

*Edin Tabak, prof.
Dr. sc. Amira Šerifović-Trbalić*

REPREZENTACIJA DIGITALNE SLIKE PROGRAMSKIM PAKETOM MATLAB

Sažetak

Digitalna slika nam pruža veliku mogućnost manipulacije njenim elementima, i samim tim nad transformacijom slike. U ovom radu će biti prikazan način na koji je digitalna slika predstavljena u matematičkom obliku preko matrica i šta predstavlja svaki piksel naše digitalne slike. Pošto sliku možemo predstaviti u obliku matrice, onda je lako vršiti njene transformacije dok se nalazi u matričnom obliku. Pokazat ćemo osnove funkcije u programskom paketu MATLAB za učitavanje slike i kreiranje matrice, prikaz slike, i na kraju snimanje slike sa promjenama koje smo napravili. Također, biti će prikazane funkcije za prikaz informacija slike, kao i za promjenu piksela u crno-bijelu kombinaciju.

Ključne riječi: *digitalna slika, matlab, piksel, matrica, uzorkovanje, kvantizacija, rezolucija*

DIGITAL IMAGE REPRESENTATION WITH SOFTWARE PACKAGE MATLAB

Summary

Digital image gives us huge possibilities for manipulating its elements, as well as image transformation. This paper shows how a digital image is represented in mathematical form with matrices and what represents every pixel of digital image. Since we can represent digital images with matrices, we can easily do its transformations while it is in matrix form. Paper will show basic functions in software package MATLAB for reading digital image and creating matrices,

display digital image and at the end for saving digital image together with changes we made. Also we will show functions for digital image information, as well as changing all pixels of our digital image to black-white combination.

Key words: *digital image, matlab, pixel, matrix, sampling, quantization, resolution*

Uvod

U svakodnevnom životu koristimo digitalne slike, pregledamo ih, objavljujemo ih na internetu, a nerijetko vršimo i neku vrstu obrade slike. Ta obrada se može zasnivati na posvjetljivanju, povećavanju kontrasta slike, izdvajanje dijelova slike, izoštravanju slike, retuširanju slike itd.

Postoje već gotovi alati za obradu slike kao što su Adobe Photoshop, GIMP, Paint Shop Pro i mnogi drugi, koji većinom imaju iste funkcije za obradu slike. Kada vršimo ove promjene nas ne interesuje na koji način to radi alat za obradu slike i zašto je to moguće. Sve ove operacije sa slikama su moguće, jer sliku možemo da prevedemo u matematički model, tj. u matricu, preko njenih piksela i da onda vršimo promjene nad svakim pojedinačnim pikselom.

U ovom radu ćemo vidjeti kako se slika predstavlja u obliku matrice i kako je moguće vršiti operacije i promjene nad tom slikom, te kako u tu svrhu koristiti programski paket MATLAB. Upoznat ćemo se sa funkcijama za učitavanje slika, prikaz učitanih i izmjenjenih slika i snimanje slika u MATLAB-u.

1. Reprezentacija digitalne slike

Sliku možemo definisati kao dvodimenzionalnu funkciju $f(x, y)$, gdje x i y predstavljaju koordinate, a vrijednost funkcije f u bilo kojem paru koordinata (x, y) predstavlja intenzitet slike u toj tački. Termin *sivi nivo* se često koristi za intenzitet crno-bijelih slika. Slike u boji se formiraju na način da se posebno daju vrijednosti intenziteta tri komponente izabranog modela boja. Na primjer, slika u RGB (Red Green Blue) sistemu, sastoji se od tri individualne komponente slike. Zbog ovoga se bilo koja metoda za obradu crno-

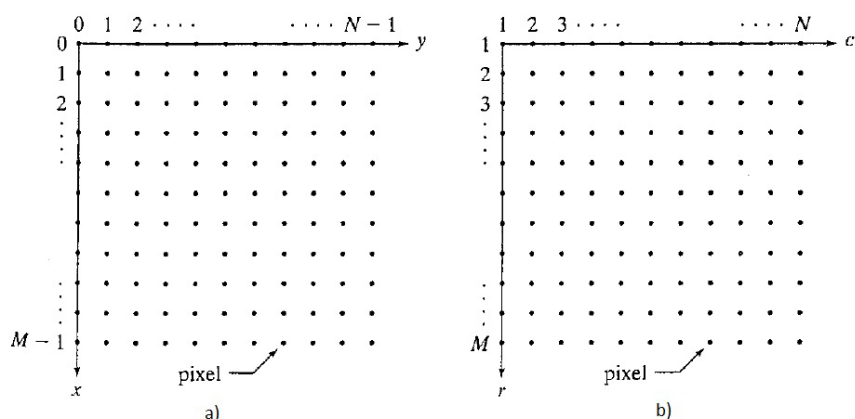
bijelih slika može proširiti i na slike u boji, primjenom metoda na svaku komponentnu zasebno.

Slika može biti neprekidna u odnosu na x i y koordinate, kao i na vrijednost funkcije f . Pretvaranje takve slike u digitalnu formu zahtijeva da se sve koordinate, kao i vrijednost funkcije, digitalizira. Digitalizacija koordinata se naziva uzorkovanje (engl. sampling), a digitalizacija vrijednosti funkcije se naziva kvantizacija. Prema tome, kad su koordinate x i y , i vrijednost funkcije f , konačne diskretne veličine, tu sliku nazivamo digitalna slika. Onda možemo da vršimo obradu nad tim konačnim diskretnim veličinama, te na taj način vršimo izmjene i nad samom slikom.

1.2. Pravila koordinata

Rezultat uzorkovanja i kvantizacije je matrica realnih brojeva. Možemo koristiti više načina za prikaz digitalnih slika u matricama. Mi ćemo sada navesti dva načina za prikaz matrice. Pretpostavimo da nam je rezultat uzorkovanja i kvantizacije digitalna slika prevedena u matricu koja ima M redova i N kolona. Onda ćemo reći da je naša slika dimenzije $M \times N$. Vrijednosti koordinata (x, y) su diskretne veličine. Ako nam je početna koordinata $(x, y) = (0, 0)$, naredna koordinata u istom redu bi bila $(x, y) = (0, 1)$. Na taj način naša bi slika imala x u rasponu od 0 do $M - 1$, i y u rasponu od 0 do $N - 1$ (Slika 1.a.).

U MATLAB-u se za koordinate koristi format (r, c) , što dobijamo od engleskih riječi *row* – red i *columns* – kolone. Također razlika je u tome što početni piksel ima koordinate $(r, c) = (1, 1)$, pa onda naša slika ima raspon za r od 1 do M , i za c od 1 do N (Slika 1.b.).



Slika 1. Pravila koordinata u koordinatnom sistemu

1.3. Slike kao matrice

Reprezentacija funkcije digitalne slike je sljedeća:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix}$$

Desna strana ove matrice je digitalna (rasterska) slika po definiciji. Svaki element ove matrice se naziva *element slike* ili *piksel*.

Digitalna slika predstavljena u MATLABU preko matrice ima oblik

$$f = \begin{bmatrix} f(1, 1) & f(1, 2) & \dots & f(1, N) \\ f(2, 1) & f(2, 2) & \dots & f(2, N) \\ \vdots & \vdots & & \vdots \\ f(M, 1) & f(M, 2) & \dots & f(M, N) \end{bmatrix},$$

gdje je $f(1, 1) = f(0, 0)$. Kao što vidimo ove dvije reprezentacije su iste, samo se razlikuju u pomaku koordinata. Zapis $f(6, 2)$ nam govori da se dati piksel nalazi u šestom redu i drugoj koloni matrice f . Obično se slova M i N koriste da bi definisali broj redova i kolona u matrici. Matrica oblika $1 \times N$ naziva se *vektor reda*, a matrica $M \times 1$ se naziva *vektor kolone*. Matrica oblika 1×1 naziva se *skalar*, što ustvari predstavlja jedan piksel.

Iako slike posmatramo kao diskretne konačne matrice, nekad je potrebno posmatrati slike i kao neprekidne prostorne signale. Ovo se radi zbog većeg broja mogućnosti za rad sa samim osobinama slike, jer nam omogućava da koristimo snažne tehnike za integralni i diferencijalni račun. Matematička analiza diskretnih slika, generalno vodi do linearnih algebarskih formula koje su u nekim slučajevima bolje.

1.4. Rezolucija slike

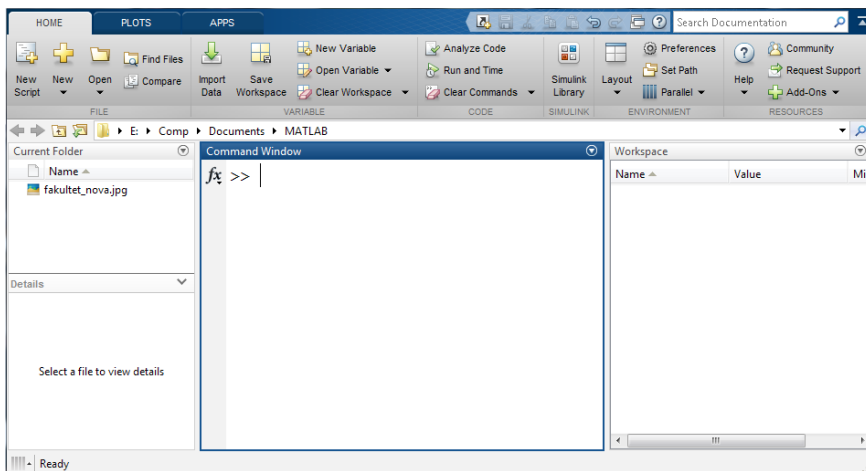
Veličina matrice zajedno sa podacima snimljenim u svakom pikselu određuje rezoluciju slike i kvantizaciju boje. Veličina slike definira se rezolucijom. Rezolucija slike može biti prikazana na jedan od sljedećih načina:

- *Prostorna rezolucija* – Broj redova i kolona naše matrice definišu broj piksela koji je potreban da se pokrije virtualni prostor slike. Ovo se odnosi na uzorkovanje slike i ponekad pokazuje pikselnu ili digitalnu rezoluciju slike. Pa tako možemo da vidimo neke od rezolucija slike: $640 \times 400, 800 \times 600, 1024 \times 768$ itd.
- *Vremenska rezolucija* – Koristi se za neprekidne signale kao što je video, i odnosi se na broj slika snimljenih u periodu vremena. Obično se računaju slike po sekundi (frames per second – fps). TV operateri rade na 25 fps, 25 – 30 fps se koristi za video nadzor, dok se kamere veće vremenske rezolucije koriste u naukama.
- *Bitna rezolucija* – Definiše broj mogućih intenziteta/vrijednosti boje koju piksel može imati i odnosi se na kvantizaciju informacije slike, a naziva se i “dubina boje (color depth)”. Na primjer binarna slika ima samo dvije boje (crnu i bijelu), dok slika na bazi sive ima 256 različitih sivih nivoa koji se nalaze između crne i bijele. Bitna rezolucija se odnosi na broj bitova koji su potrebni sa kvantizacijski nivo.

2. Rad sa slikom u MATLABU

MATLAB je okruženje za numeričke proračune i programski jezik koji proizvodi firma MathWorks. MATLAB omogućava lako manipulisanje matricama, prikazivanje funkcija i fitovanje, implementaciju algoritama, stvaranje grafičkog korisničkog interfejsa kao i povezivanje sa programima pisanim u drugim jezicima.

MATLAB je nastao kao skraćenica za „MATrix LABoratory“ („laboratorija za matrice“). Izumio ga je kasnih 1970.-ih Cleve Moler, šef katedre za informatiku na Univerzitetu „Novi Meksiko“. Procjena je da preko milion ljudi zaposlenih u industriji i na univerzitetima koristi MATLAB. MATLAB je moguće nabaviti zajedno sa Simulink-om kao i brojnim dodacima.



Slika 2. Korisnički interfejs MATLAB-a

Sada ćemo vidjeti na koji način možemo da radimo sa slikom u programskom paketu MATLAB. Obradit ćemo funkcije za učitavanje slike, prikazivanje slike i snimanje slike.

2.1. Učitavanje slike

Slike učitavamo u MATLAB okruženje koristeći funkciju *imread* čija je sintaksa sljedeća:

```
>> imread('filename').
```

Ovdje *'filename'* predstavlja kompletno ime datoteke sa slikom, uključujući i ekstenziju datoteke. Na primjer komanda

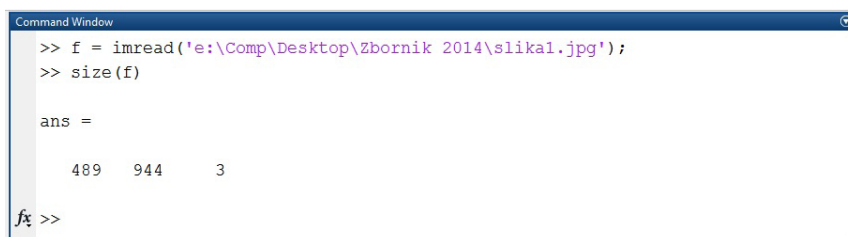
```
>> f = imread('e:\Comp\Desktop\Zbornik 2014\slika1.jpg');
```

predstavlja učitavanje slike *slika1.jpg* u matricu *f*. Ovu sliku učitavamo iz foldera 'Zbornik 2014' koji se nalazi na Desktopu. Oznake >> nam pokazuju gdje je početak naše komandne linije, dok tačka-zarez na kraju onemogućava MATLAB da nam odmah prikaže i rezultat.

Pomoću funkcije *size* možemo da vidimo broj redova i kolona matrice, to jest da vidimo koja je rezolucija slike:

```
>> size(f)
```

Sada nismo stavili tačku-zarez na kraj funkcije, jer želimo da vidimo rezultat naše funkcije.



```
Command Window
>> f = imread('e:\Comp\Desktop\Zbornik 2014\slika1.jpg');
>> size(f)

ans =

    489    944     3

fx >>
```

Slika 3. Prikaz komandnog prozora sa komandama *imread* i *size*

Kao što vidimo naša slika ima 489 redova i 944 kolone piksela. Ako želimo da vidimo još više informacija o našoj slici, možemo da koristimo funkciju *whos*, koja nam prikazuje i veličinu slike u bajtima.

2.2. Prikaz slike

Slike u MATLAB-u prikazujemo uz pomoć funkcije *imshow* koja ima sintaksu:

```
>> imshow(f, G),
```

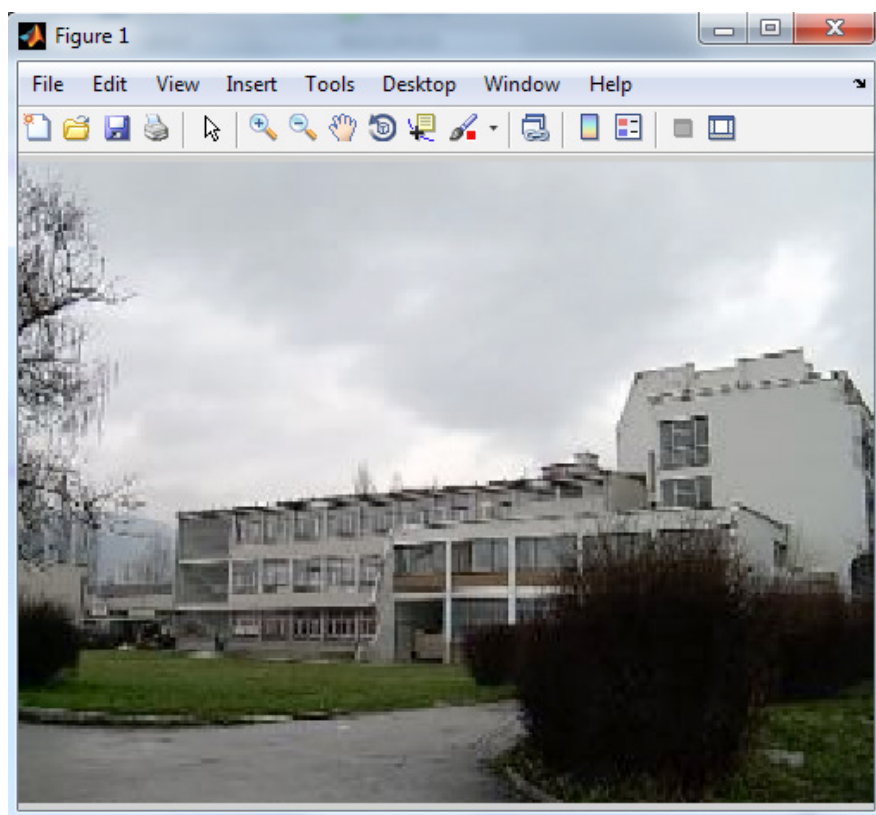
gdje je *f* matrica slike, a *G* predstavlja broj nivoa intenziteta korištenih za prikaz slike. Ako nismo upisali broj *G*, onda on iznosi 256 nivoa. Koristeći sintaksu

```
>> imshow(f, [low high]),
```

prikazujemo kao crne sve vrijednosti manje ili jednake vrijednosti *low*, i kao bijele sve vrijednosti veće ili jednake vrijednosti *high*. Brojevi između su prikazani kao

srednji inteziteti vrijednosti početnih brojevnih nivoa. Sada ćemo na primjeru da prikazemo komandu *whos* i komandu *imshow*.

```
Command Window
>> f=imread('e:\Comp\Desktop\Zbornik 2014\fakultet.jpg');
>> whos f
    Name          Size          Bytes   Class   Attributes
    f             225x300x3         202500  uint8
>> imshow(f)
```



Slika 4. Prikaz slike funkcijom *imshow*

Broj slike se nalazi na vrhu, pod nazivom Figure 1. Kao što vidimo, imamo razne menije i tipke za podešavanje slike. Možemo da zumiramo, snimamo ili eksportujemo sadržaj prikaza. U suštini, Edit meni ima funkcije za editovanje i formatiranje rezultata prije nego što ga printamo ili snimimo na računar.

Ako mi koristimo ponovo funkciju *imshow* na neku novu sliku snimljenu u matricu *g*, MATLAB će tu sliku otvoriti preko naše prethodne slike *f* koju smo prikazali. Da bismo to izbjegli i vidjeli u prikazu obje slike koristimo sintaksu

```
>> imshow(f), figure, imshow(g),
```

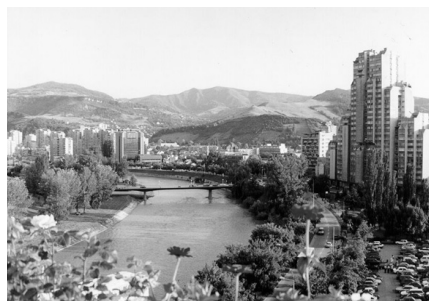
i na taj način ćemo prikazati obje slike.

2.3. Funkcija *rgb2gray*

Još jedna funkcija koju ćemo navesti je *rgb2gray*, a čija sintaksu glasi

```
>> g = rgb2gray(rgb);
```

gdje *g* predstavlja novu sliku sa izmjenjenim pikselima u nijansu sive.



Slika 5. Primjer *rgb2gray* funkcije: izvorna slika (lijevo) i slika u sivoj nijansi (desno)

Piksel ima vrijednost boja u RGB sistemu (r, g, b) , gdje r, g, b predstavljaju cjelobrojne vrijednosti na intervalu $[0, 255]$.

Funkcija *rgb2gray* konvertuje RGB vrijednosti pomoću formule

$$x = 0.299r + 0.587g + 0.114b,$$

u odgovarajuću nijansu

sive, tj. pikselu dodijeli vrijednost (x) . Kao što vidimo, boje nisu raspoređene na isti način. To je zato, jer je čista zelena boja svjetlija od čiste crvene i plave boje. Čista plava boja je najtamnija u ovom sistemu, pa ima i najmanji koeficijent.

Slika 6. Vrijednost piksela u *RGB* sistemu

Ukoliko u formulu za računanje vrijednosti ‘ x ’ uvrstimo vrijednost piksela sa slike 6., sa iznosima $r = 175$, $g = 187$ i $b = 87$, dobit ćemo sljedeće:

$$x = 0.299 \cdot 175 + 0.587 \cdot 187 + 0.114 \cdot 87$$

$$x = 172,012.$$

Vrijednost **172,012** zaokružujemo na prvu cjelobrojnu vrijednost, što iznosi **172**. Pikel sada ima vrijednost **(172)**, što predstavlja nijansu sive.

Izvorna vrijednost piksela (175, 187,87)



Novodobijena vrijednost piksela (172)



2.4. Snimanje slika

Slike se snimaju na računar uz pomoć funkcije *imwrite*, koja ima sintaksu

```
>> imwrite(f, 'filename').
```

Snimanje naše slike na ovaj način, slike koju smo prethodno učitali u matricu *f* i možda izmjenili, zahtijeva da *filename* sadrži prepoznatljivu ekstenziju za slike. Postoje dva načina snimanja slika na računar i to:

```
>> imwrite(g, 'fakultet_nova', 'jpg'),
```

ili ako stavimo ekstenziju zajedno sa nazivom slike

```
>> imwrite(g, 'fakultet_nova.jpg').
```

Imwrite funkcija može da ima i druge parametre u zavisnosti formata u kojem snimamo našu sliku. Ako sliku snimamo u .jpg formatu možemo da primijenimo atribut *quality* sa sintaksom

```
>> imwrite(g, 'fakultet_nova.jpg', 'quality', q),
```

gdje *q* predstavlja broj iz intervala **[0,100][0,100]** (što je manji broj, to je veća degradacija tokom JPEG kompresije).

Da bismo vidjeli koliku smo kompresiju uspjeli da dobijemo snimanjem naše slike koristeći atribut *quality*, i također da dobijemo ostale korisne informacije o našoj slici, možemo da koristimo funkciju

```
>> imfinfo filename,
```

gdje *filename* predstavlja kompletno ime naše slike snimljene na disku.

```
Command Window
>> imwrite(f, 'fakultet_nova.jpg', 'quality', 15);
>> imfinfo fakultet_nova.jpg

ans =

    Filename: 'E:\Comp\Documents\MATLAB\fakultet_nova.jpg'
    FileModDate: '28-Oct-2014 01:30:58'
    FileSize: 3996
    Format: 'jpg'
    FormatVersion: ''
    Width: 300
    Height: 225
    BitDepth: 24
    ColorType: 'truecolor'
    FormatSignature: ''
    NumberOfSamples: 3
    CodingMethod: 'Huffman'
    CodingProcess: 'Sequential'
    Comment: {}
```

Slika 7. Informacije na snimljenoj slici 'fakultet_nova.jpg'

Zaključak

Kako često koristimo slike i vršimo transformacije nad istima, bilo bi dobro da znamo na koji način se to ostvaruje. Obično samo otvorimo alat za uređivanje slika i vršimo transformacije bez da moramo znati kako se to realizuje. Ako želimo malo da eksperimentišemo, trebalo bi da znamo gdje to možemo uraditi. Snimanje slike u matricu i manipulacijom elemenata matrice (pikseli slike), možemo da dobijemo interesantne rezultate. Onda kada shvatimo na koji način se to ostvaruje, lakše ćemo da ostvarimo transformacije i uljepšavanje slike kada to poželimo.

Ovaj rad je dao samo uvod u reprezentaciju slike u programskom paketu MATLAB-u, i samo par funkcija za operacije nad slikama. Mnoge ostale funkcije je moguće naći u raznovrsnoj literaturi i na internetu. Sve što je potrebno je malo poznavanja računskih operacija nad matricama i malo mašte.

Literatura

Gonzales R., Woods R. (2002). Digital Image Processing 2nd Edition, Prentice Hall, USA

Gonzales R., Woods R., Eddins S. (2009). Digital Image Processing 2nd Edition, Gatesmark Publishing, USA

McAndrew A. (2011), An Introduction to Digital Image Processing with Matlab, School of Computer Science and Mathematics, Victoria University of Technology

Solomon C., Breckon T. (2011). Fundamentals of Digital Image Processing, A practical Approach with Examples in Matlab, John Wiley & Sons, Ltd, UK

The Math Works Inc., (1997). Image Processing Toolbox for use with MATLAB, Natick, USA

Thyagarajan K. S., (2011). Still Image and Video Compression with Matlab, John Wiley & Sons, Ltd, UK

Internet:

<http://www.mathworks.com/help/matlab/>

Recenzent: dr. sc. Samir Lemeš, UNZE, BiH

Lektor: autori rada