

# **Digitalna obrada slike uz pomoć MatLaba**

## **Zadaci i rješenja**

## Zadatak br 1.

Učitati RGB sliku, pa isprintati u posebnim fajlovima njezinu crvenu, zelenu i plavu komponentu.

Rj.

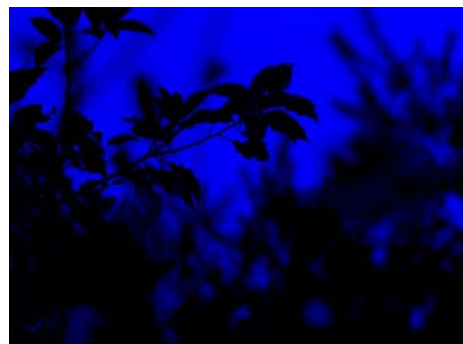
Korističu sljedeću funkciju, napravljenu samo za ovaj primjer (komponenta.m):

```
function [ boja ] = komponenta( slika, RGB )
%% =====
%% KOMPONENTA vraća crvenu, zelenu ili plavu komponentu slike
%% Objašnjenje: function [ boja ] = KOMPONENTA( slika, RGB )
%% slika = ime varijable koja sadrži RGB sliku
%% RGB = neko od slova R, G ili B, redom crvena, zelena ili plava
%% boja = slika koja je rezultat
%% =====
boja=slika;
vel_u_pikselima = size(slika);
sirina=vel_u_pikselima(1);
visina=vel_u_pikselima(2);

if RGB == 'R'
    for i=1:1:sirina
        for j=1:1:visina
            boja(i,j,2)=0;
            boja(i,j,3)=0;
        end
    end
end

if RGB == 'G'
    for i=1:1:sirina
        for j=1:1:visina
            boja(i,j,1)=0;
            boja(i,j,3)=0;
        end
    end
end

if RGB == 'B'
    for i=1:1:sirina
        for j=1:1:visina
            boja(i,j,1)=0;
            boja(i,j,2)=0;
        end
    end
end
```



Kucamo redom:

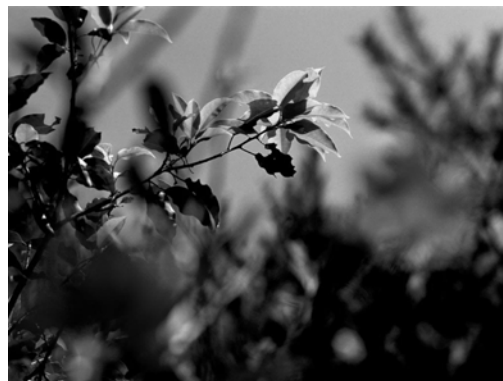
```
% učitavam sliku nature01.jpg u varijablu slika
>> slika=imread('nature01.jpg');
% pravimo crvenu, zelenu i plavu komponentu slike
>> crvena=komponenta(slika,'R');
>> zelena=komponenta(slika,'G');
>> plava=komponenta(slika,'B');
% prikazi crvene, zelene i plave komponente slike
>> figure(1); image(crvena);
>> figure(2); image(zelena);
>> figure(3); image(plava);
% snimamo komponente u fajlove
>> imwrite(crvena,'nature01_crvena.jpg','jpg');
>> imwrite(zelena,'nature01_zelena.jpg','jpg');
>> imwrite(plava,'nature01_plava.jpg','jpg');
```

### Zadatak broj 2.

RGB sliku pretvoriti u crnobiljelu, pa rezultat snimiti u zaseban fajl.

Rj.

```
>> slika=imread('nature01.jpg');  
  
% NTSC standard (uzeto iz helpa)  
>> komponente=.2989*slika(:,:,1)+.5870*slika(:,:,2)+.1140*slika(:,:,3);  
  
>> crnobiljela(:,:,1)=komponente;  
>> crnobiljela(:,:,2)=komponente;  
>> crnobiljela(:,:,3)=komponente;  
>> figure(3); image(crnobiljela);  
>> imwrite(crnobiljela,'nature01_crnobiljela.jpg','jpg');
```



### Zadatak broj 3.

Napraviti vertikalni obrat RGB slike.

Rj.

Koristiću sljedeću funkciju (vert\_flip.m):

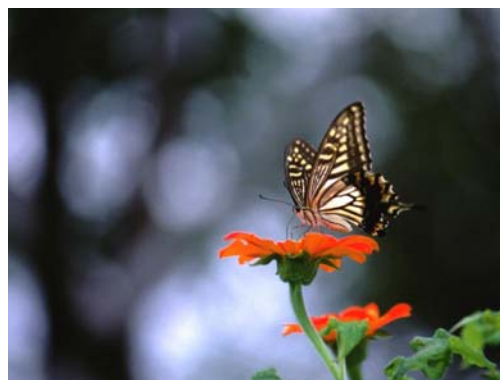
```
function [ vrac_slika ] = vert_flip( slika )  
%% =====  
%% VERT_FLIP pravi vertikalni flip slike  
%% Objašnjenje: function [ vrac_slika ] = VERT_FLIP( slika )  
%% slika = ime varijable koja sadrži RGB sliku  
%% vrac_slika = je rezultat  
%%=====
```

```
[sirina, visina, dubina]=size(slika);  
for k=1:1:dubina  
    for i=1:1:sirina  
        for j=1:1:visina  
            vrac_slika(i,j,k)=slika(i,visina-j+1,k);  
        end  
    end  
end
```



Kucamo redom:

```
>> slika=imread('nature05.jpg');  
>> ver_slika=vert_flip(slika);  
>> figure(1); image(slika);  
>> figure(2); image(ver_slika);  
>> imwrite(ver_slika,'nature05_ver_flip.jpg','jpg');
```



#### Zadatak broj 4.

Napraviti horizontalni flip slike.

Rj.

Koristim sljedeću funkciju (hor\_flip.m):

```
function [ vrac_slika ] = hor_flip( slika )
%% =====
%% HOR_FLIP pravi horizontalni flip slike
%% Objašnjenje: function [ vrac_slika ] = HOR_FLIP( slika )
%% slika = ime varijable koja sadrži RGB sliku
%% vrac_slika = je rezultat
%%=====
[sirina, visina, dubina]=size(slika);
for k=1:1:dubina
    for i=1:1:sirina
        for j=1:1:visina
            vrac_slika(i,j,k)=slika(sirina-i+1,j,k);
        end
    end
end
end
```

```
>> slika=imread('mix1.jpg');
>> slika1=hor_flip(slika);
>> figure(2); image(slika);
>> figure(3); image(slika1);
>> imwrite(slika1,'mix1_hor_flip.jpg','jpg');
```



## Zadatak broj 5.

Rotirati sliku za 90, 180 i 270 stepeni.

Rj.

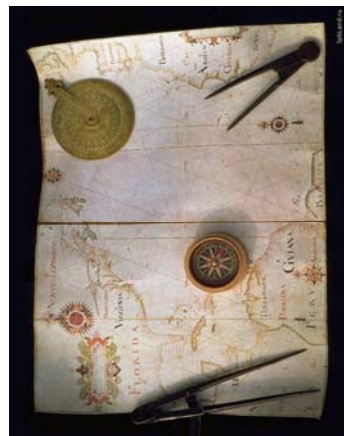
Prvo ćemo napraviti sljedeću funkciju (rotirajza.h):

```
function [ rotirana ] = rotirajza( slika, stepen )
%% =====
%% ROTIRAJZA rotira sliku za 90, 180 ili 270 stepeni
%% Objašnjenje: function [ rotirana ] = ROTIRAJZA ( slika, stepen )
%% slika = slika koju želimo rotirati
%% stepen = neki od brojeva 90, 180 ili 270
%% rotirana = rotirana slika
%% =====
[sirina, visina, dubina]=size(slika);

if stepen == 90
for k=1:1:dubina
    for i=1:1:sirina
        for j=1:1:visina
            rotirana(j,i,k)=slika(i,visina-j+1,k);
        end
    end
end
end

if stepen == 270
for k=1:1:dubina
    for i=1:1:sirina
        for j=1:1:visina
            rotirana(j,i,k)=slika(sirina-i+1,j,k);
        end
    end
end
end

if stepen == 180
for k=1:1:dubina
    for i=1:1:sirina
        for j=1:1:visina
            rotirana(i,j,k)=slika(sirina-i+1,visina-j+1,k);
        end
    end
end
end
end
end
```



Kucamo redom:

```
>> slika=imread('mix1.jpg');
>> slika1=rotirajza(slika,90);
>> slika2=rotirajza(slika,180);
>> slika3=rotirajza(slika,270);
% naredba 'axis image;' postavlja kvadratni izgled piksela
>> figure(2); image(slika1); axis image;
>> figure(3); image(slika2); axis image;
>> figure(4); image(slika3); axis image;
>> imwrite(slika1,'mix1_90.jpg','jpg');
>> imwrite(slika2,'mix1_180.jpg','jpg');
>> imwrite(slika3,'mix1_270.jpg','jpg');
```

### Zadatak broja 6.

Sliku u RGB formatu konvertovati u index sliku sa 256 boja.

```
>> slika=imread('mix2.jpg');
>> [ind_slika,map] = rgb2ind(slika, 256);
>> ind_slika=double(ind_slika)+1;
>> imwrite(ind_slika,map,'mix2_512.gif','gif');
```

### Zadatak broj 7.

Dat je filter:

$h(m,n)=1/81$  za  $m<5$  i  $n<5$ ,

$h(m,n)=0$  u ostalim varijantama.

Primjeniti ovaj filter na učitanoj slici.

Rj

```
>> slika=imread('nature16.jpg');
>> [indsl,mapa]=rgb2ind(slika,16);
>> indsl=double(indsl)+1;
>> figure(1); imshow(indsl,mapa); axis image;
>> filter=[1/81, 1/81, 1/81, 1/81; 1/81, 1/81, 1/81, 1/81; 1/81, 1/81, 1/81, 1/81; 1/81, 1/81, 1/81, 1/81]
```

filter =

```
0.0123 0.0123 0.0123 0.0123
0.0123 0.0123 0.0123 0.0123
0.0123 0.0123 0.0123 0.0123
0.0123 0.0123 0.0123 0.0123
```

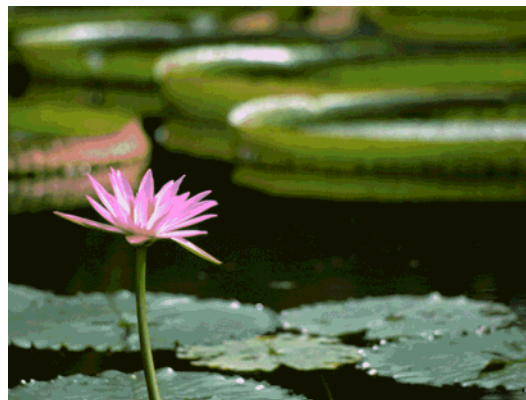
%% PRVI NAČIN

```
>> indsl_filt1=imfilter(indsl,filter);
>> figure(2); imshow(indsl_filt1,mapa); axis image;
```

%% DRUGI NAČIN

```
>> [m, n]=size(indsl);
>> F=fft2(indsl, 2*m, 2*n);
>> H=fft2(filter, 2*m, 2*n);
>> G=H.*F;
>> indsl_filt2=real(ifft2(G));
>> indsl_filt2=indsl_filt2(1:m, 1:n);
>> indsl_filt2=uint8(indsl_filt2);
>> figure(3); imshow(indsl_filt2,mapa); axis image;
```

```
>> imwrite(indsl,mapa,'nature16_ind.gif','gif');
>> imwrite(indsl_filt1,mapa,'nature16_filt1.gif','gif');
>> imwrite(indsl_filt2,mapa,'nature16_filt2.gif','gif');
```



### **Zadatak broj 8.**

Dat je filter:

h =

```
1  0  -1
2  0  -2
1  0  -1
```

Primjeniti ovaj filter na učitanoj slici.

Rj.

```
>> slika=imread('nature09.jpg');
>> [inds1,mapa]=rgb2ind(slika,16);
>> inds1=double(inds1)+1;
>> figure; imshow(inds1,mapa); axis image;
```

```
>> F=fft2(inds1);
>> figure; imshow(F, mapa); axis image;
```

```
>> S=fftshift(log(1+abs(F)));
>> figure; imshow(S, mapa); axis image;
```

```
>> h=[1 0 -1; 2 0 -2; 1 0 -1]
```

h =

```
1  0  -1
2  0  -2
1  0  -1
```