Elektrotehnički fakultet, Banjaluka
Katedra za opštu elektrotehniku
Digitalna obrada slike

Graphical User Interface (GUI) & digitalna obrada slike

- MATLAB implementation -

profesor: dr Zdenka Babić
asistent: mr Vladimir Risojević

studenti:
Igor Kolak, br. indeksa: 118/03
Dragan Šakota, br. indeksa: 007/04

UVOD

U digitalnoj obradi slike koristi se mnogo različitih algoritama pomoću kojih dobijamo odgovarajuće modifikacije polazne slike. Da bismo koristili te algoritme i primjenjivali ih prilikom obrade slike, moramo imati odgovarajuće predznanje. Međutim, "običan" korisnik ne želi da zna detalje izvršavanja nekog procesa na slici, on samo želi krajnji rezultat obrade. Moguće rješenje je svakako GUI (Graphical User Interface).

GUI nam omogućava da napravimo jednostavan interfejs za korisnika na kome će se nalaziti potrebne funkcije implementirane kroz neke menije, dugmiće, textbox-ove, liste,… Izborom opcije u meniju, pritiskom na neko dugme, pomjeranjem klizača, pozivamo odgovarajuću funkciju koja izvršava neki algoritam koji na potreban način djeluje na sliku. Na ovaj način, obrada slike postaje pristupačnija korisniku.

Zadatak ovog semestralnog rada je bio da se pomoću programskog paketa MATLAB napravi jednostavan GUI koji će sadržavati sljedeće funkcije: brisanje, isjecanje, rotiranje za nx90°, slika u ogledalu, negativ slike, dobijanje binarne slike (automatsko/rucno određivanje praga), zum. Sve ove funkcije su sadržane u glavnom meniju (Edit). Ono što ovdje trebamo naglasiti je to da se glavni meni dinamički kreira. To je realizovano tako što postoje dva foldera pod nazivima Edit i Tools, i u njima se nalaze m. fajlovi svih funkcija koje se kasnije dinamički ubace u glavni meni. U folderu Edit se nalaze funkcije koje su realizovane u okviru ovog rada, a u meniju Tools, može se naći bilo koja funkcija čiji se m. fajl nalazi u folderu Tools i koja zadovoljava odgovarajuću sintaksu. Na taj način je omogućena nadogradnja ovog GUI-a, te se bilo koja nova funkcija lako dodaje. Ako želimo da u ovo okruženje ubacimo i funkcije koje su realizovale druge kolege u okviru semestralnih radova iz predmeta Digitalna obrada slike i na taj način kompletiramo projekat, jednostavno bi modifikovali sintaksu funkcija i njihove m. fajlove iskopirali u folder Tools, te na taj način dobili bogatiji i funkcionalniji interfejs.

1. OBJAŠNJENJE IZBORA TEORIJE I TEHNOLOGIJE ZA RJEŠAVANJE PROBLEMA

Za realizovanje zadatka, koristili smo programski paket MATLAB i nejgov GUIDE (GUI Development Enviroment), wizard pomoću koga kreiramo izgled prozora (na kojoj poziciji će se nalaziti učitana slika, gdje će se nalaziti texbox-ovi, kako će izgledati glavni meni, kolika će biti veličina prozora, itd.). Prilikom pokretanja GUIDE-a, kreiraju se dva fajla: GUIname.fig, koji ustvari predstavlja izgled prozora, i GUIname.m, m-fajl u kome se nalaze funkcije koje podešavaju čitav prozor, funkcije koje inicijalizuju GUI izgled kada se otvara prvi put, Callback funkcije (npr. kad kliknemo na neko dugme, mi tada pozivamo neku callback funkciju koja se izvršava u skladu sa napisanim kodom, tj. Callback funkcije su funkcije koje se izvršavaju dok mi djelujemo na grafički interfejs: pomjeranje klizača, izborom opcije u meniju, klik na neko dugme).

Prilikom realizacije zadatka, u cilju jednostavnije nadogradnje i bolje preglednosti, m. fajl koji generiše GUIDE, sadrži samo neke osnovne funkcije. U njemu se nalazi onaj standardni dio koji se odnosi na inicijalizaciju GUI-figure, i nalaze se callback funkcije za otvaranje, zatvaranje i snimanje slike. Ono što je posebno važno naglasiti jeste to da smo ovdje definisali način kreiranja glavnog menija. Glavni meni, pod stavkama Edit i Tools se dinamički kreira. U folderu gdje se nalaze fajlovi .fig i .m, napravili smo dva nova foldera: Edit i Tools, a u njima se nalaze m. fajlovi odgovarajućih funkcija. Nazivi tih m. fajlova se dinamički, u toku pokretanja GUI-a, ubacuju u glavni meni i prilikom izbora neke stavke iz menija, vrši se pozivanje funkcija koje su definisane tim m. fajlovima. Da bi ovo sve funkcionisalo u skladu sa našim GUI-em, potrebno je da funkcije zadovoljavaju određenu sintaksu. Sve ovo omogućava laku nadogradnju postojećeg GUI-a, jer se bilo koja nova funkcionalnost lako dodaje. Sve što je potrebno jeste da se izmjeni sintaksa i da se odgovarajući m. fajl kopira, npr. u folder Tools.

2. DETALJI RJEŠENJA

Zadatak je bio da se napravi jednostavan grafički interfejs sa nekim osnovnim funkcijama za obradu slike. Kao što je već rečeno, sve te funkcije su pisane kao zasebni m.fajlovi, tako da ćemo ih ovdje pojedinačno objašnjavati.

Svaka funkcija mora da zadovoljava određenu sintaksu. U zaglavlju svake funkcije nalazi se sljedeći dio koda:

```
function name(obj,event)
% UNTITLED1 Summary of this function goes here
% Detailed explanation goes here
% handles.data.base_image
handles = guidata(obj);
image = handles.data.base_image;
```

Sada se u ***image*** nalazi učitana slika, a pomoću ***handles*** pristupamo određenim dijelovima koji se nalaze na grafičkom iterfejsu (npr. handles.data.base_image pristupamo našoj slici).

Poslije izvršavanja funkcije polazna slika se modifikuje na određeni način i nama se neće uvijek svidjeti rezultat obrade. Zbog toga se poslije svake obrade slike otvara novi prozor pomoću koga možemo izabrati da li da sačuvamo promjene ili ne. Ovdje ćemo dati dio koda koji ovo omogućava i koji se nalazi u svakoj funkciji:

```
helper = figure;
% DOS_ Don't edit!
tir = menu('Do you want to save changes?','YES','NO');

if (tir == 1)
    handles.data.base_image = image_name;
    guidata(gcbo,handles);

    axes(handles.orgIm);
    cla;
    imshow(handles.data.base_image);
    axis equal;
    axis tight;

end
close(helper);
```

Potrebno je bilo realizovati sljedeće funkcije:

- isjecanje – ***dos_crop***
- brisanje – ***dos_delete***
- slika u ogledalu
  - horizontalno – ***dos_mirrorHorizontaly***
  - vertikalno – ***dos_mirrorVerticaly***
- zumiranje – ***dos_zoom***
- negativ – ***dos_negative***
- binarna slika (automatski izbor praga i manuelni) – ***dos_threshold***
- rotacija za nx90°
  - rotacija udesno – ***dos_rotateRight***
  - rotacija ulijevo – ***dos_rotateLeft***

Sada ćemo detaljnije objasniti način realizacije svake od ovih funkcija.

2.1. Isjecanje – dos_crop

Za realizovanje ove funkcije koristili smo Matlab-ovu ugrađenu funkciju *getrect*, koja vraća matricu 1x4 u kojoj se nalaze koordinate početne tačke i širina selektovanog područja. Pošto se dobiju podataci tipa double, vršimo konverziju u integer, jer su slike tipa integer. Dio slike koji je na ovaj način selektovan pridružimo novoj slici, koju na kraju i prikažemo. Još jednostavniji način realizacije ove funkcije jeste korištenje Matlab-ove funkcije za crop: *imcrop*.

```
rect = int16(getrect);
image_crop = image(rect(2):(rect(2)+rect(4)), rect(1):(rect(1)+rect(3)) ,:);

% matlabov crop
% image_crop = imcrop;
```

2.2. Brisanje – dos_delete

Slično kao i kod isjecanja, i ovdje smo koristili funkciju *getrect* pomoću koje dobijemo selektovano područje, a onda na to područje upišemo “1”, tj. 255 (bijela boja). Kod je sljedeći:

```
rect = int16(getrect);
image_delete = image;
image_delete(rect(2):(rect(2)+rect(4)), rect(1):(rect(1)+rect(3)) ,:) = 255;
```

2.3. Slika u ogledalu vertikalno/horizontalno – dos_mirrorVerticaly/Horizontaly

Prvo smo napravili novu matricu dimenzija kao polazna slika. Prilikom realizovanja “ogledala” koristili smo Matlabove funkcije *fliplr* (vertikalno ogledalo) i *flipud* (horizontalno ogledalo). Primjenimo ovu funkciju na prvu komponentu (ukoliko se radi o slici u boji, odnosno na jedinu komponentu, ako je siva slika), a zatim primjenimo isti postupak i na preostale komponente (ako ih ima, tj. ako je slika u boji). Kod:

```
image_mirror = zeros( size(image,1), size(image,2),size(image, 3) );

% transfer color channels
image_mirror(:,:,1)=fliplr(image(:,:,1));       %%% flipud(image(:,:,1))
try
   image_mirror(:,:,2)=fliplr(image(:,:,2));    %%% flipud(image(:,:,2))
end
try
    image_mirror(:,:,3)=fliplr(image(:,:,3));   %%% flipud(image(:,:,3))
end

% ensure correct type
image_mirror = uint8(image_mirror);
```

2.4. Zum – dos_zoom

Poziv funkcije *zoom*. Kod je sljedeći:

```
function dos_crop(obj,event)
%  Detailed explanation goes here
zoom
```

2.5. Negativ – dos_negative

Negativ smo napravili korištenjem funkcije *imadjust*, koja ustvari funkcioniše djelovanjem na histogram. Jednostavno, “obrnemo” histogram, tj. zamjenimo vrijednosti svjetlina i, dobijemo negativ. Kod:

```
image_neg = imadjust(image, [0 1], [1 0], 1);
```

2.6. Binarna slika (automatski izbor praga i manuelni) – *dos_threshold*

Zadatak je bio da se od polazne slike napravi binarna slika pri čemu možemo koristiti prag koji dobijemo automatski, ili da mi proizvoljno izaberemo prag. Ovo je realizovano na taj način da se prilikom poziva funkcije za dobijanje binarne slike otvori novi prozor u kome će se prikazati vrijednost praga koja se dobije automatski, korištenjem funkcije *graythresh* , ali i klizač pomoću koga možemo vrijednost praga proizvoljno izabrati. Za dobijanje binarne slike koristili smo funkciju *im2bw* kojoj prosljeđujemo sliku i prag. Pored realizacije funkcije za dobijanje binarne slike, bilo je potrebno napisati i kod pomoću koga dobijamo novi prozor odgovarajuće veličine na kome se nalazi klizač, dugme *OK* kojim potvrđujemo izbor prikazane vrijednosti praga i Textbox u kome je prikazana trenutna vrijednost praga. Kod je sljedeći:

```
t = graythresh(image);
h = figure('Position', [300 300 120 120], 'Toolbar', 'none', 'MenuBar', 'none', 'Name',
'threshold', 'Resize', 'off');
uicontrol(h, 'Style', 'pushbutton', 'String', 'Ok',...
    'Position', [10 25 100 20], 'Callback', {@x,h});
ht = uicontrol(h, 'Style', 'text', 'Position', [10 75 100 15], 'Tag', 'valText',
'String', num2str(t));

uicontrol(h, 'Style', 'slider', 'Tag', 'sl', 'Position', [10 50 100 20], 'Min', 0,
'Max', 1, 'Value', t, 'Callback', {@intSlider_Callback,ht});

function intSlider_Callback(hObject, eventdata, ht)
    t = get(hObject,'value');
    set(ht,'String',num2str(t));
end

function x(xobj, event,h)
    close(h);
    image_prag = im2bw(image, t);
```

2.7. Rotacija za nx90° u lijevo/desno – dos_rotateLeft/Right

Slično kao i kod realizovanja slike u ogledalu, i ovdje smo prvo napravili novu matricu istih dimenzija kao i polazna slika. Takođe smo koristili funkcije *fliplr* i *flipud*, ali ovaj put smo ih primjenjivali na sliku koja je prethodno transponovana. Kada transponujemo neku sliku, dobijamo sliku koja je ustvari zarotirana i “ogledalo” polazne slike, a potom primjenom funkcija *fliplr* i *flipud* vršimo “korekciju” ogledala. Kod:

```
% rotateLeft
image_rot = zeros( size(image,2),
size(image,1), size(3) );
% transfer color channels
image_rot(:,:,1)=flipud(image(:,:,1)');
try

image_rot(:,:,2)=flipud(image(:,:,2)');
end
try

image_rot(:,:,3)=flipud(image(:,:,3)');
end
% ensure correct type
image_rot = uint8(image_rot);
```

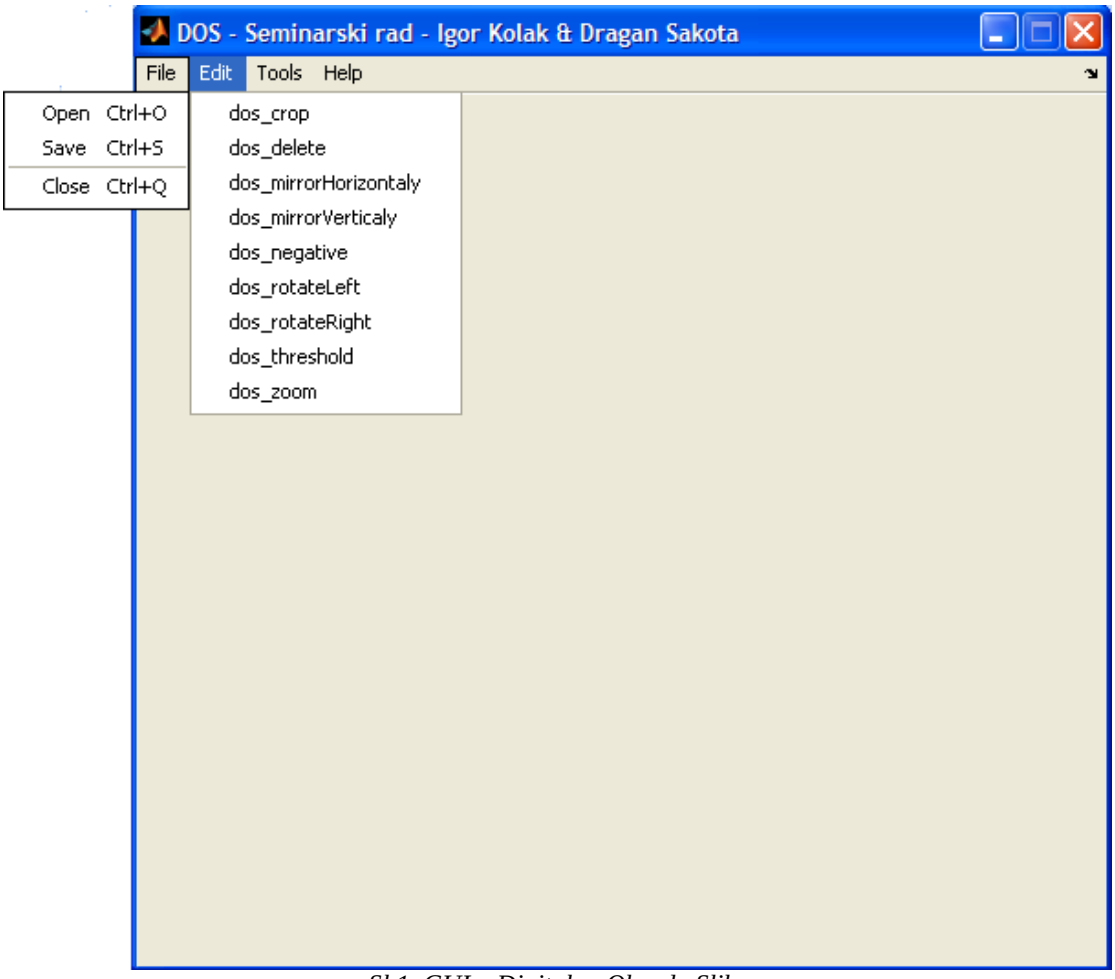
```
%rotateRight
image_rot = zeros( size(image,2),
size(image,1), size(image,3) );
% transfer color channels
image_rot(:,:,1)=fliplr(image(:,:,1)');
try
   image_rot(:,:,2)=fliplr(image(:,:,2)');
end
try

image_rot(:,:,3)=fliplr(image(:,:,3)');
end

% ensure correct type
image_rot = uint8(image_rot);
```

3. DEMONSTRACIJA FUNKCIONALNOSTI

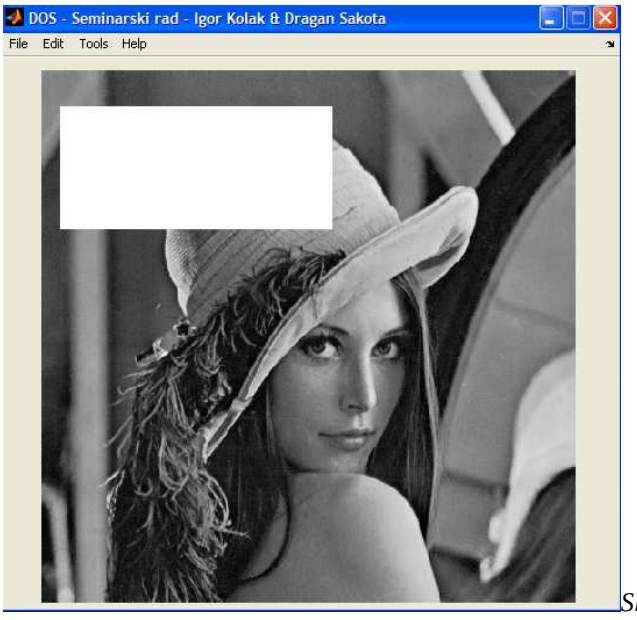
Prvo ćemo prikazati kako izgleda realizovani grafički interfejs:



Sl.1. GUI – Digitalna Obrada Slike

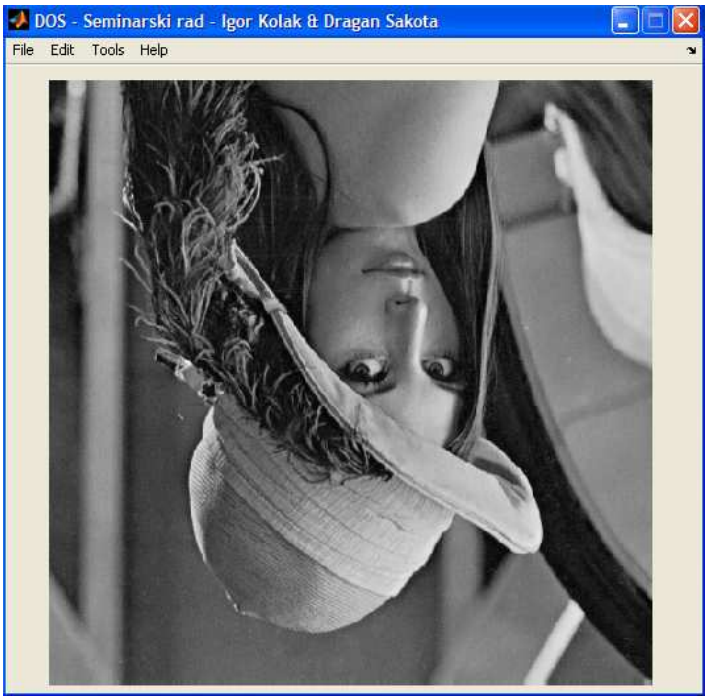
Na slici su prikazane funkcije koje se nalaze u glavnom meniju, pod opcijom File i Edit, a to su ustvari funkcije koje su realizovane u ovom radu. Njihovu funkcionalnost prikazaćemo na sljedećim slikama:

Sl.2.. dos_crop

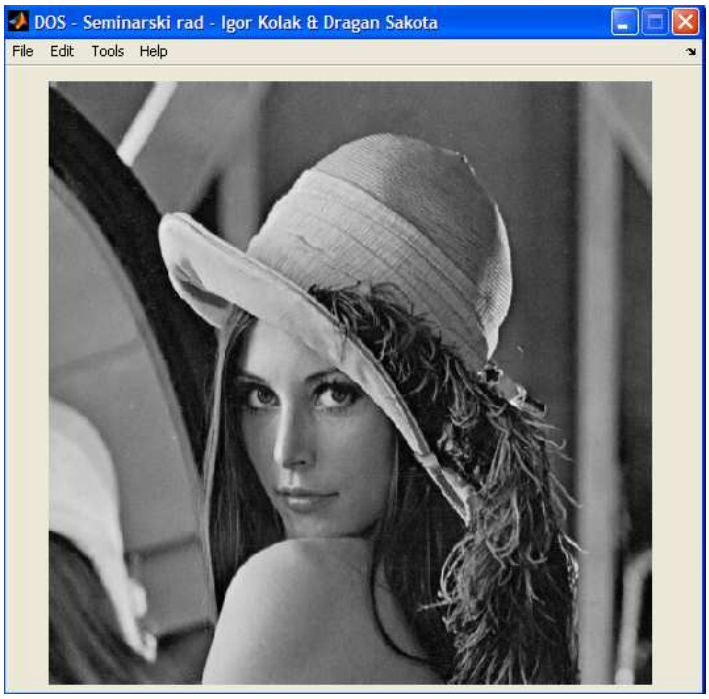


Sl.3. Sl.3.

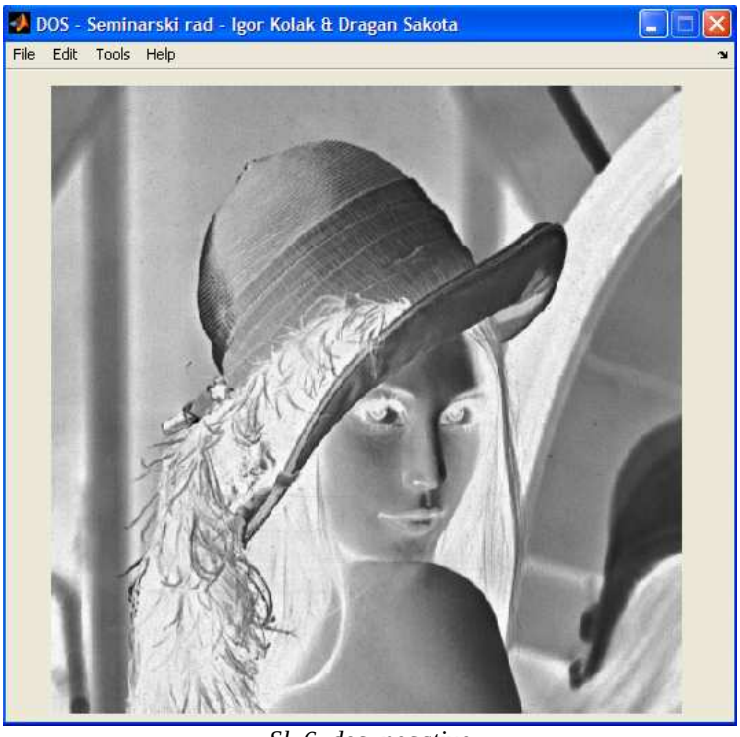
Sl.3. dos_delete



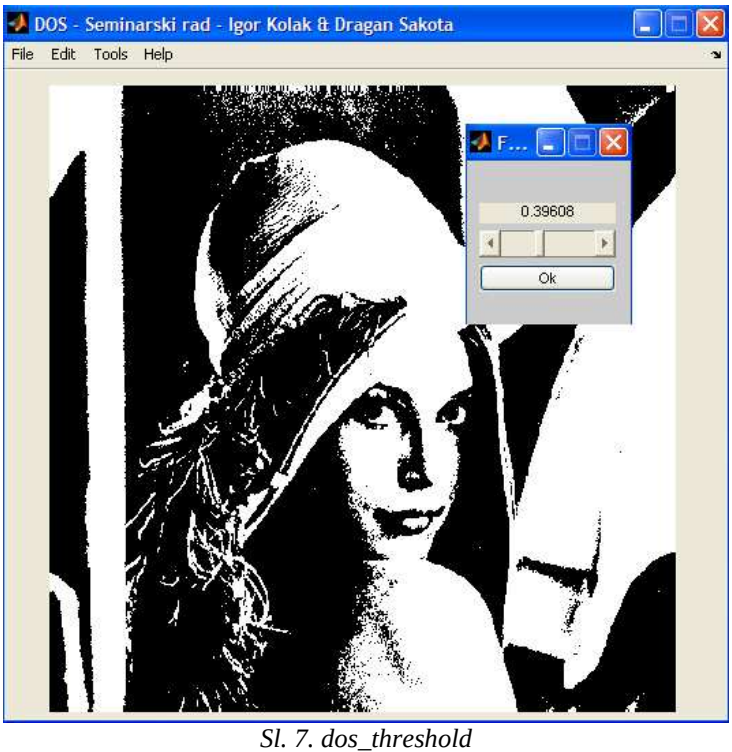
Sl.4. dos_mirrorHorizontaly



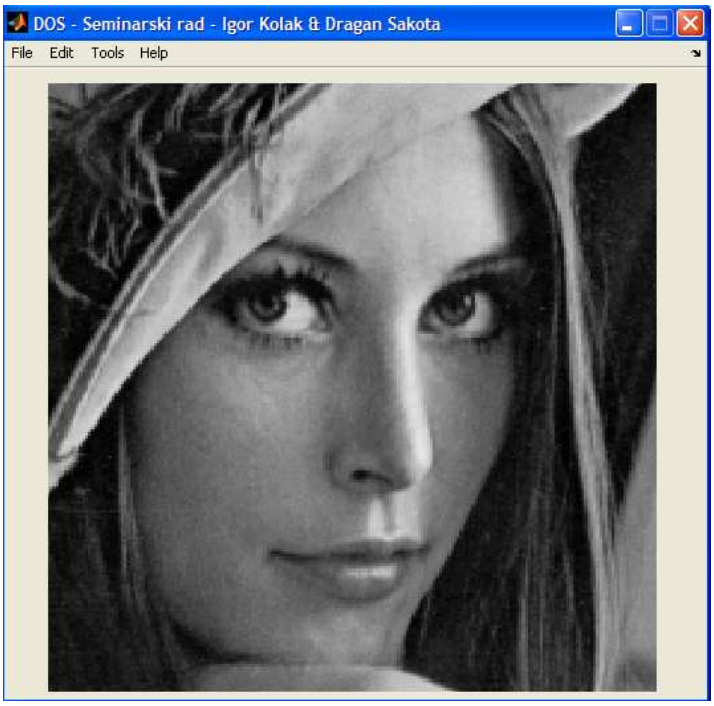
Sl. 5. dos_mirrorVerticaly



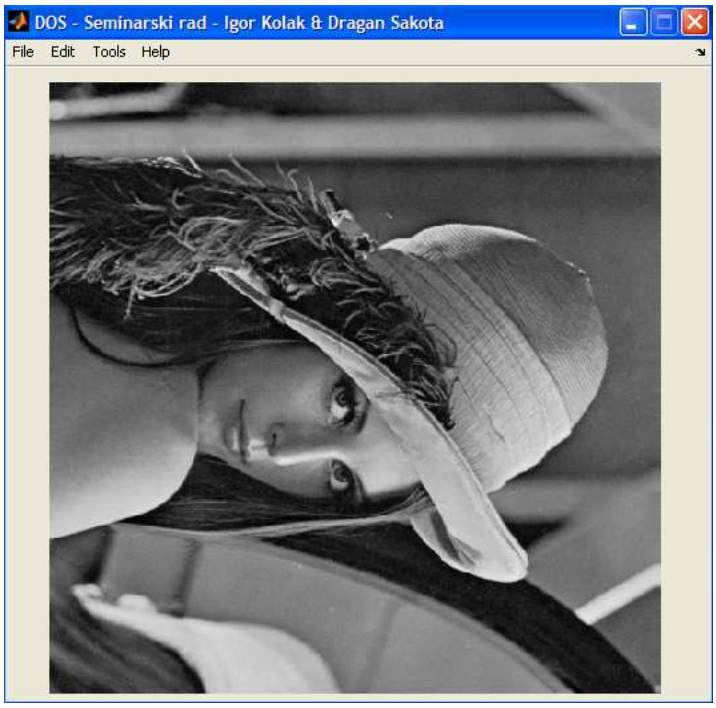
Sl. 6. dos_negative



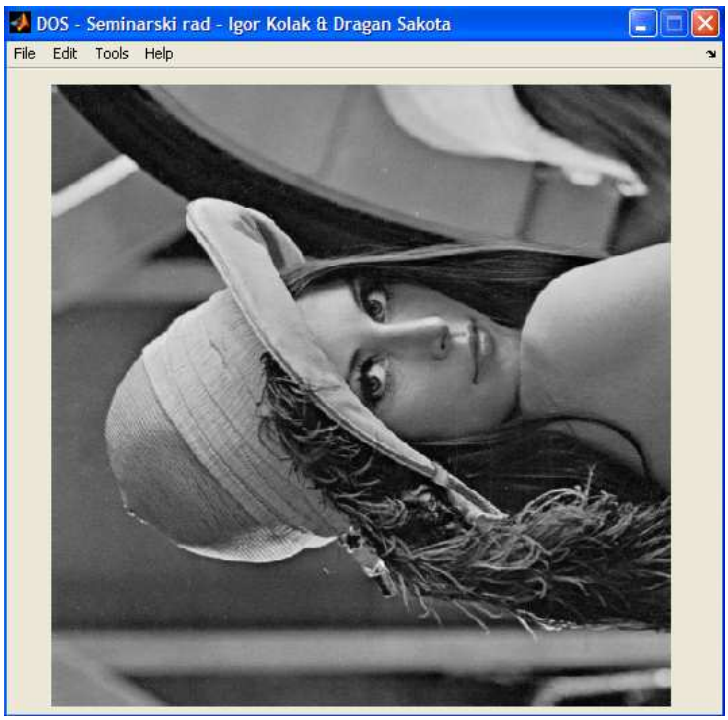
Sl. 7. dos_threshold



Sl. 8. dos_zoom



Sl. 9. dos_rotateRight



Sl. 10. dos_rotateLeft

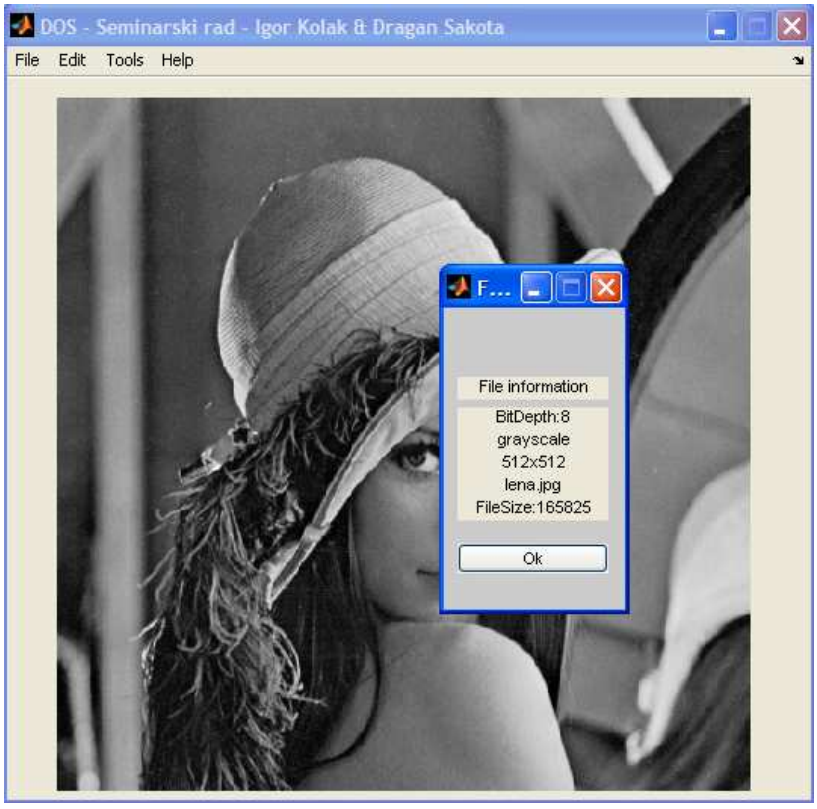
Na prethodnim slikama smo vidjeli kako djeluju pojedine funkcije na učitanu sliku. Funkcionalnost će biti dodatno objašnjena na prezentaciji zbog specifčnosti samog zadatka. Još jednom, da napomenemo da se na ovaj interfejs mogu lako dodati nove funkcije (zbog dinamičkog kreiranja menija) i na taj način se može napraviti interfejs za detaljnu obradu slike.

Dodatno smo napravili još jednu funkcionalnost, *dos_imfinfo*, koja nam daje osnovne informacije o učitanoj slici. Kod je sljedeći:

```
info = imfinfo(filename);

h = figure('Position', [300 200 120 200], 'Toolbar', 'none', 'MenuBar', 'none', 'Name',
'threshold', 'Resize', 'off');
uicontrol(h, 'Style', 'pushbutton', 'String', 'Ok',...
    'Position', [10 25 100 20], 'Callback', 'close');

uicontrol(h, 'Style', 'text', 'Position', [10 60 100 15], 'String', strcat('FileSize:
', num2str(info.FileSize)));
uicontrol(h, 'Style', 'text', 'Position', [10 75 100 15], 'String', info.Filename);
uicontrol(h,   'Style',   'text',   'Position',   [10   90   100   15],   'String',
strcat(num2str(info.Width), 'x ', num2str(info.Height)));
uicontrol(h, 'Style', 'text', 'Position', [10 105 100 15], 'String', info.ColorType);
uicontrol(h,  'Style',  'text',  'Position',  [10  140  100  15],  'String',  'File
information');
uicontrol(h, 'Style', 'text', 'Position', [10 120 100 15], 'String', strcat('BitDepth:
', num2str(info.BitDepth)));
```



Sl.11. dos_imfinfo

LITERATURA:

[1] Help Matlab
[2] Materijali sa vježbi iz predmeta *Digitalna obrada slike*
[3] matlabdb.mathematik.uni-stuttgart.de