
SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 196

**PRETRAŽIVANJE SLIKOVNIH BAZA
TEMELJENO NA ZNAČAJKAMA BOJE**

Slaven Lujić

Zagreb, lipanj 2008.

Sadržaj

Uvod.....	1
1. Pretraživanje slika temeljeno na sadržaju	2
2. Sustavi za pretraživanje slika	3
3. Pretraživanje slika po značajkama boje	5
3.1. RGB prostor boja	5
3.2. HSV prostor boja.....	6
4. Histogrami boja.....	8
4.1. Kvantizacija histograma.....	9
4.1.1. RGB kvantizacija	9
4.1.2. HSV kvantizacija.....	9
4.2. Usporedba histograma.....	10
5. Detalji rada	12
6. Slikovna baza	15
7. Grafičko korisničko sučelje.....	16
8. Rezultati rada.....	17
8.1. Preciznost	21
9. Zaključak.....	23
10. Literatura	24
11. Naslov, sažetak i ključne riječi.....	25

Uvod

U završnom radu bit će obrađeno pretraživanje slikovnih baza koje se temelji na značajkama boje. U tu svrhu objašnjeno je pretraživanje slika na temelju sadržaja (*eng. Content Based Image Retrieval*) ili skraćeno CBIR. Ukratko će biti opisane glavne značajke sustava CBIR, različiti načini pretraživanja slikovnih baza, kao i osnovni prostori boja koji se koriste u tu svrhu. Također će biti opisani načini određivanja značajki boje uz pomoć histograma, te algoritmi usporedbe tih značajki. Rezultat rada bit će programska aplikacija izrađena u programskom jeziku MatLab, a služiti će za provjeru rezultata načina određivanja značajki i algoritama usporedbe. Bit će oblikovana slikovna baza koja će služiti za evaluaciju rada i provjeru efikasnosti sustava.

1. Pretraživanje slika temeljeno na sadržaju

Pojavom digitalnih fotoaparata nastaje ogroman broj digitalnih fotografija i stvaraju se velike baze podataka, koje sadrže slike iz različitih područja. Tako se u bazi slika mogu pronaći umjetnička djela, satelitske, medicinske ili općenite slike, koje se mogu upotrebljavati u različitim profesionalnim područjima, na primjer u geografiji, medicini, arhitekturi, reklamiranju, dizajnu, modnoj i izdavačkoj industriji. Razvojem moćnih procesora i jeftinijih memorija slike postaju dostupnije velikom broju korisnika, pa se javlja potreba za efikasnijim i bržim pristupom željenim slikama. Baze slika mogu sadržavati ogroman broj slika koje su kategorizirane na različite načine. Slike se iz baze mogu dobiti na osnovu njihovog sadržaja ili klasifikacijom objekta, s tim da sadržaj može biti određen bojom, teksturom ili oblikom.. Ranije se najčešće koristila kategorizacija slika pomoću ključne riječi ili na osnovu opisa slike. Međutim, ovaj pristup ima poteškoća, jer različiti korisnici mogu slike kategorizirati na različite načine, što dovodi do problema kod pretraživanja slikovnih baza. Jedan od načina rješavanja ovog problema je uporaba Content-Based Image Retrieval (CBIR) sustava za pretraživanje slikovnih baza. U sustavu CBIR slike se iz baze dobivaju na osnovu vizualnih značajki, kao što su boja, tekstura i oblik. Razlozi za razvoj takvih sustava bili su postojanje velikog broja slikovnih baza, a tradicionalne metode kategorizacije pokazale su se manjkave i teške, pa su zastarjele. Korišteni su različiti pristupi ovom sustavu, a većina njih temelji se na konstrukciji histograma.

U sustavu CBIR upit može biti ostvaren pomoću skice ili najčešće korištenjem upitnog primjera slike (*eng. query image*). U svakom od ovih slučajeva cilj je pronaći slike koje najviše nalikuju upitnom primjeru. Svaka slika pohranjena u bazi ima svoje značajke koje su unaprijed određene. Prvi korak kod procesa pronalaženja željene slike je određivanje značajki upitne slike, a zatim slijedi usporedba tih značajki sa značajkama slika u bazi, što daje rezultate o vizualnoj sličnosti.

2. Sustavi za pretraživanje slika

Još od početka 90-ih godina prošlog stoljeća pretraživanje slikovnih baza na osnovu sadržaja postaje veoma zanimljivo područje istraživanja. Stvaraju se mnogi sustavi za pretraživanje slika, bilo u komercijalne ili istraživačke svrhe. Većina tih sustava podržava jednu ili više sljedećih opcija:

- nasumično prelistavanje,
- traženje po primjerima,
- traženje po skicama,
- traženje tekstom (uključuje ključnu riječ ili govor).

Možemo vidjeti da danas postoji veliki broj istraživačkih mogućnosti. Sustavne studije koje uključuju aktualne korisnike za praktičnu primjenu, još uvijek zahtijevaju istraživanja o načinima razmjene između različitih opcija .

Za tu svrhu razvilo se nekoliko sustava [1] :

1. QBIC je sustav za pretraživanje slika na osnovu upitnog primjera i prvi je takav sustav razvijen u komercijalne svrhe od strane IBM-a. Njegov okvirni rad i tehnike imale su veliki utjecaj na kasnije sustave pretraživanja slika.

QBIC podržava upite bazirane na primjeru slika na taj način da korisnik konstruira skicu ili crtež, odabire boju ili teksturu uzoraka.

Značajke boje koje su korištene u QBIC sustavu mogu se računati u RGB prostoru boja, kao i u Yig, Lab i MTM prostorima boja.

Značajke teksture predstavljaju kombinaciju hrapavosti, kontrasta i usmjerenosti.

Značajke oblika sastavljene su od prostornih oblika, zakrivljenosti, ekscentričnosti i osi orijentacije, kao i od skupa matematičkih invarijanti.

QBIC je jedan od nekoliko sustava koji izračunavaju višedimenzionalne značajke i tako označavaju sliku. Može se kombinirati tekstualno pretraživanje s pretraživanjem na temelju značajki.

2. The VIR Engine and Image Read/Write Toolkit je sustav za pretraživanje slika na temelju značajki, razvijen od strane Virage Inc. Slično QBIC sustavu Virage

podržava upite bazirane na boji, kompoziciji, teksturi i strukturi. Virage ide korak dalje od QBIC sustava i podržava proizvoljnu kombinaciju naprijed spomenutih upita.

Ovaj, kao i drugi sustavi za pretraživanje, mogu se koristiti za pretraživanje slikovnih baza u drugim institucijama koje imaju digitalizirane kolekcije slika

3. Retrieval Ware je sustav za pretraživanje slikovnih baza razvijen od strane Excalibur Technologies Corp. Njihov istraživački sustav koristi boje, oblike, teksture, sjaj i aspekt proporcija na slici kao upitne značajke. Također podržava kombinaciju ovih značajki i tako dopušta korisnicima da odabiru koja značajka ima veće značenje kod pretraživanja.
4. Photobook je skup interaktivnih alata za prelistavanje i traženje slika, a razvijen je na MIT Media Lab. Radi tako da uspoređuje značajke slika, a ne same slike. Te značajke su uglavnom boja, tekstura i oblik. Značajke se uspoređuju korištenjem odgovarajućih algoritama iz knjižnice Photobook-a.
5. VisualSEEk i WebSEEk

VisualSEEk je sustav za pretraživanje vizualnih značajki, a WebSEEk je orijentiran na traženje slika na World Wide Web-u.

Razvijeni su na Columbia University. Glavna značajka pretraživanja su prostorni odnosi upitne slike i vizualne značajke određene iz slikovne baze.

Visual SEEk temelji se na skupu boja i teksturama, a podržava upite na vizualne značajke kao i na njihove prostorne odnose.

WebSEEk se sastoji od tri glavna modula:

1. modul za prikupljanje slika i video zapisa
2. modul za klasifikaciju i označavanje
3. modul za pretraživanje i dobivanje rezultata

Postoje još Netra, MARS, ART MUSEUM, Blob-world i drugi sustavi za pretraživanje slika koji se koriste u različitim područjima..

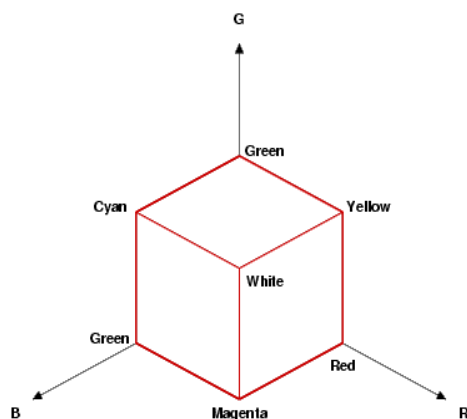
3. Pretraživanje slika po značajkama boje

Jedna od najvažnijih značajki koja ljudima omogućuje prepoznavanje i razlikovanje slika jest boja. Ta značajka je svojstvo koje ovisi o odbijanju svjetlosti od objekata u oko i obradi te informacije u mozgu. Ljudsko oko može raspoznati boje u vidljivom dijelu spektra, čije se valne duljine protežu od 380 do 780 nm. Spektar boja čine redom crvena, narančasta, žuta, zelena, plava i ljubičasta. Ljudi boje koriste za razlikovanje objekata, materijala, hrane, mjesta i ostalog. S inovacijama u području štedljivih uređaja, obrada slika u boji zauzima važno mjesto. Strojevi također mogu koristiti boje u iste svrhe kao i ljudi. Boja je naročito pogodna za klasifikaciju, jer daje višestruka mjerenja u samo jednom pixelu slike, pa nije potrebno koristiti složene prostorne odnose.

Boje su obično definirane u trodimenzionalnom prostoru, kao na primjer RGB (red, green, blue), HSV (hue, saturation, value) ili HSB (hue, saturation, brightness). HSV i HSB modeli boja ovise o ljudskoj percepciji nijanse boje, zasićenju i osvjetljenju. U sustavima za pretraživanje slika najčešće se koriste RGB i HSV modeli.

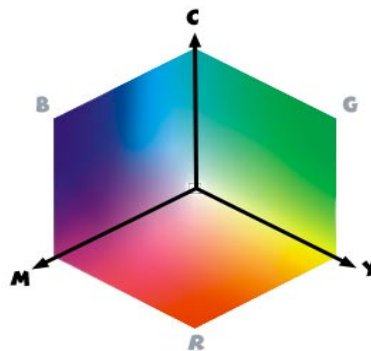
3.1. RGB prostor boja

RGB prostor [2] koristi tri osnovne boje, crvenu, zelenu i plavu, a njihovim miješanjem nastaju druge boje. Danas je ovaj prostor boja prisutan kod najvećeg broja računalnih monitora, jer ga je lako izdvojiti. U slikama u punoj boji svaki će pixel imati crvenu, zelenu i plavu vrijednost rangiranu od 0 do 255, što daje ukupan broj od $256 \times 256 \times 256 = 16777216$ boja. RGB prostor boja koristi kartezijanski koordinatni sustav (Slika 3.1).



Slika 3.1 RGB prostor boja u kartezijanskom koordinatnom sustavu

RGB prostor boja definiran je kao kocka s koordinatnim osima na kojima su smještene primarne boje. Svaku boju u prostoru predstavlja vektor s tri koordinate. Ukoliko su sve tri koordinate $[0,0,0]$ onda predstavljaju crnu boju, a koordinate $[255, 255, 255]$ predstavljaju bijelu boju. Pogled na RGB prostor boja od bijele u centru do primarnih boja pokazuje Slika 3.2.

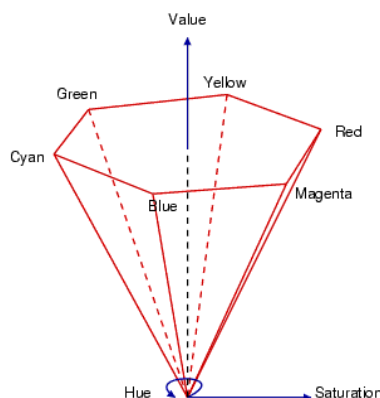


Slika 3.2 RGB prostor boja

3.2. HSV prostor boja

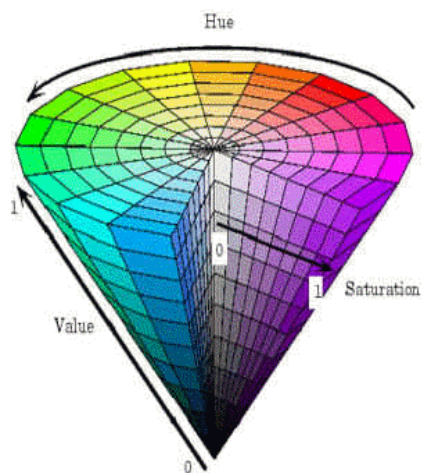
Ovaj prostor boja [3] uveo je 1978. godine Alvy Ray Smith, a temelji se na načinu na koji umjetnici miješaju boje na svojim paletama. Ovo je jako važan prostor, jer opisuje boje najbližije ljudskoj percepciji. To svojstvo može ovaj prostor učiniti boljim izborom od RGB prostora za pretraživanje slika.

HSV prostor boja definira boju njenom nijansom, zasićenosti i vrijednosti točnije od osnovnih boja. Koordinatni sustav za HSV prostor boja je šesterostrana piramida kao na Slika 3.3.



Slika 3.3 Koordinatni sustav za HSV prostor boja

Nijansa je opisana položajem boje u krugu u kojem se boje glatko mijenjaju između osnovnih boja (Slika 3.4). Zasićenost predstavlja dominantnu nijansu u boji te se rangira od 0 do 100%. Crna, bijela i nijanse sive se smatraju nezasićenim bojama. Vrijednost opisujemo kao ukupan intenzitet svjetlosti u boji.



Slika 3.4 Stožasti prikaz HSV prostora boja

4. Histogrami boja

Da bi pretraživanje slika u sustavu CBIR bilo efikasno, potrebno je odrediti neke značajke slike kako bismo ih mogli usporediti sa slikama u bazi. Boja je najpopularnija značajka za označavanje slika i pretraživanje, pa se zato konstruiraju histogrami boja [4]. Histogram boja se konstruira prebrojavanjem broja pixela svake boje, koji se smještaju u točno određenu kućicu na histogramu. Te kućice se nazivaju binovi. Histogram je zapravo dvodimenzionalni graf kojem su na osi x svi binovi boja, a na osi y količina pixela svakog bina. Histogram slike u boji može se upotrijebiti za pronalaženje slike ili prepoznavanje objekta, a također se uspoređivanjem dvaju histograma može saznati koliko su slične ili različite dvije ili više slika.

Histogram boje je jednoznačno definiran jednačbom vjerojatnosti (1):

$$h_{A,B,C}(a,b,c) = N \cdot \Pr ob(A = a, B = b, C = c) \quad (1)$$

gdje A,B i C predstavljaju tri kanala boje ovisno o odabranom prostoru boje (RGB ili HSV), a N je broj pixela u slici.

Histogrami boja mogu se podijeliti u dvije osnovne kategorije i to:

1. globalni histogrami boja
2. lokalni histogrami boja.

Globalni histogram slike predstavlja cijelu sliku samo s jednim histogramom boja, za razliku od lokanih histograma koji sliku dijele na fiksne dijelove i izračunavaju histogram boja za svaki dio posebno. Lokalni histogrami sadrže prostorne informacije , ali to zahtjeva povećanje vremena potrebnog za računanje i spremanje informacija. U pretraživanju slika na temelju boje uglavnom se koriste globalni histogrami , iako ne sadrže informaciju o podjeli boje po regijama.

4.1. Kvantizacija histograma

Uspoređivanje svih boja na dvije slike može biti kompleksan i dugotrajan proces, pa se koriste metode kojima se reducira količina informacija. Jedna od njih je kvantiziranje rasporeda boja u histogramima, a prvi su je uveli Swain i Ballard [5]. Kasnije su kvantizaciju koristili mnogi pa možemo reći da je to jedan od popularnijih pristupa u pretraživanju slika danas.

4.1.1. RGB kvantizacija

U slikama u boji RGB sustava svaki pixel ima crvenu, zelenu i plavu vrijednosti rangiranu od 0 do 255 dajući ukupan broj od 16777216 različitih binova. Problem predstavlja izračunavanje i uspoređivanje histograma takvih slika, jer to zahtijeva puno vremena. Iz tog razloga koristi se metoda redukcije binova koja se naziva kvantizacija. Tom metodom vrijednosti pojedinih binova smještaju se u manji raspon vrijednosti. U RGB prostoru najčešće se koristi raspon vrijednosti od 0 do 7 što daje $8 \times 8 \times 8 = 512$ binova. Budući da boje u RGB prostoru boja imaju istu udaljenost one se kvantiziraju u iste nivoe. Ovo je kompromis između efikasnosti i brzine. Veći broj binova od ovoga dovodi do vrlo malog povećanja učinkovitosti, a posljedica je veliko smanjenje brzine.

4.1.2. HSV kvantizacija

Kvantizacija u HSV prostoru boja uglavnom je usmjerena na kvantizaciju nijanse, jer ona ima najvažniju informaciju o boji na slici. Nijansa je u HSV prostoru boja prikazana kao krug u kojem su primarne boje odvojene za 120 stupnjeva. Rastavljanjem tog kruga na dijelove od 20 stupnjeva, razdvajaju se nijanse primarnih boja na žutu, tirkiznu i ljubičastu. Za nijansu je zbog toga dovoljno 18 nivoa kvantizacije. Zasićenje i vrijednost imaju manje važnu informaciju od nijanse, te je za njih potrebno manje nivoa kvantizacije. Ako zasićenje i vrijednost kvantiziramo u tri nivoa dobit ćemo $18 \times 3 \times 3 = 162$ bina.

4.2. Usporedba histograma

Kad su histogrami kvantizirani i izračunati potrebna je metoda usporedbe tih histograma. Postoji više metoda izračunavanja udaljenosti između histograma, a najčešće se koriste sljedeće tri :

1. Euklidova udaljenost gdje h i g predstavljaju dva histograma boja. Euklidova udaljenost može se izračunati sljedećom jednačbom (2):

$$d^2(h, g) = \sum_A \sum_B \sum_C (h(a, b, c) - g(a, b, c))^2 \quad (2)$$

Problem kod ove jednačbe je što se uspoređuju samo dva identična bina, iako dva susjedna bina predstavljaju perceptivno slične boje.

2. Presijecanje histograma (*eng. intersection*) h i g računa se jednačbom (3):

$$d(h, g) = \frac{\sum_A \sum_B \sum_C \min(h(a, b, c), g(a, b, c))}{\min(|h|, |g|)} \quad (3)$$

gdje $|h|$ i $|g|$ predstavljaju ukupan broj uzoraka. Boje kojih nema u upitnoj slici neće doprinosti presjeku. To smanjuje utjecaj pozadinskih boja.

3. Kvadratna udaljenost histograma korištena je u QBIC sustavu. Računa se sljedećom jednačbom (4):

$$d(h, g) = (h - g)^t A (h - g), \quad (4)$$

Za razliku od Euklidove jednačbe kvadratna udaljenost uzima u obzir korelaciju između svih binova. Korelacija između svih binova predstavljena je matricom A , koja se naziva matrica sličnosti.

Za RGB prostor element $a_{i,j}$ matrice A definiran je sljedećom jednačbom (5):

$$a_{i,j} = 1 - \frac{d_{i,j}}{\max(d_{i,j})} \quad (5)$$

gdje je $d_{i,j}$ Euklidova udaljenost između boje i i j u RGB sustavu.

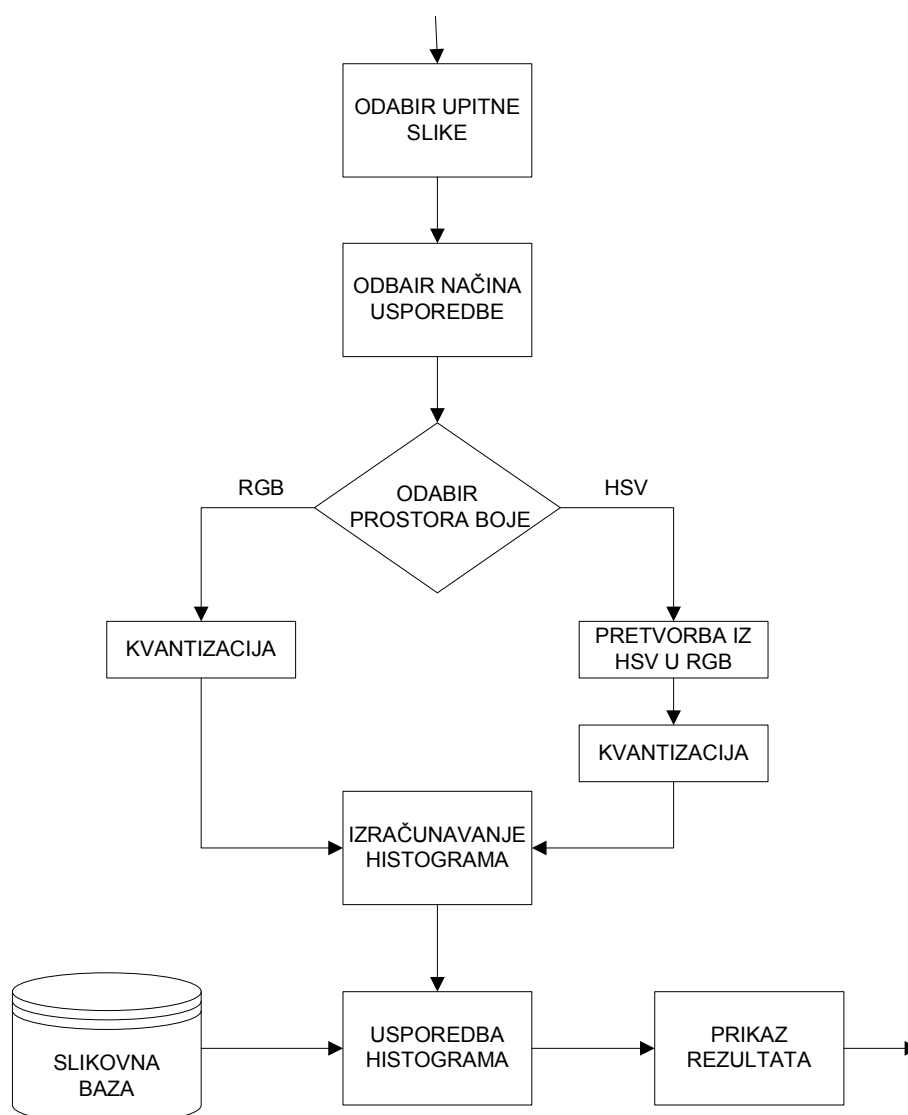
Za HSV prostor element $a_{i,j}$ matrice A definiran je sljedećom jednađžbom (6):

$$a_{i,j} = 1 - \frac{1}{\sqrt{5}} \left[(v_i - v_j)^2 + (s_i \cos(h_i) - s_j \cos(h_j))^2 + (s_i \sin(h_i) - s_j \sin(h_j))^2 \right]^{\frac{1}{2}} \quad (6)$$

koja odgovara sličnosti boja u HSV prostoru.

5. Detalji rada

Svoj sustav za pretraživanje slika napravio sam u programskom jeziku MatLab. Ovaj jezik sam odabrao zbog ugrađenih funkcija za obradu slike i zbog jednostavnosti korištenja.



Slika 5.1 Dijagram toka aplikacije

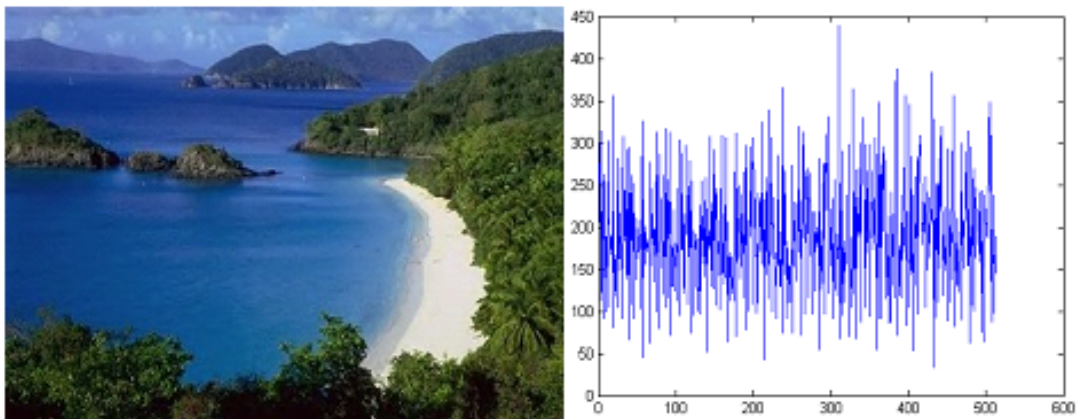
Ja sam svoju aplikaciju za pretraživanje slika implementirao prema dijagramu na Slika 5.1. Program radi tako da se na početku izabere upitna slika i način usporedbe histograma. Još je potrebno odabrati prostor boja u kojem će se vršiti pretraživanje. Slike pohranjene u slikovnoj bazi definirane su u RGB prostoru boja. Ako odaberemo taj prostor onda se upitna slika kvantizira i konstruira se histogram upitne slike. Ukoliko želimo pretraživati u

HSV prostoru boja, potrebno je najprije slike pretvoriti u taj prostor. Nakon toga vrši se kvantizacija u HSV prostoru boja i konstruira njen histogram. Ovisno o odabranoj metodi uspoređivanja histograma upitne slike s histogramima slika u slikovnoj bazi, prikazuju se rezultati pretraživanja.

Nakon početnog istraživanja sustava za pretraživanje slikovnih baza naišao sam na dilemu koji prostor boje izabrati za sustav. Ustanovio sam da se najčešće koriste RGB i HSV prostori boja. Prema nekim izvorima RGB prostor daje bolje rezultate, dok je prema drugima HSV bolji izbor za pretraživanje slikovnih baza.

Odlučio sam implementirati oba prostora boje i uvjeriti se koji prostor daje bolje rezultate, odnosno s kojim mogu brže dobiti preciznije rezultate.

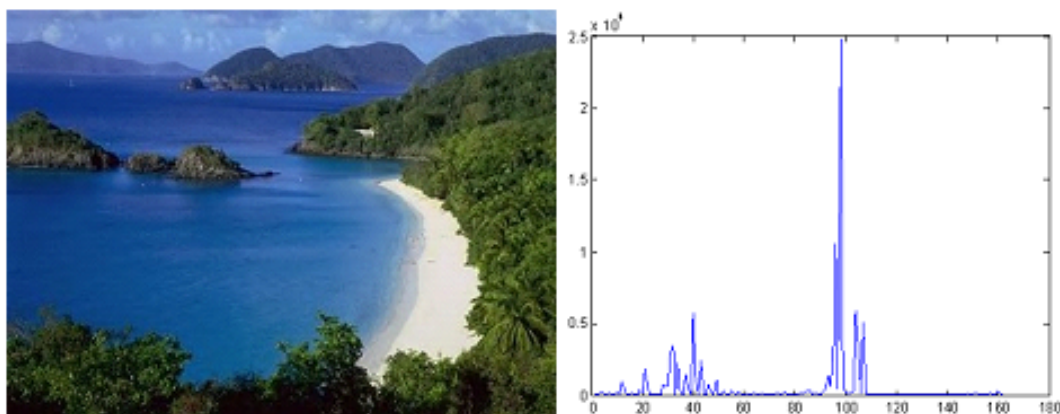
Čim sam počeo raditi u RGB prostoru shvatio sam da izračunavanje histograma zahtijeva jako puno vremena i da je takav sustav neefikasan, pa sam koristio metodu kvantizacije za redukciju broja binova potrebnih za konstrukciju histograma. U literaturi [4] sam pronašao da je idealan omjer kvantizacije $8 \times 8 \times 8 = 512$ binova, pa sam odlučio koristiti isti. Kvantizacija se provodi operacijom modulo s brojem 8. Raspon vrijednosti boje je umjesto od 0 do 255, reduciran na vrijednosti od 0 do 7.



Slika 5.2 Slika mora i njezin histogram u RGB prostoru boja

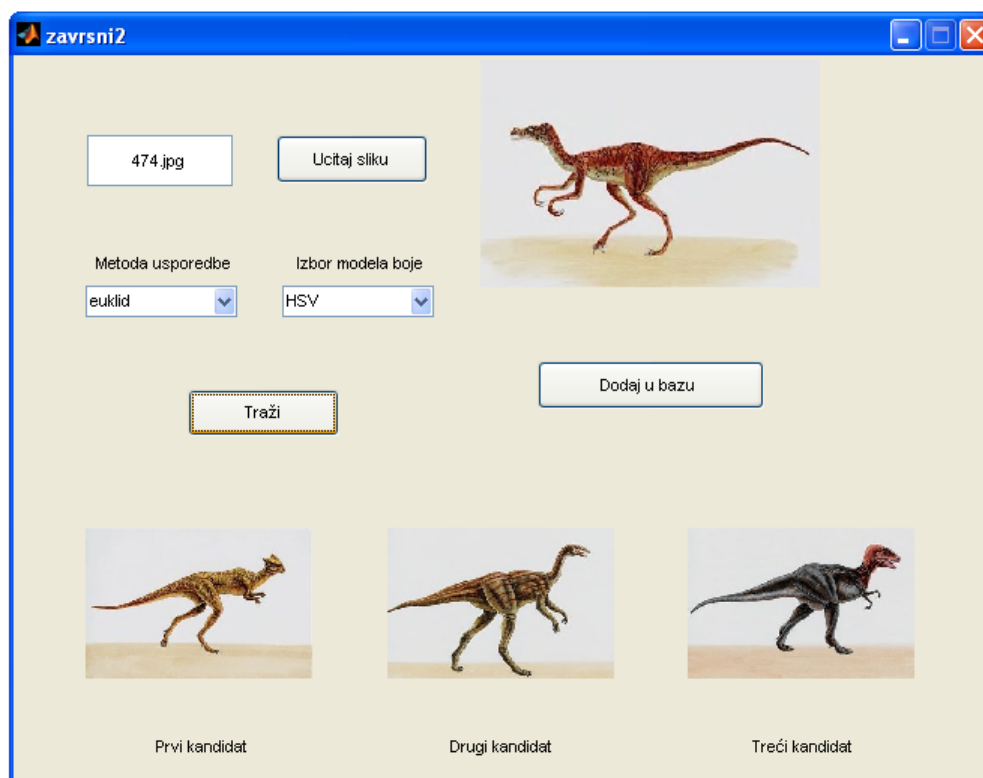
Nakon RGB kvantizacije bilo je potrebno kvantizirati i slike u HSV prostoru. Slike koje sam koristio su u RGB prostoru, te ih potrebno prvo prebaciti u HSV prostor boja. Nakon toga provodi se kvantizacija slika u HSV prostoru na način opisan u poglavlju 4.1.2.

Nakon kvantizacije slike potrebno je konstruirati njezin histogram. To je jednostavan postupak, jer je potrebno samo prebrojati broj pixela određenog bina.



Slika 5.3 Slika mora i njezin histogram u HSV prostoru boja

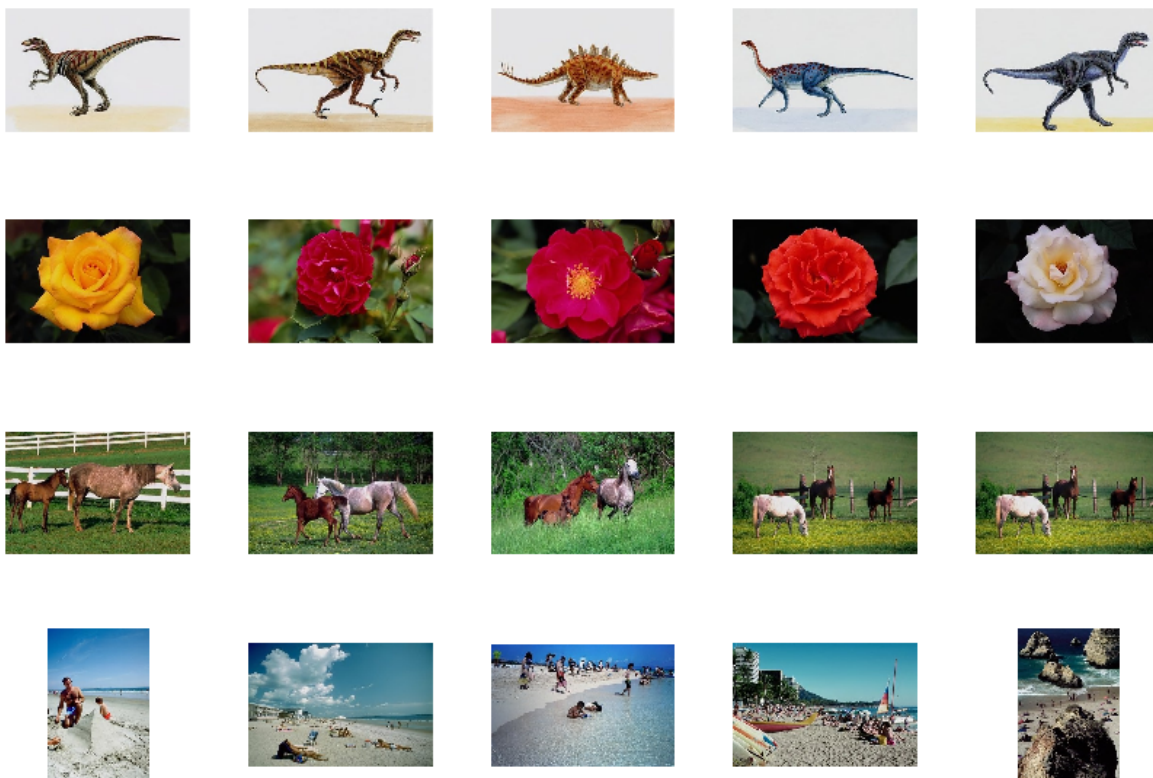
Za uspoređivanje histograma koristio sam dvije metode usporedbe. To su Euklidova udaljenost (2) i presijecanje histograma (3). Nakon što je konstruiran histogram upitne slike, on se ovisno o odabranoj metodi uspoređuje sa svim histogramima slika u slikovnoj bazi. Moja aplikacija prikazuje samo tri najslićnije slike (Slika 5.4), dok se ostale indeksiraju po redoslijedu sličnosti



Slika 5.4 Prikaz rezultata pretraživanja

6. Slikovna baza

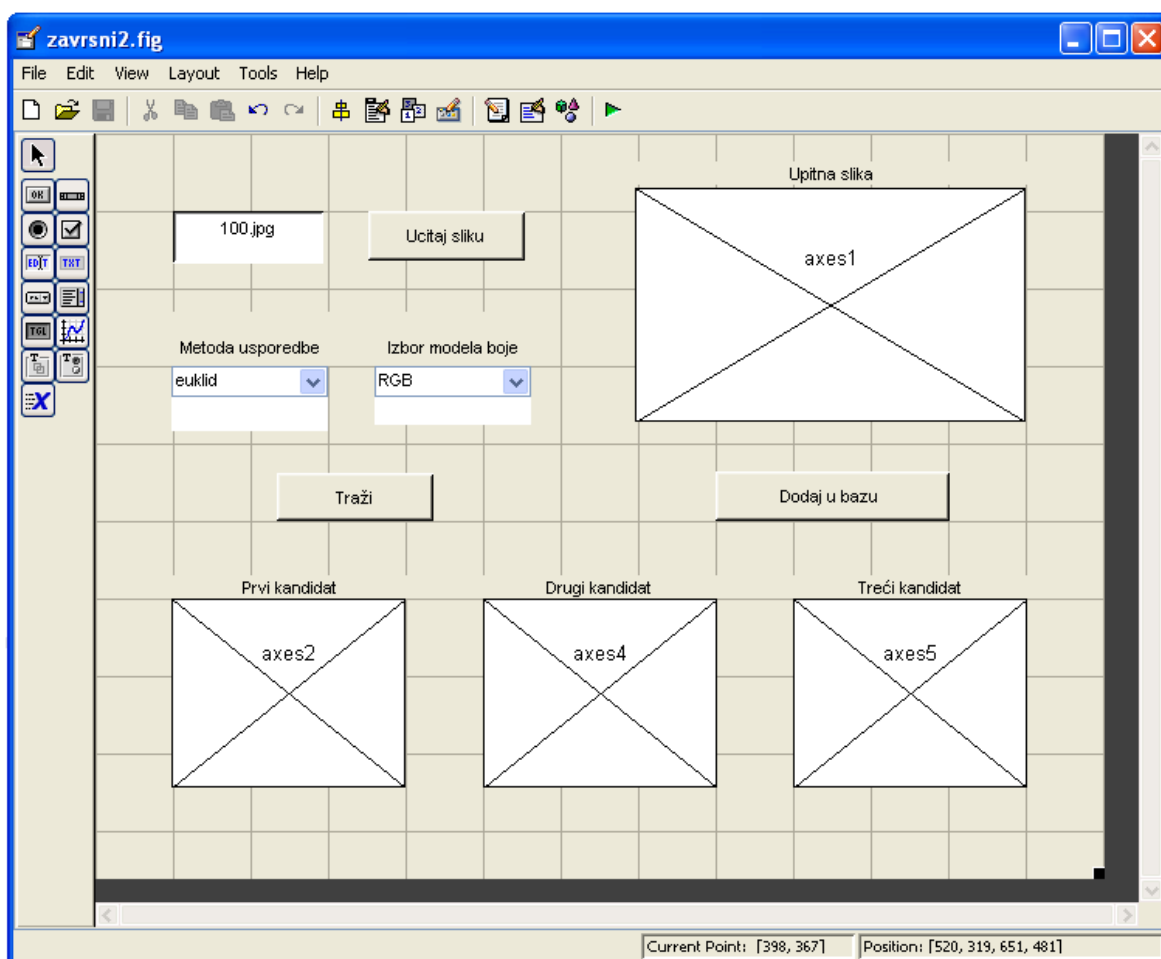
Za izradu slikovne baze koristio sam slike dostupne na (<http://wang1.ist.psu.edu/>). Taj paket sadrži 1000 slika. Slike su u formatu *JPEG*. Za evaluaciju rezultata rada oblikovao sam vlastitu slikovnu bazu za proučavanje različitih kombinacija značajki boje i mjera sličnosti. Ta baza se sastoji od 80 slika podijeljenih po grupama. Jednu grupu čine dinosaurusi, drugu cvijeće, treću konji a zadnju slike mora. Svaka grupa sadrži 20 primjeraka. Dio slikovne baze prikazan je na Slika 6.1 Ostale slike iz tih kategorija služile su mi kao upitni primjeri za provjeru rezultata. Kad govorimo o slikovnoj bazi, potrebno je naglasiti da ona zapravo predstavlja matricu s brojem redaka, koji je jednak broju slika u bazi, a svaki redak predstavlja histogram određene slike. U zadnjem stupcu te matrice nalazi se broj koji jednoznačno određuje sliku čiji je histogram u pripadajućem retku.



Slika 6.1 Dio slikovne baze

7. Grafičko korisničko sučelje

Za izradu ispitne okoline bilo je potrebno napraviti grafičko korisničko sučelje (Slika 7.1). U tu svrhu koristio sam modul GUIDE u programskom jeziku MatLab, koji omogućuje jednostavnu i brzu izgradnju sučelja. Sučelje se sastoji od više elemenata koji služe za odabir slike, prostora boja, metode usporedbe, upitnog primjera slike te od tri prozora za prikaz rezultata. Također postoji tipka koja konstruira histogram upitne slike i dodaje ga u slikovnu bazu. Nakon izbora slike, metode usporedbe i prostora boje pritiskom na tipku „Traži“ pokreće se pretraživanje i pokazuju se tri najbolja kandidata.



Slika 7.1 Grafičko korisničko sučelje

8. Rezultati rada

Za provjeru rezultata rada sustava odlučio sam s jednom upitnom slikom iz svake grupe slika pretražiti slikovnu bazu i proučiti dobivene rezultate.



HSV Euklidova udaljenost



HSV presijecanje histograma

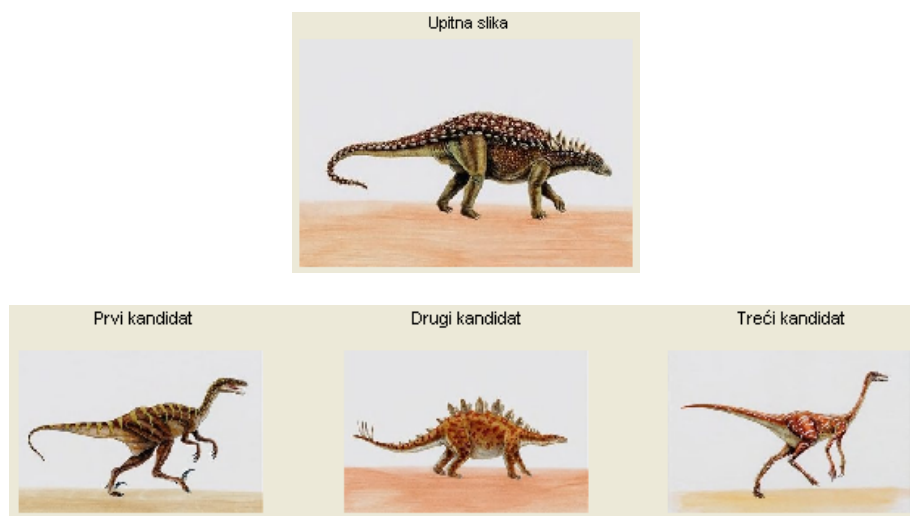


RGB Euklidova udaljenost



RGB presijecanje

Iz dobivenih rezultata vidljivo je da sustav bolje radi u HSV prostoru boja na slikama s temom mora. Metoda presijecanja histograma daje uvjerljivije rezultate od Euklidove udaljenosti.



HSV Euklidova udaljenost



HSV presijecanje histograma



RGB Euklidova udaljenost



RGB presijecanje histograma

Na slikama koje prikazuju dinosaure vidljivo je da obje metode daju gotovo jednako dobre rezultate, odnosno imaju dovoljno dobru preciznost. Pretraživanjem u RGB prostoru boja malo je bolje pogođena boja dinosaura.



HSV Euklidova udaljenost



HSV presijecanje histograma



RGB Euklidova udaljenost



RGB presijecanje histograma

U grupi koja prikazuje cvijeće vidljivo je da pretraživanje slika radi bolje u HSV prostoru boja, iako i RGB prostor ima jako dobru preciznost. Također je vidljivo da boja pozadine ima veći utjecaj na slike u RGB prostoru boja.



HSV Euklidova udaljenost



HSV presijecanje histograma



RGB Euklidova udaljenost



RGB presijecanje histograma

U grupi slika s motivom konja, HSV prostor boja opet pokazuje bolje osobine i ima puno veću preciznost, dok za RGB prostor boja primjećujemo veliki utjecaj pozadine.

8.1. Preciznost

Za provjeru dobivenih rezultata također sam napravio funkciju koja računa preciznost određene metode i prostora boja.

Preciznost sustava za pretraživanje slika može se računati sljedećom formulom:

$$preciznost = \frac{a}{a + b}$$

gdje a predstavlja broj nađenih relevantnih slika, a b broj nađenih irelevantnih slika. Budući da se u mojem sustavu za pretraživanje, slike indeksiraju od sličnijih prema manje sličnima i obzirom da je unaprijed određen broj slika po grupama, preciznost nije teško izračunati. Preciznost bi bila 100% kad bi sustav na osnovu upitne slike sve relevantne slike u mojoj bazi poredao na prva mjesta po sličnosti. Na primjer, u bazi podataka postoji 20 slika cvijeća. Preciznost bi bila 100% kad bi sustav svih 20 slika stavio na prvih 20 mjesta po sličnosti. Ukoliko sustav na jedno od prvih 20 mjesta stavi sliku iz neke druge grupe preciznost se smanjuje.

Tabela 8.1 Efikasnost sustava za pretraživanje slika

	RGB euklid	RGB presijecanje	HSV euklid	HSV presijecanje
More	25%	25%	25%	42%
Dinosaurusi	68%	68%	86%	70%
Cvijeće	71%	71%	66%	74%
Konji	76%	76%	59%	77%

Efikasnost sustava prikazana je u Tabela 8.1. Tabela je dobivena na osnovu 10 upitnih slika iz svake grupe slika, koje sam usporedio sa slikama u bazi i to u oba prostora boja i pomoću obje metode usporedbe. Dobivena preciznost je prosjek preciznosti svih 10 slika. Iz tabele je vidljivo da pretraživanje u HSV prostoru boja metodom presijecanja histograma daje najbolje rezultate. Također se pokazalo da pretraživanje slika s dinosaurusima daje najbolje rezultate. Razlog tome je da na tim slikama uglavnom prevladava jedna boja. Najlošije rezultate daju slike s motivom mora, jer je na njima

različnost boja najistaknutija. Također se pokazalo da pretraživanje u RGB prostoru boje daje potpuno identične rezultate za obje metode usporedbe. Važno je napomenuti da je izračunavanje histograma u HSV prostoru znatno brže, a također je i metoda presijecanja histograma brža od metode računanja Euklidove udaljenosti.

9. Zaključak

Velika ekspanzija različitih uređaja u boji, koji se koriste, kako u svakodnevnom životu, tako i u raznim profesionalnim područjima, dovela je do stvaranja slikovnih baza s ogromnim brojem slika. Razvoj i usavršavanje sustava CBIR omogućili su efikasnije i brže pretraživanje slikovnih baza na osnovu upitnog primjera slike.

Slikovne baze mogu se pretraživati različitim metodama, ovisno o odabranoj značajki slike. Kod pretraživanja baza temeljenih na značajkama boje, najčešće se koriste RGB i HSV prostori boja. Oba prostora temelje se na konstrukciji i usporedbi histograma boje upitne slike s histogramima spremljenim u bazu.

U literaturi sam naišao na različita mišljenja o tome koji prostor daje bolje rezultate. Oblikovanjem vlastite ispitne okoline i evaluacijom rezultata rada, došao sam do zaključka da pretraživanje slikovnih baza na temelju primjera upitne slike, u HSV prostoru boja, u obje metode usporedbe, daje bolje rezultate od RGB prostora boja.

CBIR sustav osim pretraživanja baza po značajkama boje, koristi i ostale značajke kao što su tekstura i oblik. Da bi moj sustav za pretraživanje davao još bolje i preciznije rezultate trebalo bi implementirati i pretraživanje po navedenim značajkama.

10. Literatura

- [1] Colin C. Venters and Dr. Matthew Cooper, *A Review of Content-Based Image Retrieval Systems*
URL: <http://www.jisc.ac.uk/media/documents/programmes/jtap/jtap-054.pdf>
Datum pristupa dokumentu : 1.6.2008
- [2] Wikipedia članak o RGB prostoru boja, *RGB color model*
URL: http://en.wikipedia.org/wiki/RGB_color_model
Datum pristupa dokumentu : 2.6.2008
- [3] Wikipedia članak o HSV prostoru boja, *HSL i HSV*
URL: http://en.wikipedia.org/wiki/HSL_and_HSV
Datum pristupa dokumentu : 2.6.2008
- [4] Linda G. Shapiro i George C. Stockman, *Computer vision: Color and Shading*, Prentice Hall, 2001.
- [5] Michael J. Swain i Dana H. Ballard, *Color Indexing*, International Journal of Computer Vision, Broj 7(1), str. 11-32, 1991

11. Naslov, sažetak i ključne riječi

Naslov rada: Pretraživanje slikovnih baza temeljeno na značajkama boje

Sažetak: U ovom radu objašnjeno je pretraživanje slikovnih baza temeljeno na značajkama boje. Vezano za to prezentiran je kratak uvod u CBIR i postojeće sustave za pretraživanje slikovnih baza. Obradena su dva najčešće korištena prostora boje (RGB i HSV) za pretraživanje slikovnih baza, te dvije metode za usporedbu histograma upitne slike s histogramima u bazi. Ukratko je objašnjena programska aplikacija ispitne okoline u programskom jeziku MatLab za provjeru korištenih metoda i prikaz rezultata rada. Oblikovana je vlastita slikovna baza i aplikacija za grafičko korisničko sučelje. Na kraju su prikazani i ocijenjeni rezultati rada. Provjera rezultata pokazala je da pretraživanje baza u HSV prostoru boja daje bolje rezultate od RGB prostora boja, te da je metoda presijecanja histograma bolja od metode Euklidove udaljenosti.

Ključne riječi: pretraživanje, boja, značajka, upitna slika, RGB prostor boja, HSV prostor boja, histogram, kvantizacija, presijecanje, preciznost,

Title: Color based image database retrieval

Abstract: In this report color based image retrieval was described. A short introduction to CBIR and existing retrieval systems was presented. Two of the most used color models for image retrieval (RGB and HSV) and two methods for comparing image histograms were described. Program application for test environment in computing language MatLab for evaluating and displaying results was shortly described. Image database and graphical user interface was built. In the end of report final results were displayed and evaluated. Results have shown that image retrieval in HSV color model gives better results than RGB color model, and that histogram intersection is better method than Euclidian distance for comparing histograms.

Keywords: retrieval, color, feature, query image, RGB color space, HSV color space, histogram, quantization, intersection, precision

Dodatak A: Programski kod

Zavrsni.m

```
function varargout = zavrsni2(varargin)
%ZAVRSNI2 M-file for zavrsni2.fig
%     ZAVRSNI2, by itself, creates a new ZAVRSNI2 or raises the existing
%     singleton*.
%
%     H = ZAVRSNI2 returns the handle to a new ZAVRSNI2 or the handle to
%     the existing singleton*.
%
%     ZAVRSNI2('Property','Value',...) creates a new ZAVRSNI2 using the
%     given property value pairs. Unrecognized properties are passed via
%     varargin to zavrsni2_OpeningFcn. This calling syntax produces a
%     warning when there is an existing singleton*.
%
%     ZAVRSNI2('CALLBACK') and ZAVRSNI2('CALLBACK',hObject,...) call the
%     local function named CALLBACK in ZAVRSNI2.M with the given input
%     arguments.

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @zavrsni2_OpeningFcn, ...
                  'gui_OutputFcn',  @zavrsni2_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before zavrsni2 is made visible.
function zavrsni2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% Choose default command line output for zavrsni2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes zavrsni2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = zavrsni2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% Hints: get(hObject,'String') returns contents of edit1 as text
%           str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Pokreće se pritiskom na tipku "Učitaj sliku" i prikazuje upinu
sliku.
function push1_Callback(hObject, eventdata, handles)
% hObject      handle to push1 (see GCBO)
image = get(handles.edit1,'String');
I = imread(char(image));
axes(handles.axes1);
imshow(I);
model = get(handles.pop4,'Value')

% Pokreće se pritiskom na tipku Traži i
% počinje izračunavanje histograma i usporedba.
function push2_Callback(hObject, eventdata, handles)
% hObject      handle to push2 (see GCBO)
image = get(handles.edit1,'String'); %ime slike
I = imread(char(image));             %citanje slike
nacin = get(handles.pop,'Value');
model = get(handles.pop4,'Value');
if(model == 1)
    if(nacin == 1)
        histogram;           %odabir modela boje,
                               %nacina usporedbe,
                               %racunanje histograma
        euklid;               %i usporedba s bazom slika
    else
        histogram;
        intersection;
    end
else
    if(nacin == 1)
        histogramHSV;         %odabir modela boje
                               %nacina usporedbe,
                               %racunanje histograma
    else
        histogramHSV;         %i usporedba s bazom slika
        intersectionHSV;
    end
end
end

```

```

% --- Executes on selection change in pop.
function pop_Callback(hObject, eventdata, handles)
% hObject    handle to pop (see GCBO)

% --- Executes during object creation, after setting all properties.
function pop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop (see GCBO)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Pokreće se protiskom na tipku Dodaj u bazu.
function push3_Callback(hObject, eventdata, handles)
% hObject    handle to push3 (see GCBO)

dodaj_u_bazu;                %funkcija za dodavanje upitne slike
                             %u bazu podataka

% --- Executes on selection change in pop4.
function pop4_Callback(hObject, eventdata, handles)
% hObject    handle to pop4 (see GCBO)

% --- Executes during object creation, after setting all properties.
function pop4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pop4 (see GCBO)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

kvantizacija.m

```

%------%
%funkcija koja kvantizira RGB sliku u prostor 8x8x8
%------%
[R,C,colourDepth] = size(I);
for z = 1 : 3;
for j = 1 : R;
for i = 1 : C;

    I(j, i, z) = mod(I(j, i, z), 8);

end
end
end

```

histogram.m

```

%------%
%funkcija koja računa histogram kvantizirane RGB slike
%------%
his = ones(1,512);

```

```

[R,C,colourDepth] = size(I);           %veličina slike
kvantizacija;
for j = 1 : R;
for i = 1 : C;
z = 1;
    for x = 0 : 7;
    for y = 0 : 7;
    for n = 0 : 7;
        if(I(j,i,1) == x && I(j,i,2) == y && I(j,i,3) == n)
            his(z) = his(z) + 1;
            z = z + 1;
        else
            z = z + 1;
        end
    end
    end
end
end
end

```

histogramHSV.m

```

%-----%
%funkcija koja pretvara sliku u HSV prostor boja,
%kvantizira ju i računa njezin histogram
%-----%
H = rgb2hsv(I);
his = ones(1,162);
[R,C,colourDepth] = size(H);           %veličina slike

for j = 1 : R;
for i = 1 : C;
z = 1;
    for x = 0 : 17;
    for y = 0 : 2;
    for n = 0 : 2;
        if(H(j,i,1) > x/18 && H(j,i,1) < (x+1)/18 && H(j,i,2) > y/3 && H(j,i,2) <
(y+1)/3 && H(j,i,3) > n/3 && H(j,i,3) < (n+1)/3)
            his(z) = his(z) + 1;
            z = z + 1;
        else
            z = z + 1;
        end
    end
    end
end
end
end
end

```

euklid.m

```

%-----%
%funkcija koja računa Euklidovu udaljenost histograma
%upitne slike u RGB prostoru boja sa svim histogramima
%u bazi i rezultat sprema u polje d
%-----%
load DB;
[m n] = size(baza);

```

```

d = zeros([1 m]);
for x = 1 : m;
for z = 1 : 512;
    d(x) = d(x) + sqrt((his(z) - baza(x, z))*(his(z) - baza(x, z)));
end
end
x=d
Rezultati_Euklid          %prikaz 3 najbolja kandidata
preciznost                %poziv funkcije koja računa preciznost

```

euklidHSV.m

```

%-----%
%funkcija koja računa Euklidovu udaljenost histograma
%upitne slike u HSV prostoru boja sa svim histogramima
%u bazi i rezultat sprema u polje d
%-----%
load DBHSV;
[m n] = size(bazaHSV);
d = zeros([1 m]);
for x = 1 : m;
for z = 1 : 162;
    d(x) = d(x) + sqrt((his(z) - bazaHSV(x, z))*(his(z) - bazaHSV(x, z)));
end
end
x=d;
Rezultati_Euklid          %prikaz 3 najbolja kandidata
preciznost                %poziv funkcije koja računa preciznost

```

intersection.m

```

%-----%
%funkcija koja računa presijecanje histograma
%upitne slike u RGB prostoru boja sa svim histogramima
%u bazi i rezultat sprema u polje d
%-----%
load DB;
[m n] = size(baza);
d = zeros([1 m]);

for x = 1 : m;
for z = 1 : 512;
    d(x) = d(x) + min(his(z), baza(x, z));
end
    d(x) = d(x)/(min(sum(his), sum(baza(x, :))));
end
x=d
Rezultati_Intersection;   %prikaz 3 najbolja kandidata
preciznost                %funkcija koja računa preciznost

```

intersectionHSV.m

```
%-----%
%funkcija koja računa presijecanje histograma
%upitne slike u HSV prostoru boja sa svim histogramima
%u bazi i rezultat sprema u polje d
%-----%
load DBHSV;
[m n] = size(bazaHSV);

d = zeros([1 m]);

for x = 1 : m;
for z = 1 : 162;
    d(x) = d(x) + min(his(z), bazaHSV(x, z));
end
    d(x) = d(x)/(min(sum(his), sum(bazaHSV(x, :))));
end
x=d
Rezultati_Intersection;          %prikaz 3 najbolja kandidata
preciznost;                      %funkcija koja računa preciznost
```

Rezultati_Euklid.m

```
%-----%
%funkcija koja služi za prikaz 3 najbolja kandidata
%-----%
if(model == 1)                    %odabir modela boje
    pom1 = 513;                  %da se sazna koja će
    load DB                      %se baza koristiti
    bazatmp1 = baza;
else
    load DBHSV
    bazatmp1 = bazaHSV;
    pom1 = 163;
end
axes(handles.axes2);             %prikaz prvog kandidata
max = max(d);
[min1, ind] = min(d);
d(ind) = max+1;

tmp = bazatmp1(ind, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);

axes(handles.axes4);            %prikaz drugog kandidata

[min2, ind2] = min(d);
d(ind2) = max;

tmp = bazatmp1(ind2, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);
```

```

axes(handles.axes5);           %prikaz trećeg kandidata
[min3, ind3] = min(d);
d(ind3)= max;

tmp = bazatmp1(ind3, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);

```

Rezultati_Intersection.m

```

%-----%
%fukcija koja služi za prikaz 3 najbolja kandidata
%-----%
if(model == 1)                %odabir modela boje
    pom1 = 513;                %da se sazna koja će
    load DB                    %se baza koristit
    bazatmp1 = baza;
else
    load DBHSV
    bazatmp1 = bazaHSV;
    pom1 = 163;
end
axes(handles.axes2);          %prikaz prvog kandidata
min = min(d);
[max1, ind] = max(d);
d(ind)= min;

tmp = bazatmp1(ind, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);

axes(handles.axes4);          %prikaz drugog kandidata

[max2, ind2] = max(d);
d(ind2)= min;

tmp = bazatmp1(ind2, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);

axes(handles.axes5);          %prikaz trećeg kandidata
[max3, ind3] = max(d);
d(ind3)= min;

tmp = bazatmp1(ind3, pom1);
str = int2str(tmp);

slika = [ str '.jpg' ];
imshow(slika);

```

preciznost.m

```
%-----%
%funkcija koja računa preciznost u ovisnosti o odabranoj
%metodi pretrage, vrsti slika i modelu boje
%-----%
a=0;
b=0;
image = get(handles.edit1,'String');
[broj ime] = strread(image, '%d %s', 'delimiter', '. ');
broj = floor(broj/100);
if(model == 1)
    load DB
    bazatmp = baza;
    pom = 513;
else
    pom = 163;
    load DBHSV
    bazatmp = bazaHSV;
end
if(nacin == 1)
    [recall, index] = sort(x);
    index
else
    [recall, index] = sort(x, 'descend');
    index
end
[m n] = size(bazatmp);
for brojac = 1 : m
    if(brojac<21)
        if(floor(bazatmp(index(brojac),pom)/100) == broj)
            a=a+1;
        else
            b = b+1;
        end
    end
end
precision = a/(a+b)
```

dodaj_u_bazu.m

```
%-----%
%funkcija koja dodaje histogram slike u slikovnu bazu
%histogram u RGB i HSV prostoru boja
%-----%
image = get(handles.edit1,'String'); %dohvat slike
I = imread(char(image));
histogram;
load DB;
[m, n] = size(baza);
m= m+1;
for i = 1 : 512 %spremanje histograma
    baza(m, i) = his(i); %u bazu histograma u RGB
end %prostoru boja

[broj ime] = strread(image, '%d %s', 'delimiter', '. ');
baza(m, 513) = broj; %označavanje slike u bazi
save DB baza;
```

```
image = get(handles.edit1, 'String');
I = imread(char(image));
H = rgb2hsv(I);
histogramHSV;
load DBHSV;
[m, n] = size(bazaHSV);
m= m+1;
for i = 1 : 162                                %spremanje histograma
    bazaHSV(m, i) = his(i);                    %u bazu histograma u HSV
end                                              %prostoru boja

[broj ime] = strread(image, '%d %s', 'delimiter', '. ')
bazaHSV(m, 163) = broj;                        %označavanje slike
save DBHSV bazaHSV;
```