Co-funded by the
Erasmus+ Programme
of the European Union

MIND COURSE SUPPORT

LECTURE 2
VISION TEHNOLOGY (VT)

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Contents

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. INTRODUCTION

This chapter presents extracts from the book written by Stan Sergiu-Dan "INDRUMĂTOR DE LUCRĂRI - Programare avansata in MATLAB", published at EDITURA RISOPRINT in 2018, which is used for teaching students image processing with MATLAB (Chapter 7) [STA18] and from the book written by Stan Sergiu-Dan "Programare avansată în MATLAB -Suport de curs" EDITURA RISOPRINT" in 2019 (Chapter 4) [STA19].

IMAGE PROCESSING IN MATLAB

MATLAB / Simulink provides solutions for image processing, analysis and development of image algorithms [www1] [www18] [STA18]. The chapter presents aspects regarding the processing and analysis of images in MATLAB using its functions for the purpose of acquiring, storing, viewing, changing and exploiting the visual information in images (fig. 1).

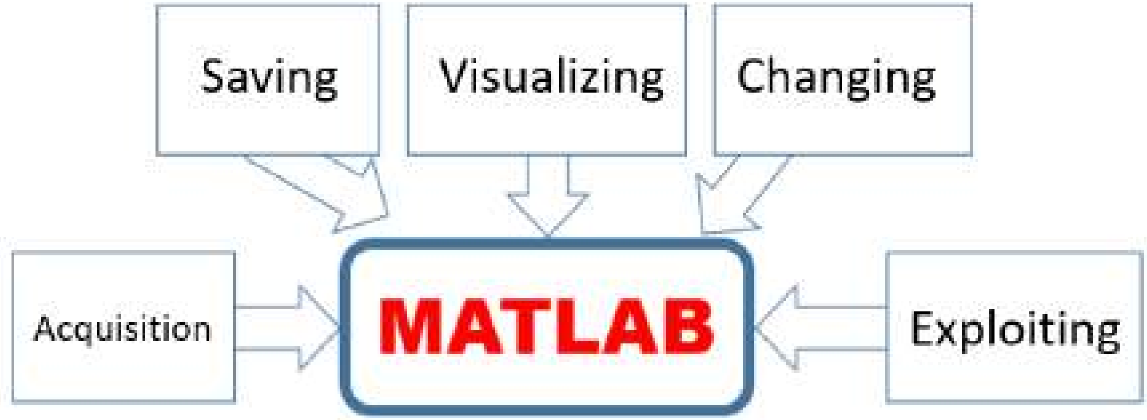


Fig. 1 Image processing and analysis in MATLAB

MATLAB supports the following types of image formats: BMP (Microsoft Windows Bitmap), GIF (Graphics Interchange Files), JPEG (Joint Photographic Experts Group), PCX (Paintbrush), TIFF (Tagged Image File Format), etc.

MATLAB functions used in image processing

The following table shows the MATLAB functions used in image processing.

Table 1 MATLAB functions used in image processing

Function	The meaning
imread	Reading an image from various formats
imwrite	Writing a picture
imfinfo	Image information
rgb2gray	Converting an ***RGB*** or ***colomap*** in gray scale
imshow	Display the image
imbinarize	Binarizing an image

Function *imread*

The MATLAB syntax for the imread function is as follows [www20]:

```
I = imread(filename)
I = imread(___,Name,Value)
[I,map] = imread(___)
```

The imread function will read in the format of a matrix various image formats. Further details and specifications on the imread function can be found in the official MATLAB documentation available online at [www20].

Example 1.

```
%program 1
clear all;
close all;
clc
```

```
I = imread('imagine.jpg');
```

In order to view the selected image, the following MATLAB image command can be used and the MATLAB imfinfo command is used to display more information.

```
>> image(I)
>> info = imfinfo('imagine.jpg');
```

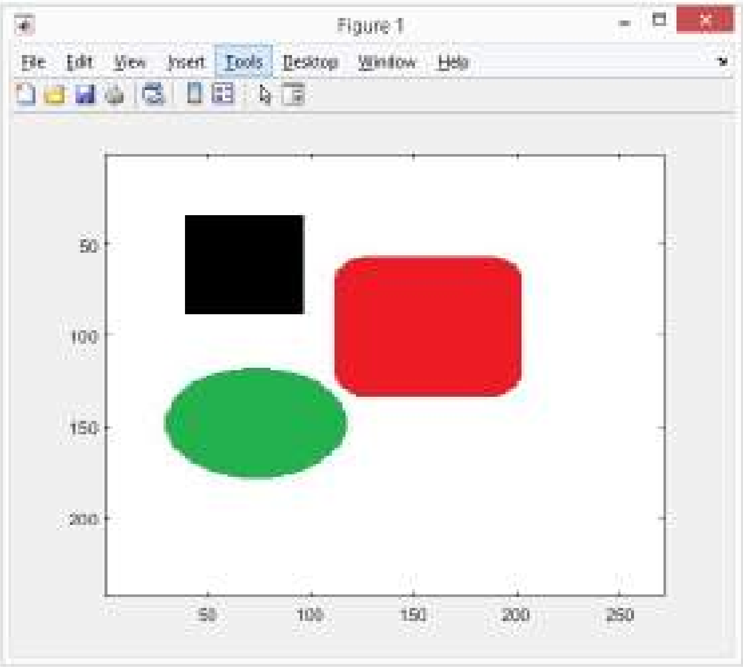


*Fig. 2 Example of using the function **imread***

```
>> info = imfinfo('imagine.jpg')
info =
 struct with fields:
   Filename: 'E:\New Folder\imagine.jpg'
  FileModDate: '09-Dec-2018 10:20:26'
   FileSize: 5651
    Format: 'jpg'
  FormatVersion: ''
    Width: 271
    Height: 243
   BitDepth: 24
   ColorType: 'truecolor'
 FormatSignature: ''
 NumberOfSamples: 3
  CodingMethod: 'Huffman'
  CodingProcess: 'Sequential'
   Comment: {}
  Orientation: 1
```

Function *imshow*

The MATLAB syntax for the ***imshow*** function is as follows [www21]:

```
imshow(I)
imshow(I,[low high])
```

The imshow function will display the images. Further details and specifications on the imshow function can be found in the official MATLAB documentation available online at [www21].

Example 2.

```
%program 2
clear all;
close all;
clc
I = imread('imagine.jpg');
imshow(I)
```

In order to view the image in a new figure, the MATLAB figure command is used. E.g:

```
I = imread('imagine.jpg');
figure
imshow(I)
```

Function *rgb2gray*

The MATLAB syntax for the ***rgb2gray*** function is as follows [www23]:

```
I = rgb2gray(RGB)
New_img = rgb2gray(New_img)
```

The rgb2gray function will convert a color image to gray levels. Further details and specifications on the rgb2gray function can be found in the official MATLAB documentation available online at [www23].

Example 3.

```
%program 3
clear all;
close all;
clc

I = imread('imagine.jpg');
imshow(I)
I_gri=rgb2gray(I);
imshow(I_gri)
```

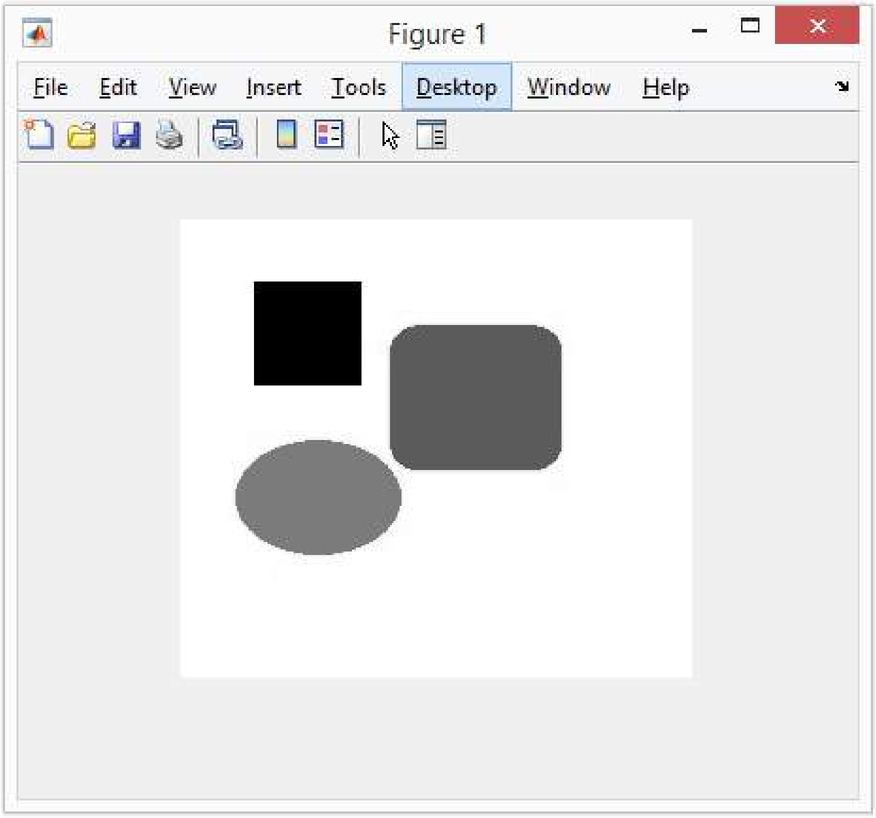


Fig. 3 Example of using the rgb2gray function

Function *imbinarize*

The MATLAB syntax for the join function is as follows [www24]:

```
I = imread('imagine.jpg');
BW= imbinarize(I);
```

The join function converts any image to an image with only two values, 0 and 1. Further details and specifications on the join function can be found in the official MATLAB documentation available online at [www24].

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Example 4.

```
%program 4

clear all;

close all;

clc

I = imread('imagine.jpg');

imagine_gri=rgb2gray(I);

BW = imbinarize(imagine_gri);

figure

imshowpair(I,BW,'montage')
```

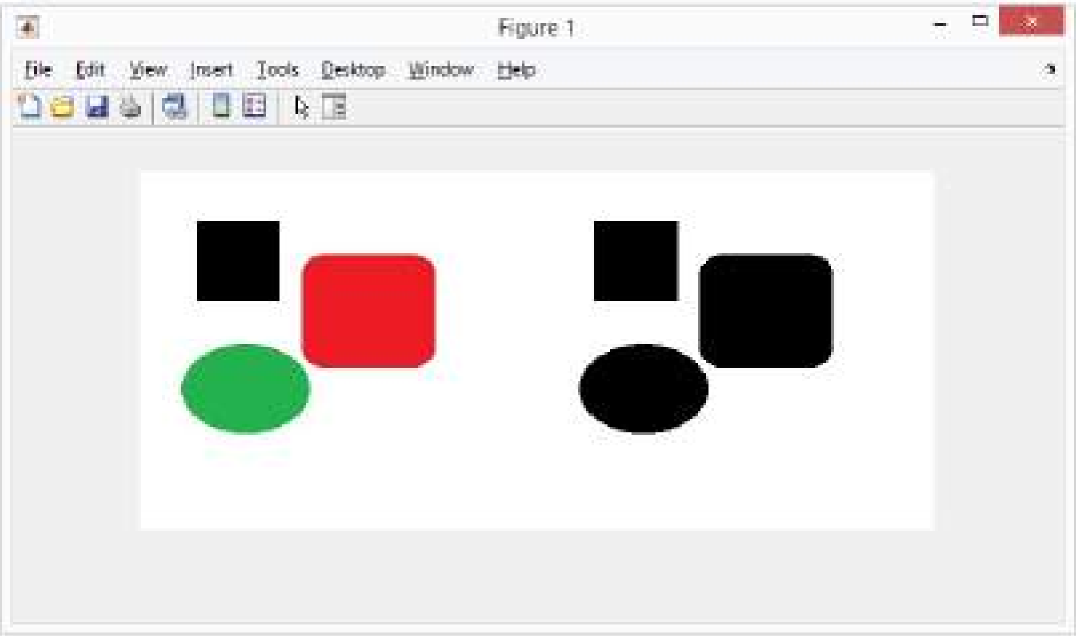


Fig. 4 Example of using of the function ***imbinarize***

Function *imwrite*

The MATLAB syntax for the imwrite function is as follows [www22]:

```
imwrite(I,filename)
imwrite(I,map,filename)
```

The imwrite function saves an image to disk. Further details and specifications on the imwrite function can be found in the official MATLAB documentation available online at [www22].

Example 5.

```
%program 5
clear all;
close all;
clc
I = imread('imagine.jpg');
imagine_gri=rgb2gray(I);
BW = imbinarize(imagine_gri);
figure
imshowpair(I,BW,'montage')
imwrite (255*uint8(BW), 'imagine.bmp', 'bmp');
```

The image is saved on the disc (fig. 5).

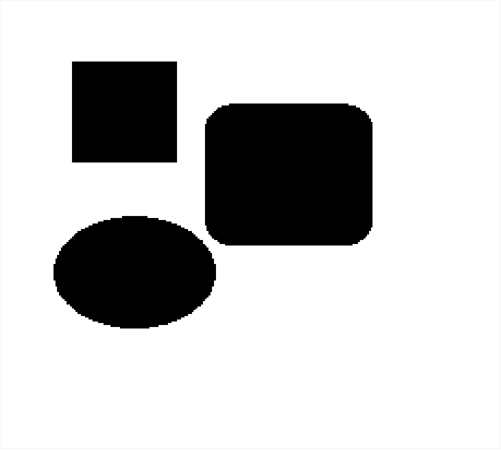


Fig. 5 Example of using the imwrite function

OpenCV and MATLAB

MATLAB works with the OpenCV image processing package for both algorithm development, image or video file analysis. The main purpose of the MATLAB-OpenCV connection is efficiency. MATLAB provides integration with OpenCV through the OpenCV C ++ API package [www25]. OpenCV is a free function library "Open Source Computer Vision Library".

Further details and specifications on how MATLAB works with OpenCV can be found in the official MATLAB documentation available online at [www25].

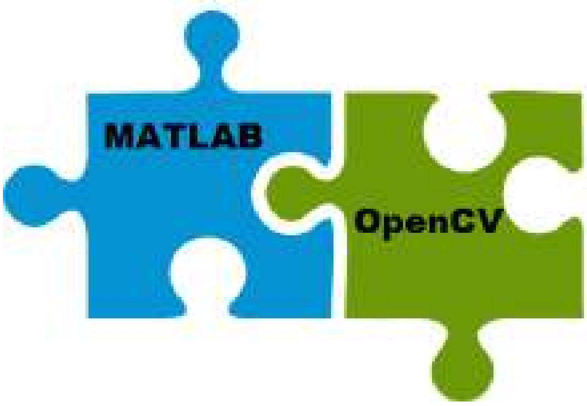


Fig. 6 MATLAB and OpenCV

Image processing in Simulink

In order to process images, Simulink offers the possibility to use the function package called Computer Vision System Toolbox [www26] [www27]. In fig. 7 and fig. 8 shows the Simulink blocks corresponding to the Computer Vision System Toolbox function package. More details on image processing in SIMULINK can be found in the official MATLAB documentation available online at [www26] [www27].

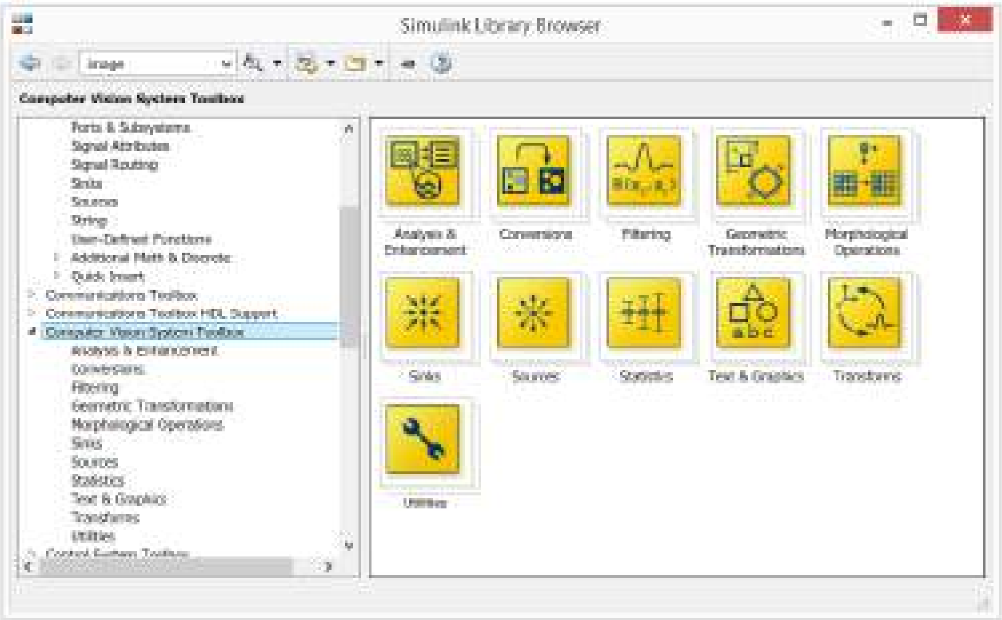


*Fig. 7 **Toolbox** Computer Vision System Toolbox*

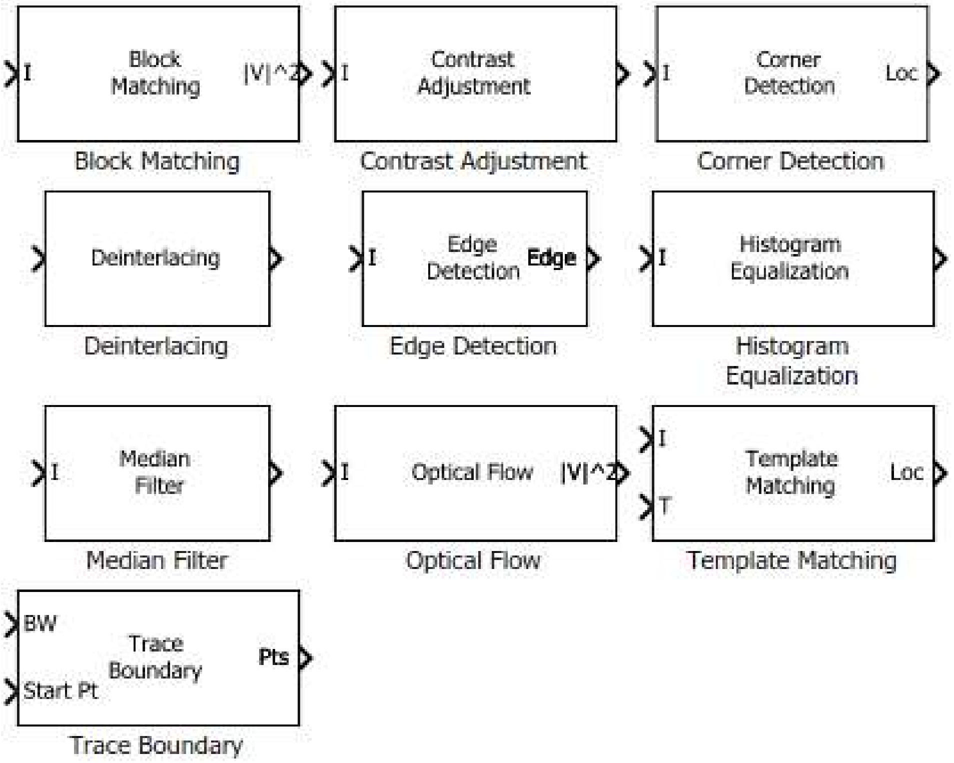


Fig. 8 Simulink blocks from toolbox Computer Vision System Toolbox

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

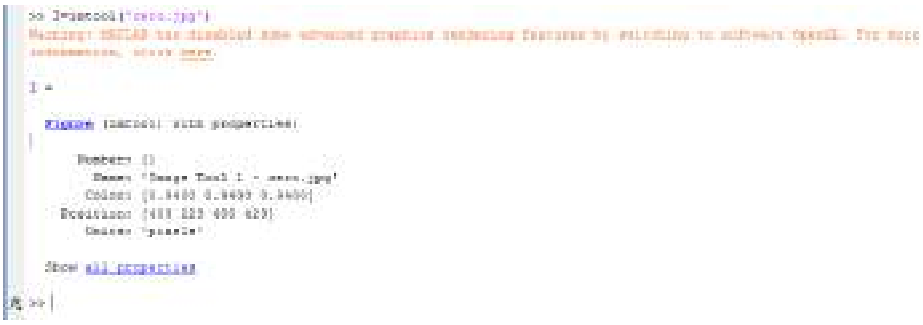
Image processing techniques in MATLAB

The basic functions in image processing in MATLAB are **imread**, **imshow**, **imtool** [www5] [www6]. It is recommended that at the beginning of each application made in MATLAB to call the commands for memory release, closing all open windows, cleaning the command window, respectively ***clear all; close all; CLC***.

For example, the ***imread*** [www5] [www7] command is used to read an image.

>> imread('imagine.jpg')

Once the image has been read, it can be viewed. In order to display the selected image, you can use the ***imshow*** (most commonly used function for displaying images) or ***imtool*** (used for simple image processing operations) [www5].



```
>> I=imtool('cerc.jpg')
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more
information, click here.

I =

  Figure (imtool) with properties:

      Number: []
        Name: 'Image Tool 1 - cerc.jpg'
       Color: [0.9400 0.9400 0.9400]
    Position: [411 223 400 623]
       Units: 'pixels'

  Show all properties

>> 
```

*Fig. 9 The **imtool** function used to open a test image*

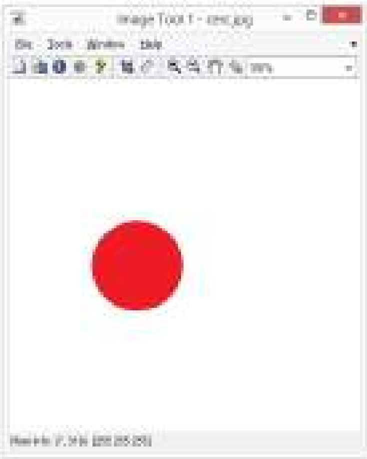


*Fig. 10 The **imtool** function used to open a test image*

Saving an image to a file is done using the ***imwrite*** command. For other functions, the MATLAB documentation available at [www5] is recommended.

EXAMPLES

B1. EXAMPLE OF RECOGNIZING THE DIMENSIONS OF A RECTANGLE IN AN IMAGE

The test image is the following image (Fig. 11). The file name is **dreptunghi.jpg**.

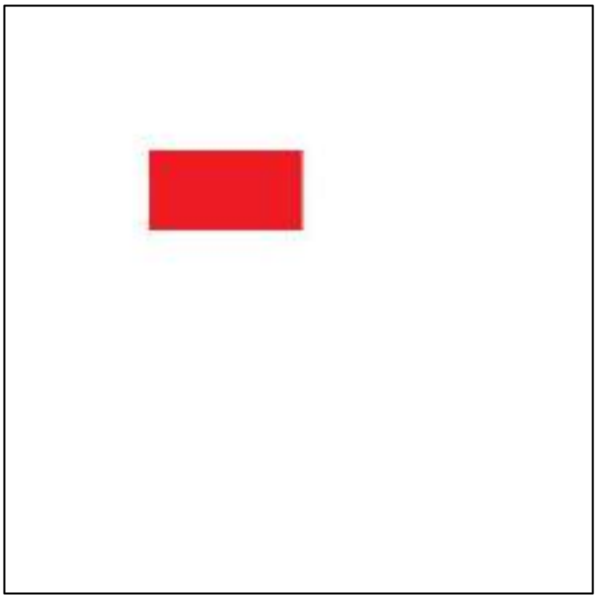


Fig. 11 Test image

```
%program 1
%program for recognizing the dimensions of a
%rectangle from an image dreptunghi.jpg

clear all;
close all;
clc

imagine_test = imread('dreptunghi.jpg');
```

```
figure;

imshow(imagine_test);

imagine_gri=rgb2gray(imagine_test);
figure;
imshow(imagine_gri)

prag_gri=graythresh(imagine_gri);

bw=im2bw(imagine_gri,prag_gri);

bw=~bw;
figure;
imshow(bw);

colturi = corner(bw);

figure; imshow(imagine_test);hold on
plot(colturi(:,1), colturi(:,2), 'blue*');
laturaX=colturi(3,1)-colturi(1,1);
laturaY=colturi(2,2)-colturi(1,2);

[L Ne]=bwlabel(bw);
hold on
prop=regionprops(L);
hold on

for n=1:length(prop)
```

```
rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
x=prop(n).Centroid(1);
y=prop(n).Centroid(2);

coordX(n)=prop(n).BoundingBox(1);
coordY(n)=prop(n).BoundingBox(2);

txtX = ['x= ' num2str(laturaX) ' px'];
txtY = ['y= ' num2str(laturaY) ' px'];
text(coordX(n),coordY(n)-10,txtX);
text(coordX(n)-10,coordY(n)+10,txtY) ;

plot(x,y,'*')

end
```

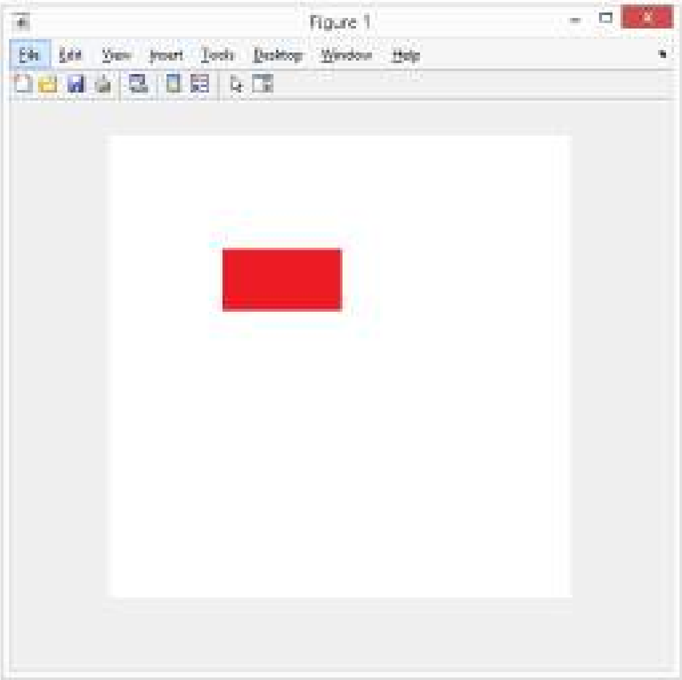


Fig. 12 Test image

In order to obtain a binary version of a given image, the method referred to as threshold detection is used. In MATLAB the function that performs the threshold detection is called ***graythresh***. Threshold detection automatically calculates a reasonable threshold value that will be used to convert a multi-step image to a binary image. In order to perform the actual conversion, the ***im2bw*** function is applied.

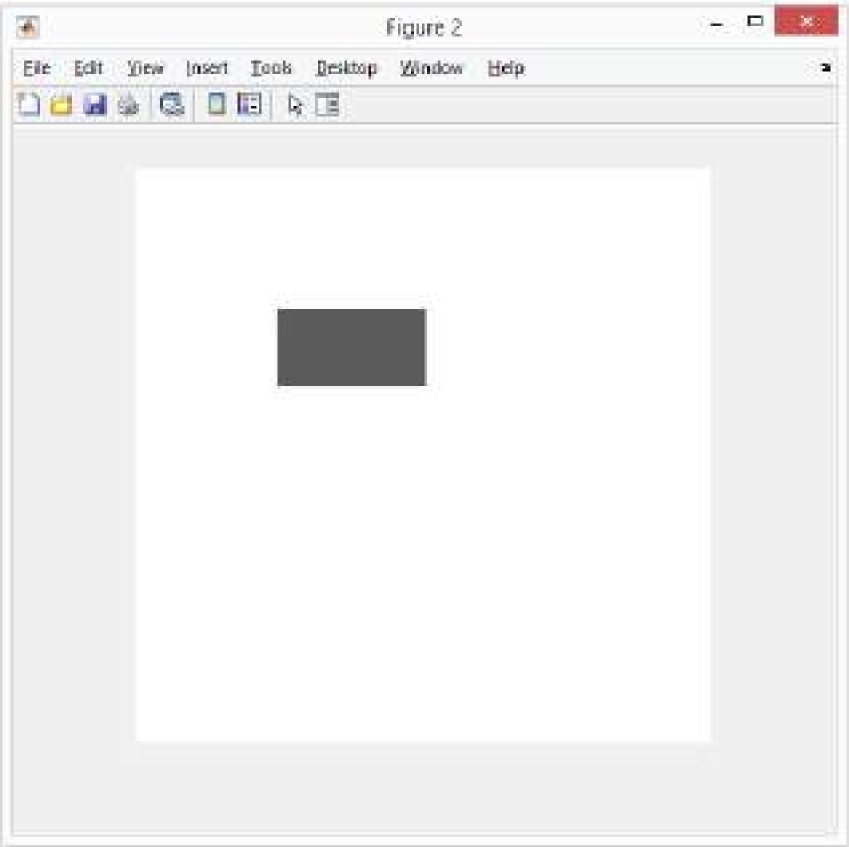


Fig. 13 Test image after applying the threshold

A binary image resulting from the ***im2bw*** command (called ***bw*** in this example) is obtained. It is of the logical class, as can be seen in the ***whos*** call.

```
bw=im2bw(imagine_gri,prag_gri);
A=~bw;
figure;
imshow(A);
```

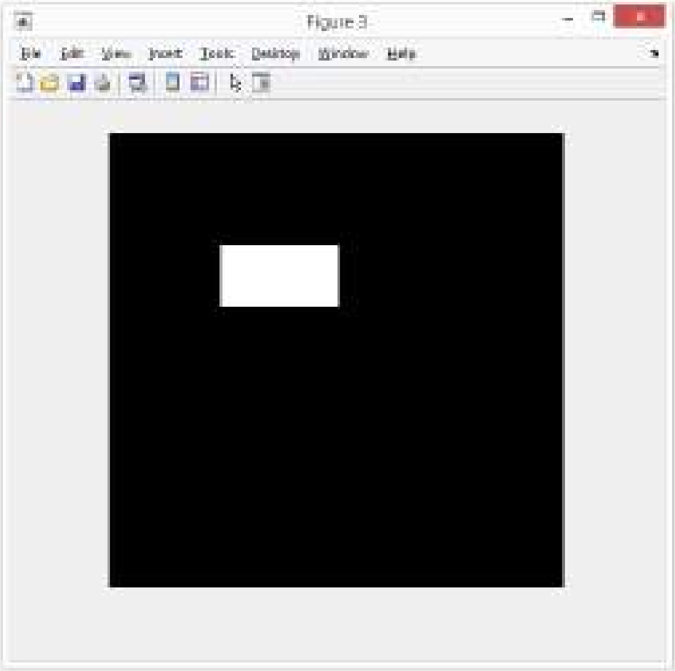


Fig. 14 Test image in binary format

The dimensions of the rectangle are found, the result is shown in fig. 15.

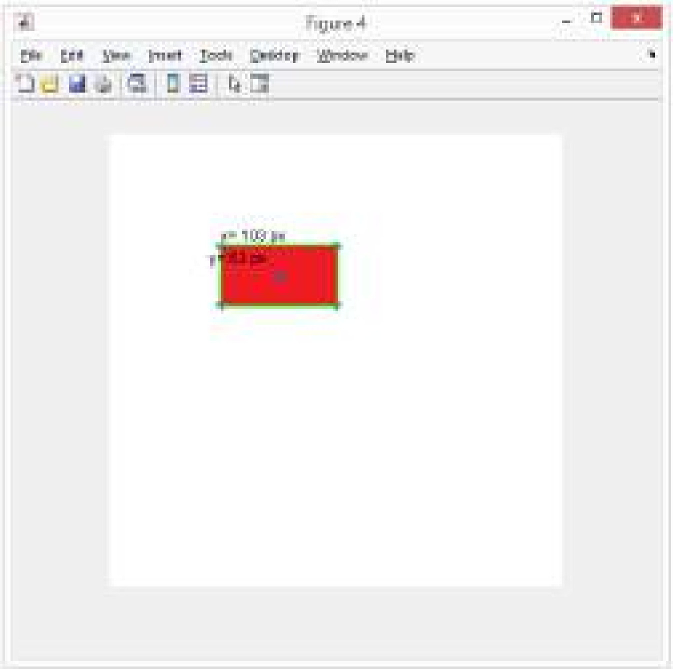


Fig. 15 Test image

B2. Example of recognizing the radius and center of a circle in an image

The test image is the following image (fig. 8). The file name is **cerc.jpg.**

```
clear all;
close all;
clc



imagine_test = imread('cerc.jpg');


figure;


imshow(imagine_test);


imagine_gri=rgb2gray(imagine_test);
figure;
imshow(imagine_gri)


prag_gri=graythresh(imagine_gri);


bw=im2bw(imagine_gri,prag_gri);


A=~bw;


[centers, radii, metric] = imfindcircles(A,[50 100]);


centersStrong5 = centers(1:1,:);
radiiStrong5 = radii(1:1);
metricStrong5 = metric(1:1);
```

```
viscircles(centersStrong5, radiiStrong5,'EdgeColor','r');

centers;
radii

hold on

x=centers(1);
y=centers(2);

txtXX = ['Coordonatele cercului sunt:'];
txtYY = ['Raza cercului este:'];
txtX = ['x= ' num2str(x) ' px'];
txtY = ['y= ' num2str(y) ' px'];
raza = ['r = ' num2str(radii) ' px'];

text(10,20,txtXX);
text(10,80,txtYY);
text(10,40,txtX);
text(10,60,txtY);
text(10,100,raza);

hold off

imagine_test = imread('cerc.jpg');
figure;
imshow(imagine_test);
```

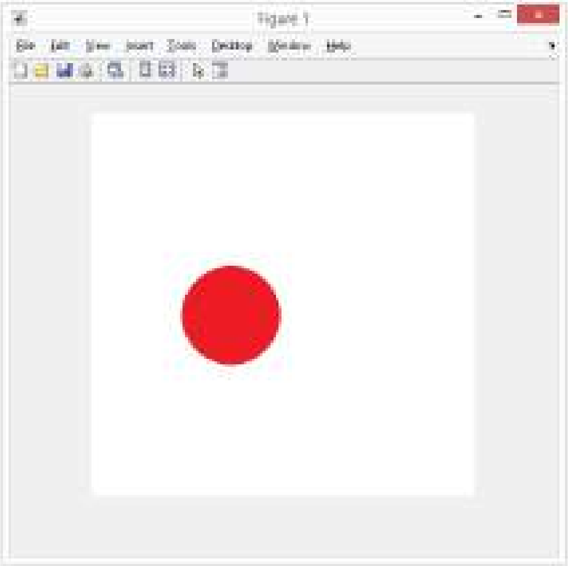


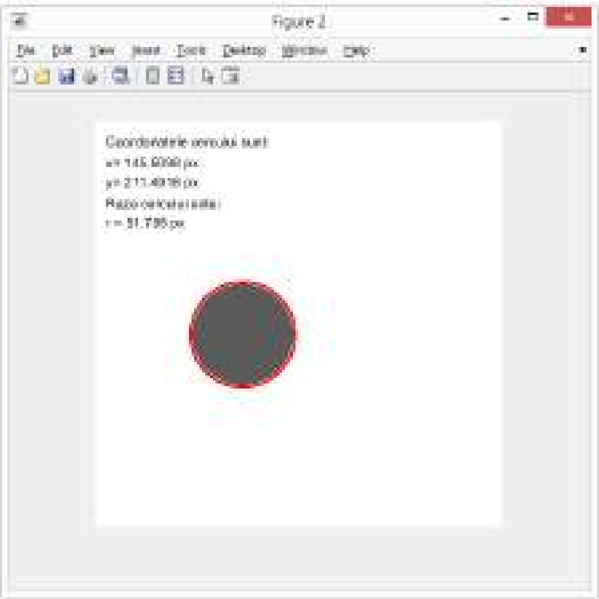
Fig. 16 Test image

```
imagine_gri=rgb2gray(imagine_test);

figure;

imshow(imagine_gri)
```

In fig. 17 the result obtained is presented.



Coordonatele cercului sunt:
x= 145.8098 px
y= 211.4918 px
Raza cercului este:
r = 51.795 px

Fig. 17 The result obtained for the test image

The coordinates of the circle and the radius are displayed on the image.

REFERENCES

[STA18] *Stan Sergiu-Dan, ÎNDRUMĂTOR DE LUCRĂRI - Programare avansată în MATLAB, EDITURA RISOPRINT, ISBN 978-973-53-2303-5, Cluj-Napoca 2018.*

[STA19] *Stan Sergiu-Dan, Programare avansată în MATLAB -Suport de curs" EDITURA RISOPRINT" 978-973-53-2330-1, Cluj-Napoca 2019.*

[STA03] Stan Sergiu-Dan, Diplomarbeit, *Analyse und Optimierung der strukturellen Abmessungen von Werkzeugmaschinen mit Parallelstruktur*, IWF-TU Braunschweig, 2003, Germania.

[STA04] Stan Sergiu-Dan, Referat nr. 1, *Stadiul actual al construcţiei şi implementării roboţilor paraleli*, Universitatea Tehnică din Cluj-Napoca, 2004.

[STA05] Stan Sergiu-Dan, Referat nr. 2, *Concepte şi metode de optimizare*, Universitatea Tehnică din Cluj-Napoca, 2005.

[STA06a] S. Stan, *Workspace optimization of a two degree of freedom mini parallel robot*, 2006 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics – AQTR 2006 (THETA 15), May 25-28 2006, Cluj-Napoca, Romania, IEEE Catalog number: 06EX1370, ISBN: 1-4244-0360-X, pp. 278-283.

[STA06b] Stan Sergiu-Dan, *A comparison of genetic algorithm and simulated annealing optimization methods applied to workspace optimization of a RPRPR mini parallel robot*, Robotica 2006, 7th-9th September 2006, The 3rd International Conference on Robotics, ISSN 1011-2855, Buletinul Institutului Politehnic din Iasi, Editat de Universitatea Tehnica "GH. Asachi", Iasi, Tomul LII (LVI), Fascicula 7B, pp. 187-192.

[STA07] Sergiu-Dan Stan, Optimizare cu algoritmi genetici - aplicatii in mecatronica, Ed. Todesco, Cluj-Napoca, 2007, ISBN 978-973-7695-38-3.

[www1] https://www.mathworks.com/ ultima accesare dec. 2018.

[www2] http://www.tmt.ugal.ro/crios/Support/ANPT/Tutoriale/MATLAB_IN_INGINERIE.pdf Cristian Iosifescu, Suport Curs: *Analiza Numerica a Proceselor Termo-mecanice*, f.d., ultima accesare dec. 2018.

[www3] https://www.mathworks.com/solutions/mechatronics.html. ultima accesare dec. 2018.

[www4] http://web.ulbsibiu.ro/catalina.neghina/Resurse/carti/carte_Matlab.pdf MATLAB: un prim pas spre cercetare / Cătălina Neghină, Alina Sultana, Mihai Neghină. - Sibiu : Editura Universităţii "Lucian Blaga" din Sibiu, 2016, ISBN 978-606-12-1213-2, ultima accesare dec. 2018.

[www5] Documentație Matlab ‘Help MATLAB Documentation’, https://www.mathworks.com/help/matlab/ ultima accesare dec. 2018.

[ZAH14] ZAHARIEA, DĂNUȚ, MATLAB: calcul numeric și simbolic / Zahariea Dănuț - Iași: PIM 2014 Bibliogr. ISBN 978-606-13-2091-2.

[www6] http://users.utcluj.ro/~tmarita/SVR/Lab/lab_2_ro.pdf ultima accesare dec. 2018.

[www7] http://ac.upg-ploiesti.ro/cursuri/pi/laborator/PI%20-%20Lab%202.pdf ultima accesare dec. 2018.

[www8] https://www.arduino.cc/ ultima accesare dec. 2018.

[www9] https://www.mathworks.com/matlabcentral/fileexchange/47522-matlab-support-package-for-arduino-hardware ultima accesare dec. 2018.

[www10] https://www.mathworks.com/hardware-support/arduino-matlab.html ultima accesare dec. 2018.

[www11] http://fritzing.org/home/ ultima accesare dec. 2018.

[www12] https://www.mathworks.com/help/supportpkg/arduino/index.html ultima accesare dec. 2018.

[www13] https://www.raspberrypi.org/ ultima accesare dec. 2018.

[www14] https://www.mathworks.com/hardware-support/raspberry-pi-matlab.html ultima accesare dec. 2018.

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

[www15] https://www.mathworks.com/help/supportpkg/raspberrypi/index.html ultima accesare dec. 2018.

[www16] https://www.mathworks.com/discovery/matlab-gui.html ultima accesare dec. 2018.

[www17] https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html;jsessionid=ad2ba5b5232713b01d7b16a03836 ultima accesare dec. 2018.

[www18] https://www.mathworks.com/products/image.html ultima accesare dec. 2018.

[www19] https://www.mathworks.com/help/images/image-analysis.html ultima accesare dec. 2018.

[www20] https://www.mathworks.com/help/matlab/ref/imread.html ultima accesare dec. 2018.

[www21] https://www.mathworks.com/help/images/ref/imshow.html ultima accesare dec. 2018.

[www22] https://www.mathworks.com/help/matlab/ref/imwrite.html ultima accesare dec. 2018.

[www23] https://www.mathworks.com/help/matlab/ref/rgb2gray.html ultima accesare dec. 2018.

[www24] https://www.mathworks.com/help/images/ref/imbinarize.html ultima accesare dec. 2018.

[www25] https://www.mathworks.com/discovery/matlab-opencv.html ultima accesare dec. 2018.

[www26] https://www.mathworks.com/help/vision/ ultima accesare dec. 2018.

[www27] https://www.mathworks.com/help/vision/computer-vision-with-simulink.html ultima accesare dec. 2018.

[www28] https://www.mathworks.com/products/simulink.html ultima accesare dec. 2018.

[www29] https://www.mathworks.com/help/dsp/ug/generate-c-code-from-simulink-model.html ultima accesare dec. 2018.

[www30] https://www.mathworks.com/products/3d-animation.html ultima accesare dec. 2018.

[www31] Documentație Matlab 'Help MATLAB Documentation', https://www.mathworks.com/help/matlab/ ultima accesare dec. 2018.

[www32] https://www.mathworks.com/help/sl3d/import-vrml-models-from-cad-tools.html ultima accesare dec. 2018.

[www33] https://www.mathworks.com/hardware-support/arduino-matlab.html ultima accesare dec. 2018.

[www34] https://www.mathworks.com/help/supportpkg/arduinoio/ug/connect-to-arduino-hardware.html ultima accesare dec. 2018.

[www35] https://www.mathworks.com/hardware-support/raspberry-pi-matlab.html ultima accesare dec. 2018.

[www36] https://www.arduino.cc/ ultima accesare dec. 2018.

[www37] https://www.raspberrypi.org/ ultima accesare dec. 2018.

[www38] https://matlab.mathworks.com/ ultima accesare dec. 2018.

[www39] https://www.mathworks.com/hardware-support/raspberry-pi-simulink.html ultima accesare dec. 2018.

[NEG16] http://web.ulbsibiu.ro/catalina.neghina/Resurse/carti/carte_Matlab.pdf MATLAB: un prim pas spre cercetare / Cătălina Neghină, Alina Sultana, Mihai Neghină. - Sibiu : Editura Universităţii "Lucian Blaga" din Sibiu, 2016, ISBN 978-606-12-1213-2, ultima accesare dec. 2018.

[STA03] Stan Sergiu-Dan, Diplomarbeit, *Analyse und Optimierung der strukturellen Abmessungen von Werkzeugmaschinen mit Parallelstruktur*, IWF-TU Braunschweig, 2003, Germania.

[STA07] Sergiu-Dan Stan, Optimizare cu algoritmi genetici - aplicatii in mecatronica, Ed. Todesco, Cluj-Napoca, 2007, ISBN 978-973-7695-38-3.

This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

[STA18] Stan Sergiu-Dan, Programare avansată în MATLAB: Îndrumător de lucrări, Editura Risoprint, 2018, ISBN 978-963-53-2303-5.

[ZAH14] ZAHARIEA, DĂNUȚ, MATLAB: calcul numeric și simbolic / Zahariea Dănuț - Iași: PIM 2014 Bibliogr. ISBN 978-606-13-2091-2.