

Sivoskalirane slike

Irhad Šarić

Prirodno-matematički fakultet u Sarajevu
Matematičke metode u obradi digitalnih slika
irhad.saric@hotmail.com

Abstract – U današnje vrijeme skoro sve fotografije su u boji. Međutim, ponekad nam se javi potreba da te fotografije konvertujemo u fotografiju u nijansama sive boje, npr. kada je želimo isprintati isl. Takve fotografije se lahko daju prebaciti u binarne, koje se najčešće koriste u industrijskim aplikacijama, kod detekcije ivica, itd. U ovom radu će biti opisan način na koji se vrši pretvaranje fotografije u boji u sivoskaliranu fotografiju.

I. UVOD

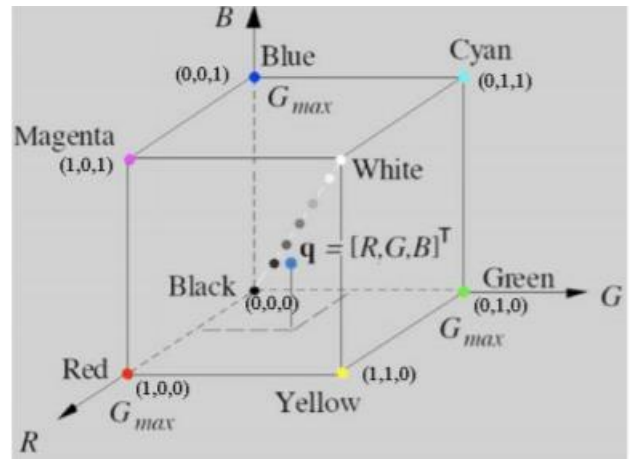
Poznato je da se kod binarnih slika koristi jedan bit za predstavljanje jednog piksela. Kako bit ima svega dva stanja, 0 i 1, to znači da je svaki piksel u jednoj od dvije predefiniране boje, a najčešće su to crna i bijela. Nemogućnost prikazivanja međunijansi sive je veliko ograničenje za binarne slike u prikazivanju fotografija. Slike u nijansama sivog (eng. Grayscale), koje se često nazivaju i crno-bijelim slikama, salinjene su od piksela čije boje su različiti intenziteti sivog, u rasponu od bijele do crne boje. Najčešće se radi o 256 nivoa intenziteta sivog, pa je za prikaz jednog piksela potreban jedan bajt. Imajući u vidu da je eksperimentima pokazano da ljudsko oko može da razlikuje najviše 200 nijansi sivog, 256 nivoa je i više nego dovoljno da se prikaže pun spektar između bijele i crne boje bez uočljivih prelaza, pa je ovaj tip digitalnih slika pogodan za prikaz fotografija.

Slike u boji čine pikseli koji pored informacija o intenzitetu sadrže i informacije o boji. Za vizuelno prihvatljive rezultate, neophodne su tri vrijednosti, odnosno kanala, koji opisuju boju svakog piksela u određenom prostoru boja. Najčešće korišteni prostor boja je RGB, o kojem će biti govora u nastavku, ali u upotrebi su i drugi prostori, kao što su CMY, CMYK, HSV, YCbCr.

II. PROSTORI BOJA

A. RGB prostor boja

Koordinatne ose (R, G, B) predstavljaju tri osnovne boje: crvena, zelena i plava. Maksimum osvjetljenja u bilo kojoj boji je 255, a minimum 0. Na slici gore smo umjesto 255 pisali 1, jer smo interval (0,255) skalirali na (0,1). Ukoliko je $G_{max} = 255$ odnosno $G_{max} = 1$, tada imamo 16 777 216 različitih boja, pri čemu uređena torka (255,0,0) predstavlja crvenu, (255,255,0) žutu itd. Na glavnoj dijagonali kocke na Slici 1. nalaze se sivoskalirane (ahromatske) nijanse (eng. grayscale).



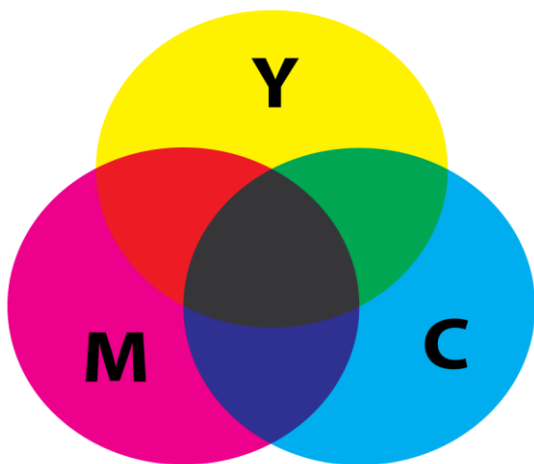
Slika 1. RGB kocka u Dekartovom koordinatnom sistemu

RGB je osnova ekranskog prikazivanja. Na primjer, monitori su premazani sa fosfornim zrcima koja daju crveno, zeleno i plavo osvjetljenje. RGB je aditivni sistem, što znači da se sve boje dobijaju dodavanjem različitih intenziteta tri primarne komponente. Crna boja predstavlja odsustvo sve tri komponente, dok se bijela dobija kombiniranjem sve tri komponente maksimalnih intenziteta, što je i rezultat Newtonovog eksperimenta, koji pokazuje da se bijela boja dobija kao puno osvjetljenje u svim osnovnim bojama.

Memorijski zapis podataka u RGB modelu zahtjeva $N \times 3 \times k$ (piksela \times boje \times bita za zapis svake boja). Najčešće je $k = 8$, ali postoje i drugi modeli. Stoga, RGB zapis je memorijski jako zahtjevan.

B. CMY prostor boja

CMY je substraktivni sistem, koji se koristi kod uređaja koji ne emitiraju svjetlost, već se zasnivaju na refleksiji, kao što su npr. štampači. Osnovne komponente su: cijan, magenta i žuta (eng. cyan, magenta, yellow). Cijan komponenta apsorbira crvenu svjetlost, ali reflektira zelenu i plavu. Magenta apsorbira zelenu svjetlost i reflektira crvenu i plavu. Žuta apsorbira plavu, a reflektira crvenu i zelenu svjetlost, što se vidi na Slici 2.



Slika 2. CMY prostor boja

Prelazak iz RGB u CMY model realizira se na osnovu sljedećih relacija:

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$C = 1 - R'$$

$$M = 1 - G'$$

$$Y = 1 - B'$$

C. CMYK prostor boja

Nedostatak CMY modela je nemogućnost reprodukcije velikog broja boja. Teško se reproduciraju drečave i fluorescentne nijanse, kao i crna boja. Svega oko milion boja je moguće dobro reproducirati. Najveći problem je crna boja na koju su ljudi veoma osjetljivi.

Stoga se osnovni model CMY proširuje sa crnom bojom, pa nastaje CMYK čime je omogućeno reproduciranje tamnijih nijansi. Prelazak iz CMY u CMYK data je relacijama ispod:

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K \end{pmatrix} = \begin{pmatrix} C - K \\ M - K \\ Y - K \\ K \end{pmatrix}$$

$$K = \min\{C, M, Y\}$$

Kako RGB i CMYK zavise od uređaja koji reproducira boje određenog modela, relacije za konverziju nisu jedinstvene, već se one definiraju uz pomoć profila boja.

III. OPIS RJEŠENJA

A. Metoda kvadratnog korijena

Kako sivoskalirane nijanse predstavljaju dijagonalu kolorne kocke, konverzija kolornih nijansi u odgovarajuće sivoskalirane nijanse obavlja se na sljedeći način:

$$gray = \sqrt{R^2 + G^2 + B^2}$$

B. Tehnika usrednjavanja

Kako je kvadratni korijen zahtjevnja operacija, i kako se rezultat mora kvantizirati u odgovarajući broj nivoa, ovaj način konverzije se izbjegava, jer na kraju se njime dobija slika koja nije suviše realistična odnosno prirodna. Bolji i jeftiniji način jeste da se konverzija obavi tehnikom usrednjavanja, tj:

$$gray = \frac{R + G + B}{3}$$

Ponekad se koriste jednostavnije šeme za konverziju, kod kojih se sivoskaliranim kanalom proglašava crveni ili zeleni kanal, npr. $gray = R$ ili $gray = G$. Plavi kanal se veoma rijetko koristi zato što nije realističan. Međutim, postoje još bolji načini za konverziju slike u boji u sivoskaliranu sliku.

C. Standardi

Bolja opcija za promjenu slike u boji u sivoskaliranu sliku je ITU-R Recommendation BT.601-7 standard koji definiše metode za digitalno kodiranje video signala normalizujući vrijednosti. Za konverziju slike u boji u sivoskaliranu sliku je definisana sljedeća formula:

$$gray = 0.299 * R + 0.587 * G + 0.114 * B$$

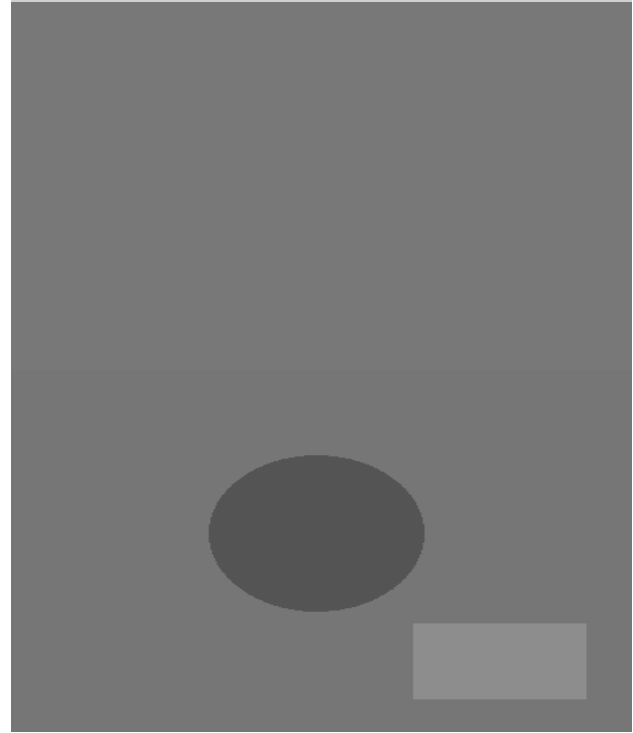
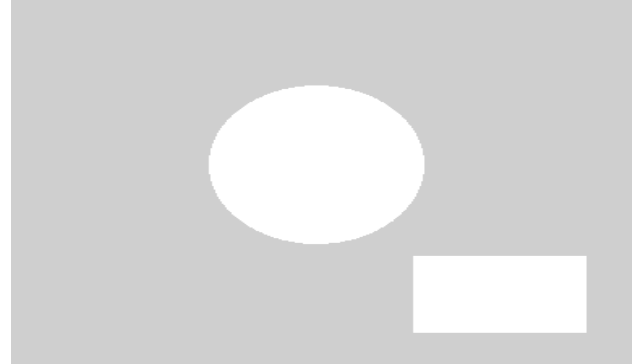
Za realizaciju konverzije je korišten Pillow, modifikacija Pythonove biblioteke PIL (eng. Python Imaging Library). Prije nego počnemo sa manipulacijom slike, moramo biti u mogućnosti otvoriti sliku, sačuvati promjene, kreirati praznu sliku biti u mogućnosti izvući svaki piksel iz slike. Ove metode su napravljene uz pomoć metoda iz biblioteke Pillow. Bitno je napomenuti da funkcija za uzimanje piksela sa slike vraća piksel u RGB formatu.

IV. ANALIZA RJEŠENJA



Slika 3. Primjer rezultata algoritma

Na Slici 3. vidimo rezultate koje nam daju redom algoritmi korijena, aproksimacije i rezultat korištenja standard BT.601-7. Očigledno je da korijeni algoritam daje slabije rezultate od preostala dva, tj. gube se mnogi detalji i slika je drastično svjetlija nego što bi trebala biti. Međutim razlika između aproksimativnog algoritma i korištenja standarda nije očita uopšte. Da bismo razjasnili razliku između ova dva algoritma, posmatrat ćemo sljedeći primjer.



Slika 4. Primjer rezultata algoritma

Na Slici 4. Primjećujemo razliku između svih algoritama. Naime na originalnoj slici su prikazana dva objekta koji imaju jednaku aritmetičku sredinu kao i pozadina. Korijeni algoritam uspijeva izdvojiti geometrijske objekte, ali im ne razlikuje boje, dok tehnikom usrednjavanja algoritam uopšte ne prepoznaje razliku u boji.

V. ZAKLJUČAK

U ovom radu smo vidjeli neke prostore boja i na koji način se slike zadate u jednom prostoru mogu konvertovati u drugi isl. Također smo vidjeli tri algoritma za konvertovanje slike u boji u sivoskaliranu sliku. Također smo vidjeli u čemu se razlikuju ti algoritmi kroz slikovite prikaze rješenja koje nam je algoritam, koji je napravljen u Pythonu pomoću biblioteke Pillow, vratio.

REFERENCES

- [1] Asia Mahdi Naser, "Color to Grayscale Image Conversion Based Dimensionality Reduction with Stationary Wavelet Transform," University of Karbala, College of Science, May 2016.
- [2] C. Saravanan, Ph.D., Assistant Professor, "Color Image to Grayscale Image Conversion", Computer Centre, National Institute of Technology, Durgapur, West Bengal, IndiaK. Elissa, "Title of paper if known," unpublished.
- [3] R. Nicole, "A Novel Framework for Optimal RGB to Grayscale Image Conversion" Yi Wan, Qisong Xie.
- [4] Adis Alihodžić, "Matematske metode u obradi I vizualizaciji digitalne slike,"
- [5] <https://www.codementor.io/isaib.cicourel/image-manipulation-in-python-du1089j1u>