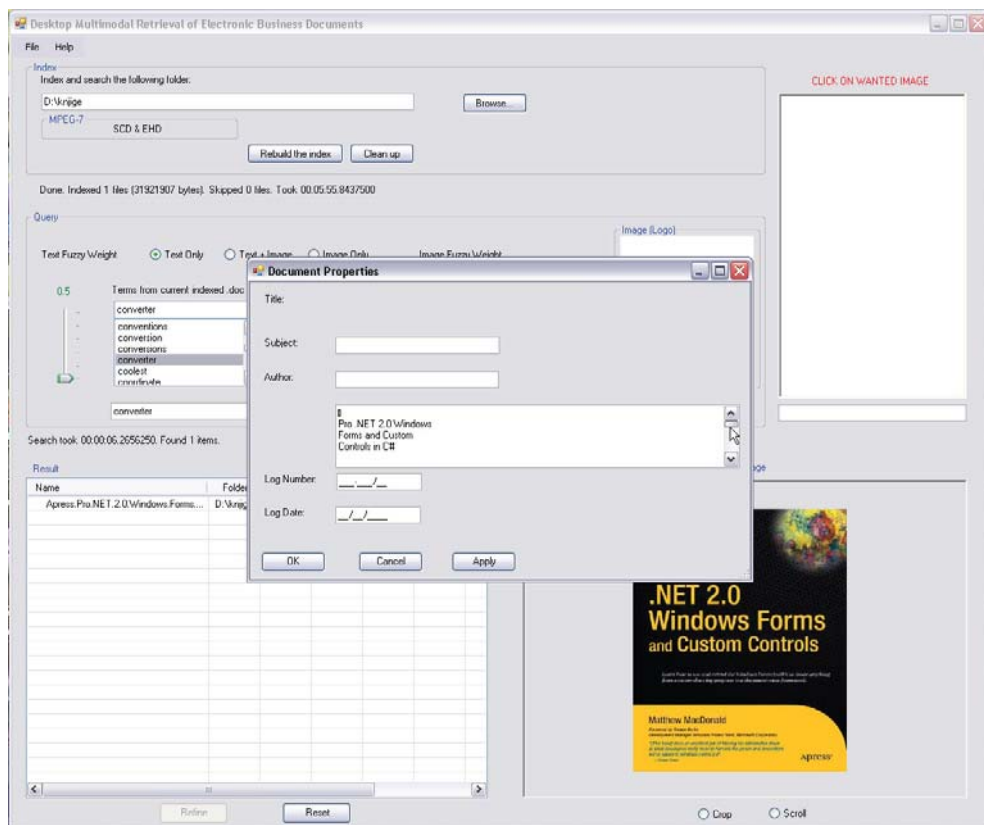
Na kraju, postavljen je kombinovani upit koji se sastojao od tekstualnog terma „košarka” sa fazi ponderom vrednosti 0,5 i logo znakom *Košarkaškog saveza Srbije* sa fazi ponderom 0,99. Rezultat je obuhvatio ranije pomenuti dokument bez logo znaka i rangirao ga na 3-će mesto, odmah posle dva dokumenta pomenutog Saveza koja imaju ugrađene logo znake. Rezultat se može objasniti činjenicom da aplikacija dokumentu koji nema slike kod indeksiranja deskriptorima slike SCD i EHD dodeljuje vrednost 0, tako da kada se izračunava rang dokumenta deo koji se odnosi na tzv. MPEG7 distancu nema uticaj na pogoršanje ranga, dok dokumenti koji imaju slike koje se razlikuju od zadate zbog uticaja MPEG7 distance pogoršavaju svoj rang bez obzira na to što sadrže zadati tekstualni term ili sličan tekstualni term.



Slika 7.15 Kombinovani upit je u rezultate uključio sve relevantne dokumente

Primer 12:

Program je testiran i na jednoj knjizi u PDF formatu obima 1000 strana. Knjiga je smeštena u poseban folder i aplikacija je indeksirala samo taj dokument. Ograničena je da ekstrahuje samo sliku prve stranice knjige, tj. naslovnu stranicu. Vreme indeksiranja je bilo preko 5 minuta a zadati teksutalni term je pronađen za 6 sekundi.



Slika 7.16 Aplikacija zbog performansi nije pogodna za rad sa PDF dokumentima velikog obima

Na osnovu pokazanih vremena evidentno se može zaključiti da aplikacija nije pogodna za rad sa velikim PDF dokumentima na slabim hardverskim platformama (Sempron 1,6 GHz).

Prilagođena verzija aplikacije biće korišćena u jednom edukacionom modelu za košarkaške sudije koji je tema jednog magistarskog rada na Elektrotehničkom fakultetu u Beogradu.

8. Zaključak

Aplikacija je na praktičnim primerima potvrdila osnovnu hipotezu da uključivanje u upit modaliteta slike doprinosi boljem i preciznijem pronalaženju relevantnih dokumenata. Upotrebljiva je na manjim kolekcijama poslovnih dokumenata manjeg obima. Zbog invazivnog delovanja na dokumente, ekstrakcija slika, potrebno ju je pažljivo optimizovati u smislu korišćenja memorije računara i povećanja pouzdanosti u radu. Korisno bi bilo uključiti u aplikaciju i kompaktan deskriptor oblika kod slika čime bi se svakako poboljšali rezultati pronalaženja. Takođe, trebalo bi istražiti odnose između pondera tekstualnog i slikovnog sadržaja kod određivanja ranga dokumenta prema postavljenom upitu.

Za primenu u digitalnim arhivama potrebno bi bilo razviti modul za klasifikaciju dokumenata, kao i inteligentni modul za automatsko rezimiranje teksta.

Aplikacija bi se uz određena prilagođenja mogla koristiti kao klijentski deo veb servisa za pronalaženje informacija u poslovnim dokumentima.

Dalji pravci istraživanja mogli bi biti uključivanje modula za pronalaženje slika na osnovu tekstualnog opisa, kao i implementacija govornog interfejsa za srpski jezik za postavljanje upita koji se odnose na tekstualni sadržaj dokumenata.

Prave mogućnosti programa bi se mogle sagledati uključivanjem lingvističkih resursa za srpski jezik, što bi se moglo realizovati uspostavljanjem saradnje sa Grupom za jezičke tehnologije Univerziteta u Beogradu.

9. Literatura

- [1] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: Fulfilling the Memex vision. In *Proceedings of ACM Multimedia'02*, pages 235–238, 2002.
- [2] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. In *Proceedings of WWW'03*, pages 366–375. ACM Press, 2003.
- [3] S. Dumais, E. Cutrell, JJ Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In *Proceedings of SIGIR'03*, pages 72–79. ACM Press, 2003.
- [4] W. Nejdl and R. Paiu. Desktop search - how contextual information influences search results and rankings. In *Proceedings of the IRIx Workshop at SIGIR'05*, 2005.
- [5] Google desktop. <http://desktop.google.com/features.html#search>.
- [6] Copernic desktop search. www.copernic.com/.
- [7] Spotlight. <http://www.apple.com/macosx/features/spotlight/>.
- [8] M. Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, 1988.
- [9] The Lowell database research self assessment. Attendees at Lowell Workshop, May 2003.
- [10] David McG. Squire, Wolfgang Müller, Henning Müller, Thierry Pun, “Content-based query of image databases: inspirations from text retrieval,” *Pattern Recognition Letters* 21:1193-1198, 2000.
- [11] R.R. Korfhage. *Information Storage and Retrieval*. pp. 84-85. John Wiley & Sons, Inc. 1997.

- [12] Richardo Baeza-Yates & Berthier Ribeiro-Neto. Modern Information Retrieval. Glossary. Addison Wesley Longman Publishing Co. Inc. May, 1999. (web: <http://www.sims.berkeley.edu/~hearsst/irbook/glossary.html>)
- [13] Jamie Callan. IR class notes. 2000.
- [14] R. B. Yates and B. R. Neto. Modern information retrieval. Addison-Wesley/ACM Press, 1999.
- [15] M. Swain, C. Frankel, and V. Athitsos. Webseer: an image search engine for the world wide web. CVPR, 1997.
- [16] R. Zhao and W. I. Grosky. Narrowing the semantic gap-improved text-based web document retrieval using visual features. IEEE Transactions on Multimedia, 4(3):189-200, June 2002.
- [17] S. Mukherjea and J. Chost. Automatically determining semantics for World Wide Web multimedia information retrieval. Journal of Visual Languages and Computing (1999) 10, 585-606.
- [18] S. Sclaroff, M. La Cascia, and S. Sethi. Unifying textual and visual cues for content-based image retrieval on the world wide web. Comp. Vis. IU, 75:86-98, 1999.
- [19] D. Doermann. The indexing and retrieval of document images: a survey. Computer vision and image understanding, 70(3): 287-298. 1998.
- [20] M. Mitra and B.B. Chaudhuri. Information retrieval from documents: a survey. Information retrieval, vol. 2, nos. 2/3, pp.141-163, 2000.
- [21] D. P. Lopresti. Robust retrieval of noisy text. In Proc. of ADL'96, pp. 76-85, 1996.
- [22] K. Taghva, J. Borsack, A. Condit, and S. Erva. The effects of noisy data on text retrieval. Journal of the American Society for Information Science, 45:50-58, 1994.
- [23] A. F. Smeaton and A. L. Spitz. Using character shape coding for information retrieval. In Proc. of 4th ICDAR, pp. 974-978, 1997.
- [24] Y. Lu, L. Zhang, C.L. Tan. Retrieving imaged documents in digital libraries based on word image coding. In Proc. of the first international workshop on Document Image Analysis for Libraries (DIAL'04).
- [25] Y. Rui and T. S. Huang. Image retrieval: current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation 10,39-62(1999).
- [26] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, J. Malik. Blobworld: a system for region-based image indexing and retrieval. In Proc. of the Third International Conference on Visual Information and Information Systems (VISUAL'99), Appears in Lecture Notes in Computer Science, Vol. 1614, 1999, pp. 509-516.

- [27] F. Cesarini, S. Marinai, and G. Soda. Retrieval by layout similarity of documents represented with MXY. Document Analysis Systems V (S. V.-L. 2423, ed.), pp. 353–364, 2002.
- [28] Nagy and S. Seth. Hierarchical representation of optically scanned documents. In Proceedings of the 7th international conference on pattern recognition, Los Alamitos, California, USA, 1984, pp 347-349.
- [29] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620.1975.
- [30] C. J. van Rijsbergen. Information Retrieval, on-line book. 1979.
- [31] F. Crestani, M. Lalmas, C. J. van Rijsbergen, and I. Campbell. Is this document relevant? ... Probably: a survey of probabilistic models in information retrieval. ACM Computing Surveys 30(4): 528–552. 1998.
- [32] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In Proc. of SIGIR 21. 1998.
- [33] A. Berger and J. Lafferty. Information retrieval as statistical translation. *SIGIR 22*, pp. 222–229. 1999.
- [34] S. Antani, R. Kasturi, R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern Recognition, 35(2002), 945-965.
- [35] N. Vasconcelos and A. Lippman. A probabilistic architecture for content-based image retrieval. In Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2000), IEEE Computer Society, Hilton Head Island, South Carolina, USA, 2000, pp. 216-221.
- [36] N. Vasconcelos. Bayesian models for visual information retrieval. PhD thesis, Massachusetts Institute of Technology, 2000.
- [37] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. In Michael Lew, editor, Principles of Visual Information Retrieval, pages 87–119. Springer, 2001. ISBN 1-85233-381-2.
- [38] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349-1380, Dec. 2000.
- [39] M. Ortega, K. Porkaew and S. Mehrotra. Information retrieval over multimedia documents. In Proc. of 1999 SIGIR post-conference workshop on Multimedia Indexing and Retrieval, Berkeley, CA, August,1999.
- [40] C. Meghini, F. Sebastiani, U. Straccia. A model of multimedia information retrieval. Journal of ACM, 48(5):909-970, 2001.
- [41] S. Paek and J.R. Smith. Detecting image purpose in World Wide Web documents. Trans. Patt. Analys. Mach. Intell. 22, 12, 1349–1380. 1998.
- [42] S. Chang, J. Smith, M. Beigi & A. Benitez. Visual information retrieval from large distributed online repositories. Communications of ACM 40, 63-71.

- [43] A. Bruce, V. Chalana, M.Y. Jaisimha, and T. Nguyen. The DocBrowse system for information retrieval from document image data. In Proc. Symposium on Document Image Understanding Technology, Annapolis, MD, 181-192.1997.
- [44] R. K. Srihari, R. Chopra, D. Burhans, M. Venkataraman, and V. Govindaraju. Use of collateral text in image interpretation. In Proc. of ARPA Image Understanding Workshop 1994.
- [45] N.C. Rowe. Retrieving captioned pictures using statistical correlations and a theory of caption-picture co-reference. In Symposium on Document Analysis and Information Retrieval, pages 525-534, 1995.
- [46] N.C. Rowe. Precise and efficient retrieval of captioned images: the MARIE project. Library Trends, Fall 1999.
- [47] X. S. Zhou and T. S. Huang. Unifying keywords and visual contents in image retrieval. IEEE Multimedia, 4(2):23-33, June 2002.
- [48] Z. Chen, W.Y. Liu, F. Zhang, M.J. Li and H.J. Zhang. Web mining for web image retrieval. Journal of the American Society for Information Science and Technology, 52(10), (2001), 831--839.
- [49] R.K. Srihari. Intelligent indexing and semantic retrieval of multimodal documents. Information Retrieval, 2, 245-275, 2000.
- [50] S. Santini. Multimodal search in collections of images and text. Journal of Electronic Imaging, Volume 11, Issue 4, pp. 455-468, 2002.
- [51] A. Borgida. Description logics in data management. IEEE Transactions on Data and Knowledge Engineering, 7:671-682,1995.
- [52] X.J. Wang, W.Y.Ma, G.R.Xue, X.Li. Multi-modal similarity propagation and its application for web image retrieval. MM'04, Oct. 10-16, NewYork, NY, USA.
- [53] J. Kandola, J. Shawe-Taylor, N. Cristianini. Learning semantic similarity. NIPS (2002).
- [54] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, Volume 41, Issue 6, 1990.
- [55] T. Westerveld. Image retrieval: Content versus context. In Proc. of Computer-Assisted Information Retrieval, Vol. 1, Paris, France, 2000, pp. 276-284.
- [56] D. M. Squire, W. MTuller, H. MTuller, T. Pun. Content-based query of image databases: inspirations from text retrieval. Pattern Recognition Letters (Selected Papers from The 11th-Scandinavian Conference on Image Analysis SCIA '99) 21 (13-14) (2000) 1193-1198.
- [57] G.A. Miller. WordNet: a lexical database for English. Comm. of the ACM, Vol. 38, No. 11, pp. 39-41, Nov.1995.
- [58] E.M. Voorhees. Query expansion using lexical-semantic relations. In Proc. of the 17th International Conference on Research and Development in Information Retrieval SIGIR'94 Dublin, Ireland. 1994.

- [59] G. Salton. Automatic text processing. Addison-Wesley, 1989.
- [60] V. E. Ogle and M. Stonebraker. Chabot: retrieval from a relational database of images. *IEEE Computer*, Vol. 28, No. 9 (Sept. 1, 1995): 40-48.
- [61] G. Duffing. Text-image interaction for image retrieval and semi-automatic indexing. In *Proc. of the 20th Annual BCS-IRSG Colloquium on IR*, 1998.
- [62] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288-297, 1990.
- [63] M. Worring, A. W. M. Smeulders, S. Santini. Interaction in content-based image retrieval: An evaluation of the state of the art. In: R. Laurini Ed.), *Fourth International Conference On Visual Information Systems (VISUAL'2000)*, no. 1929 in *Lecture Notes in Computer Science*, Springer-Verlag, Lyon, France, 2000, pp. 26-36.
- [64] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems* 8: 536–544 (2003).
- [65] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System -- Experiments in Automatic Document Processing*, pp. 313-323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.
- [66] Y. Lu, H. Zhang, W.Y. Liu and C. Hu. Joint semantics and feature based image retrieval using relevance feedback. *IEEE Transactions on Multimedia*, 5(3):339-347, September 2003.
- [67] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1): 1-47. 2002.
- [68] N. Chen and D. Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. Published by the *International Journal of Document Analysis and Recognition*. 2004.
- [69] R.W. Picard and T. P. Minka. Vision texture for annotation. *Multimedia Systems*, 3(1):3–14, 1995.
- [70] P. Willet. Recent trends in hierarchical document clustering: A critical review. *Information Processing and Management*, 24:577-597, 1988.
- [71] Paek, C. Sable, V. Hatzivassiloglou, A. Jaimes, B. Schiman, S. F. Chang, and K. McKeown. Integration of visual and text-based approaches for the content labeling and classification of photographs. In *Proceedings of the ACM SIGIR Workshop on Multimedia Indexing and Retrieval (SIGIR-99)*.
- [72] C. Lu, M. Drew. Construction of a hierarchical classifier schema using a combination of text-based and image-based approaches. In *Proc. of SIGIR'01*, September 9-12, Louisiana, USA.
- [73] F. Chen, U. Gargi, L. Niles, and H. Schutz. Multi-modal browsing of images in web documents. In *Proc. of SPIE Vol. 3651*, p. 122-133. *Document Recognition and Retrieval VI*, Daniel P. Lopresti; Jiangying Zhou; Eds. 1999.

- [74] D.R. Cutting, D.R. Karger, J.O. Pedersen and J.W. Tukey. Scatter/Gather: a cluster-based approach to browsing large document collections. In Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), 318-329, 1992.
- [75] A.B. Benitez and S.F. Chang. Semantic knowledge constructions from annotated image collections. In Proceedings of the 2002 International Conference On Multimedia & Expo (ICME-02), Lausanne, Switzerland.
- [76] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. In ICCV, volume 2, pp. 408-415. IEEE Computer Society, 2001.
- [77] T. Hofmann. Learning and representing topic, a hierarchical mixture model for word occurrence in document databases. Workshop on learning from text and the web. 1998.
- [78] Bradford, S.C. "Sources of Information on Specific Subjects". Engineering: An Illustrated Weekly Journal (London), 137, 1934 (26 January), pages 85-86.
- [79] Saracevic, T. (1975), "Relevance: a review of and a framework for the thinking on the notion in information science", Journal of the American Society for Information Science, Vol. 26 No. 6, pp. 321-43.
- [80] S. C, Bradford „Sources of information on specific subjects”, Engineering, 137, 26 Jan. 1934, 85-86; Documentation, 1948.
- [81] Lotka, A. J. 1926. Statistics - the frequency distribution of scientific productivity. Journal of the Washington Academy of Science, 16: 317- 25.
- [82] George K. Zipf (1949) Human Behavior and the Principle of Least-Effort. Addison-Wesley.
- [83] Derek J. de Solla Price (1970), "Citation Measures of Hard Science, Soft Science, Technology, and Nonscience", in Nelson, C. E. & Pollock, D.K. (eds.), Communication among Scientists and Engineers, Lexington, MA: D.C. Heath and Company, pp. 3-22.
- [84] Urquhart, D.J. (1959a), "Use of scientific periodicals", Proceedings of the International Conference on Scientific Information, Washington, DC, November 16-21, 1958, National Academy of Sciences, National Research Council, Washington, DC, Vol. 1 pp.287-300.
- [85] M. Konchady. Building Search Applications, Mustru Publishing, Oakton, Virginia, 2008.
- [86] I. Obradović, R. Stanković Softverski alati za korišćenje jezičkih resursa za srpski jezik, INFOTEKA, Broj 1-1 god. IX, maj 2008.
- [87] S.M. Beitzel, E.C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. Proceedings of Sheffield SIGIR 2004. The Twenty-Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 321–8, 2004.

- [88] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing and Management*, 43(4):866–886, 2007.
- [89] Yonggang Qiu and H.P. Frei. Concept based query expansion. *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169, 1993.
- [90] R.R. Joshi and Y.A. Aslandogan. Concept-based web search using domain prediction and parallel query expansion. *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration (IEEE Cat. No.06EX1467)*, page 6 pp., 2007.
- [91] Olga Vechtomova, Ying Wang. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4):324–333, 2006.
- [92] Lucene search engine library, Link: <http://jakarta.apache.org/lucene>
- [93] Lucene in Action by Erik Hatcher, Manning Publications (December 31, 2004),
- [94] Egothor search engine library, Link: <http://www.egothor.org/>
- [95] Xapian, an Open Source Probabilistic Information Retrieval library, Link: <http://www.xapian.org/>
- [96] PDFBox, Java PDF processing Library, Link: <http://www.pdfbox.org/>
- [97] Jakarta POI, Java API to Access Microsoft Format Files, Link: <http://jakarta.apache.org/poi/>
- [98] Aspose PTY LTD, Aspose.Words, <http://www.aspose.com>
- [99] Aspose PTY LTD, Aspose.Pdf.Kit, <http://www.aspose.com>
- [100] John Miano. *Compressed Image File Formats, JPEG, PNG, GIF, XBM, BMP*. Addison-Wesley, 1993.
- [101] E. M. Rasmussen. 1997. Indexing images. *Annual review of information science and technology*. vol. 32, pp 169-196
- [102] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic and Will Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3, 3/4, July 1994, pp. 231-262.
- [103] Neil A. Stillings et al. *Cognitive Science, An Introduction*. 2nd Edition. The MIT Press, Cambridge, Massachusetts, 1995.
- [104] Y. Rui, T.S.Huang, and S.F. Chang. Image Retrieval: Current techniques, promising directions and open issues, *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39-62, March 1999.
- [105] J.P. Eakins. Automatic image content retrieval ?are we getting anywhere? *Proceedings of Third International Conference on Electronic Library and Visual Information Research (ELVIRA3)*, pp. 123-135, De Montfort University, Milton Keynes, May 1996.

- [106] J.P. Eakins. Techniques for image retrieval. Library and Information Briefings, 85. London: South Bank University, Library Information Technology Centre, 1998
- [107] Deselaers, D. Keysers, H. Ney: Features for image retrieval: an experimental comparison, [Information Retrieval](#), Springer Netherlands, 1386-4564, [Volume 11, Number 2 / April, 2008](#), p. 77-107
- [108] Michael Swain. and Dana Ballard. Color indexing. Int. J. Comput. Vis. 7(1):11-32, 1991.
- [109] Gong Y, Chuan C.H, Xiaoyi G. *Image indexing and and retrieval using color histograms*, Multimedia Tools and Applications, vol.2 pp. 133-156, 1996
- [110] Shengjiu Wang, *A Robust CBIR Approach Using Local Color Histograms*, Technical Report TR 01-03, Departement of computing science, University of Alberta, Canada. October 2001.
- [111] Lu, G. J. Phillips and S. Rahman. *Techinques to Improve Color Based Image Retrieval Performance*. Proceedings of International Symposium on Audio, Video, Image Processing and Intelligent Applications, Baden Baden, Germany. pp 57-61. August 17-21 1998.
- [112] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. *Query by image and video content the qbic system*. *IEEE Computer*, 28(9):23–32, 1995.
- [113] Jeff Berens. *Image Indexing using Compressed Colour Histograms*. Thesis submitted for the Degree of Doctor of Philosophy in the School of information Systems, University of East Anglia, Norwich.
- [114] J.R. Smith and S.F. Chang, “Transform features for texture classification and discrimination in large image databases,” Proc. of IEEE Intl. Conf. on Image Processing, 3:407-411, 1994.
- [115] Deselaers T, Weyand T, Keysers D, Macherey W, Ney H. FIRE in ImageCLEF 2005: Combining Contentbased Image Retrieval with Textual Information Retrieval. In: Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005. vol. 4022 of Lecture Notes in Computer Science. Vienna, Austria; 2006. p. 652–661.
- [116] Deselaers T. Features for Image Retrieval. Master’s thesis. Human Language Technology and Pattern Recognition Group, RWTH Aachen University. Aachen, Germany; 2003.
- [117] Deselaers T, Keysers D, Ney H. Features for Image Retrieval – A Quantitative Comparison. In: DAGM 2004, Pattern Recognition, 26th DAGM Symposium. vol. 3175 of Lecture Notes in Computer Science. Tubingen, Germany; 2004. p. 228–236.
- [118] Deselaers T, Keysers D, Ney H. Discriminative Training for Object Recognition using Image Patches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). vol. 2. San Diego, CA; 2005. p. 157–162.

- [119] Fergus R, Perona P, Zisserman A. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). vol. 2. San Diego, CA, USA: IEEE; 2005. p. 380–389.
- [120] Paredes R, Perez-Cortes J, Juan A, Vidal E. Local Representations and a Direct Voting Scheme for Face Recognition. In: Workshop on Pattern Recognition in Information Systems. Setubal, Portugal; 2001. p. 71–79.
- [121] Pentland A, Picard R, Sclaroff S. Photobook: Content-Based Manipulation of Image Databases. *International Journal of Computer Vision* 1996;18(3):233–254.
- [122] Datta R, Li J, Wang J Z. Content-based Image Retrieval – Approaches and Trends of the New Age. In: ACM Intl. Workshop on Multimedia Information Retrieval, ACM Multimedia. Singapore; 2005.
- [123] Manjunath, B. S., Salembier, P., Sikora, T.: Introduction to MPEG-7 Multimedia Content Description Interface. John Wiley & Sons (2002)
- [124] Matthew MacDonald, Pro .NET 2.0 Windows Forms and Custom Controls in C#, SPRINGER, 2006.

10. Skraćenice

ASCII	American Standard for Information Interchange
BMP	Bitmap picture
CBIR	Content Based Image Retrieval
DCT	Discrete Cossinuse Tranform
DF	Document Frequency
DWT	Discrete Wavelet Transofrm
EHD	Edge Histogram Descriptor
GIF	Graphic Interchange Format
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HSV	Huse Saturation Value
IDF	Inverse Document Fraquency
IR	Information Retrieval
ISO	International Standard Organization
JPEG	Joint Picture Expert Group
KNN	K Nearest Neighbour
LSI	Latent Semantic Indexing
MPEG	Moving Picture Expert Group
NLP	Natular Language Processing
OCR	Optical Character Recognition
PIM	Personal Information Management
RDBMS	Relational Database Management System

RF	Relevance feedback
RGB	Red Green Blue
RLE	Run Length Encoding
SCD	Scalable Color Descriptor
SOM	Self-Organizing Map
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF	Term Frequency
TIFF	Tagged Image File Format
VQ	Vector Quantization
WDS	Windows Desktop Serch
XML	Extended Markup Language