



15 გამონასახის დამუშავება

MATLAB გააჩნია სტანდარტული ალგორითმებისა და გრაფიკულ საშუალებათა ერთობლიობა გამონასახის დამუშავებისათვის. შესაძლებელი დაბალი გარჩევის, დაზიანებული გამონასახის აღდგენა და გაუმჯობესება. ამოვჭრათ დეტალები გამონასახიდან და გავაანალიზოთ გამოსახულება ფორმისა თუ სტრუქტურის თვალსაზრისით. **Tulbox- Image Processing** – გვთავაზობს ამისათვის უკვე დაწერილ **M-ფუნქციებს**, თუმცა მომხმარებელს შეუძლია თავად შექმნას საჭირო ალგორითმის მიხედვით ახალი **MATLAB** ფუნქციები.

MATLAB საშუალებას გვაძლევს დავამუშაოთ გამონასახები, რომელნიც მიღებულია სხვადასხვა ტიპის მიმღებებით როგორცაა ციფრული კამერა, ფრეიმ-გრაბერი, კოსმოსურ თანამგზავრებზე დადგმული სხვადასხვა ტიპის მიმღები, მიმღები, რომლითაც აღჭურვილია საელიცინო მიკროსკოპი, ტელესკოპი თუ სხვა მრავალი სამეცნიერო ინსტრუმენტი.

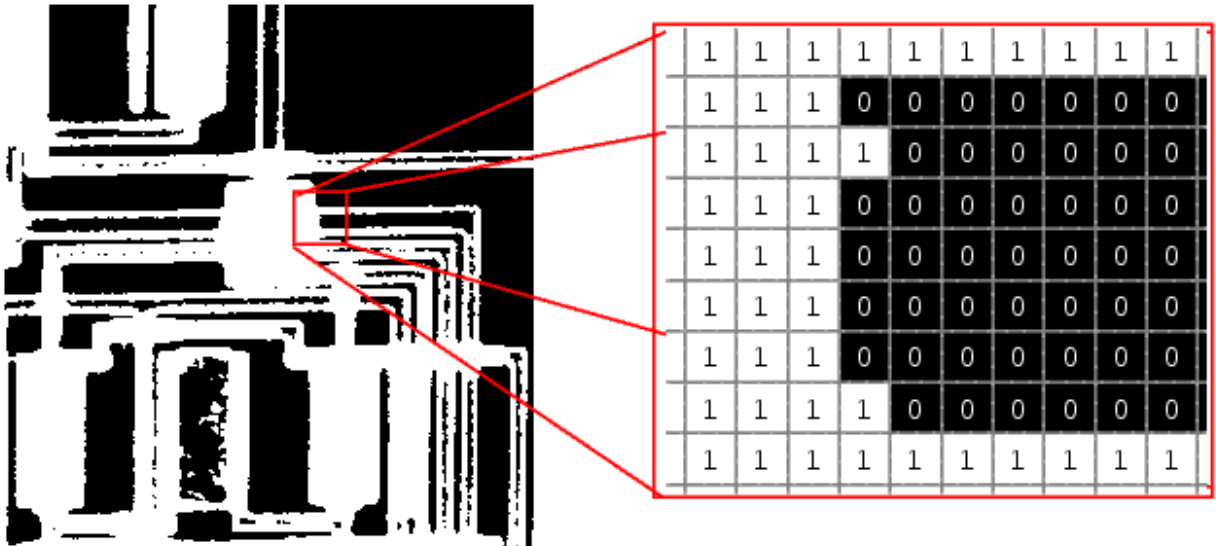
- 15.1 გამონასახის იმპორტ-ექსპორტი
- 15.2 არითმეტიკული ოპერაციები გამონასახზე
- 15.3 გამონასახის ზომის შეცვლა
- 15.4 გამონასახის შემობრუნება
- 15.5 დეტალის ამოჭრა გამონასახიდან
- 15.6 ინფორმაცია გამონასახის შესახებ და სტატისტიკური ანალიზი

15.7 ბრძანებები და ფუნქციები

15.1 შესავალი

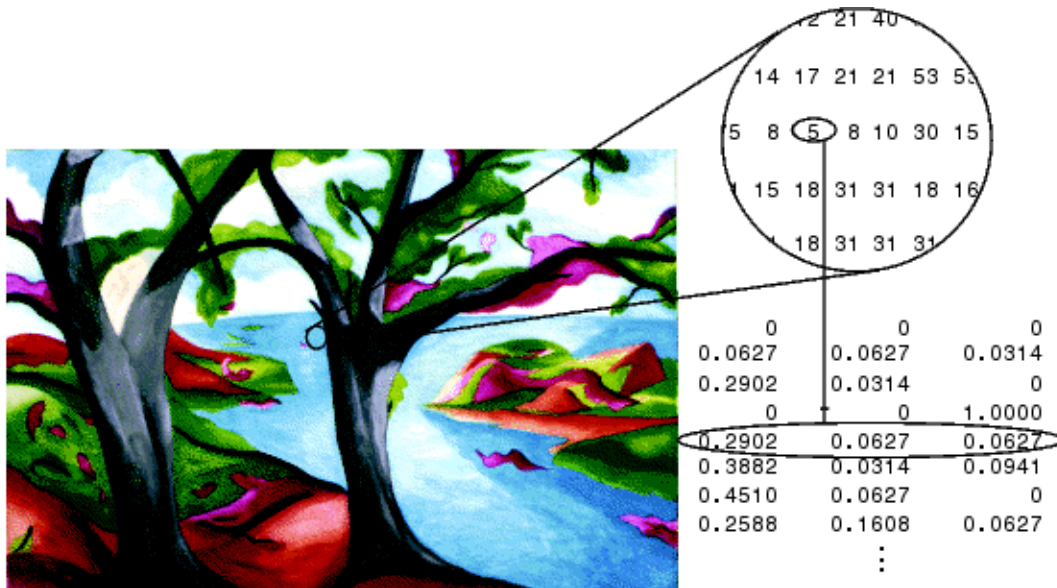
MATLAB გააჩნია სპეციალური ფუნქციათა კრებული გამონასახისათვის. იმისათვის რომ შევამოწმოთ არის თუ არა სისტემაში ინსტალირებული **Image Processing Toolbox** ვსარგებლობთ ფუნქციით **ver** ფუნქცია გვაწვდის ინფორმაციას MATLAB და Toolbox შესახებ, რომლებიც ინსტალირებულია მოცემულ სისტემაში. გამონასახის 4 ძირითადი ტიპი არსებობს:

- ბინარული, ორობითი გამონასახი – ლოგიკური მასივი რომელიც შეიცავს მხოლოდ 0 და 1 მნიშვნელობებს.



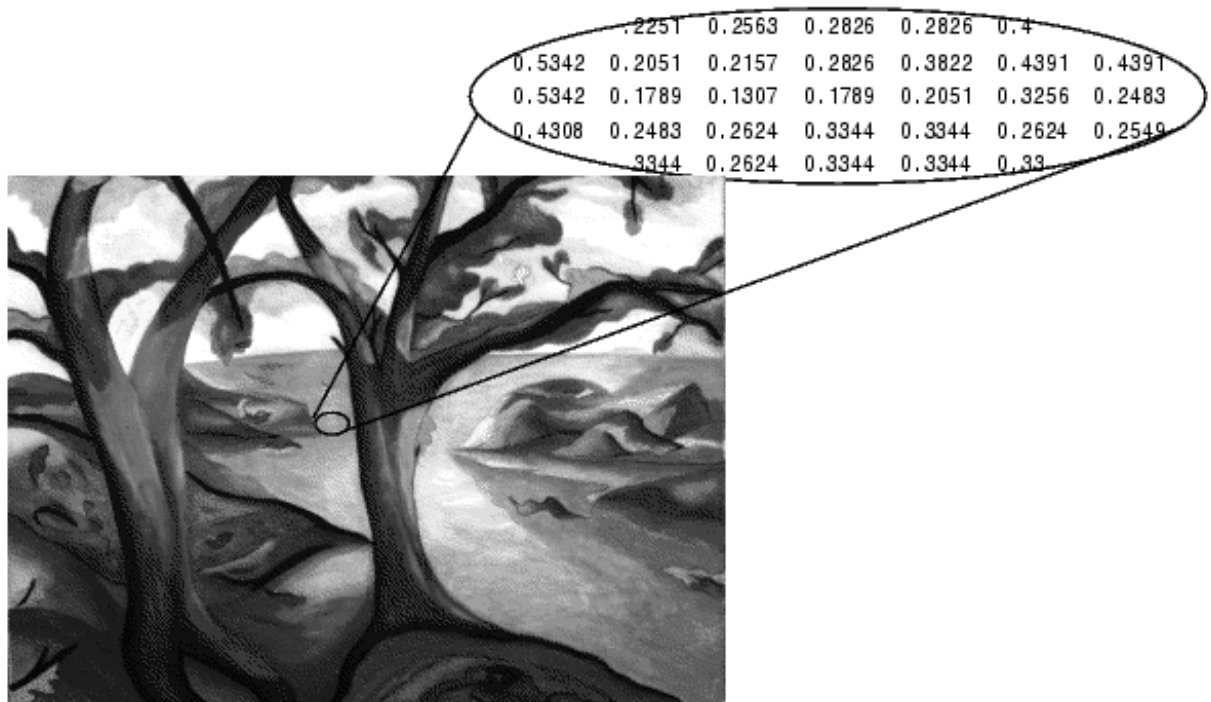
ნახ. 15.1 ორობითი გამონასახი

- ინდექსირებული – მასივი რომელიც შესაძლოა იყოს ლოგიკური, uint8, uint16, single ან double ტიპის. რომლის პიქსელების მნიშვნელობა არის პირდაპირ color map ინდექსები, ხოლო colormap არის $m \times 3$ double ტიპის მატრიცა ინფორმაციით ფერების შესახებ. მას სხვაგვარად ფსევდოფერად გამონასახსაც უწოდებენ.



ნახ. 15.2 ინდექსირებული გამონასახი

- შავ-თეთრი - uint8, uint16, single ან double ტიპის მატრიცა, რომლის პიქსელებიც წარმოდგენილია ინტენსივობებში. single ან double ტიპისათვის პიქსელის მნიშვნელობებია [0,1], uint8 - [0,255], uint16 – [0, 65535].



ნახ. 15.3 ინდექსირებული გამონასახი

- რეალური (truecolor) – სამგანზომილებიანი მატრიცა uint8, uint16, single ან double ტიპის, რომლის პიქსელებიც წარმოდგენილია ინტენსივობებში. ასევე უწოდებენ RGB გამონასახს. single ან double ტიპისათვის პიქსელის მნიშვნელობებია [0,1], uint8 - [0,255], uint16 – [0.65535]



ნახ. 15.4 RGB გამონასახი

15.2 გარდაქმნა გამონასახის ტიპებს შორის

შემდეგი ფუნქციები გამოიყენება გამონასახის ერთი ტიპის სხვა ტიპის გამონასახად გარდასაქმნელად:

dither – შავ-თეთრი გამონასახი ბინარულში, ან რეალური - ინდექსირებულში

gray2ind – შავ-თეთრი - ინდექსირებულში

im2bw – შავ-თეთრი, ინდექსირებული ან რეალური გამონასახი ორობით გამონასახში

ind2gray – ინდექსირებული შავ-თეთრში

ind2grb – ინდექსირებული რეალურ გამონასახში (MATLAB 9.0 version)

mat2gray – მონაცემთა მატრიცა გადაყავს შავ-თეთრ გამონასახში მონაცემთა სკალირების საფუძველზე

rgb2gray – რეალური –შავ-თეთრში

rgb2ind – რეალური – ინდექსირებულში

15.3 გამონასახის იმპორტ-ექსპორტი

გამონასახის იმპორტისა და ექსპორტისათვის **MATLAB** გარემოში, რამდენიმე გზა არსებობს.

- **Tulbox – Image Ecquisition**, რომელიც საშუალებას იძლევა პირდაპირ მიმღებიდან **MATLAB** გარემოში მივიღოთ გამოსახულება.
- **Tulbox – Database** შექმნილია ODBC/JDBC ფორმატის მონაცემთა ბაზაზე სამუშაოდ
- შესაძლებელია სტანდარტული ფორმატის გამონასახზე მუშაობა, როგორცაა: JPEG, TIFF, PNG, HDF, HDF-EOS, FITS, Microsoft Excel, ASCII და ბინარული ფაილები.

გამონასახის გამოძახება **MATLAB** გარემოში ხორციელდება ფუნქციის **imread**, გრაფიკულ ფანჯარაში მისი წარმოდგენა – **imshow**.

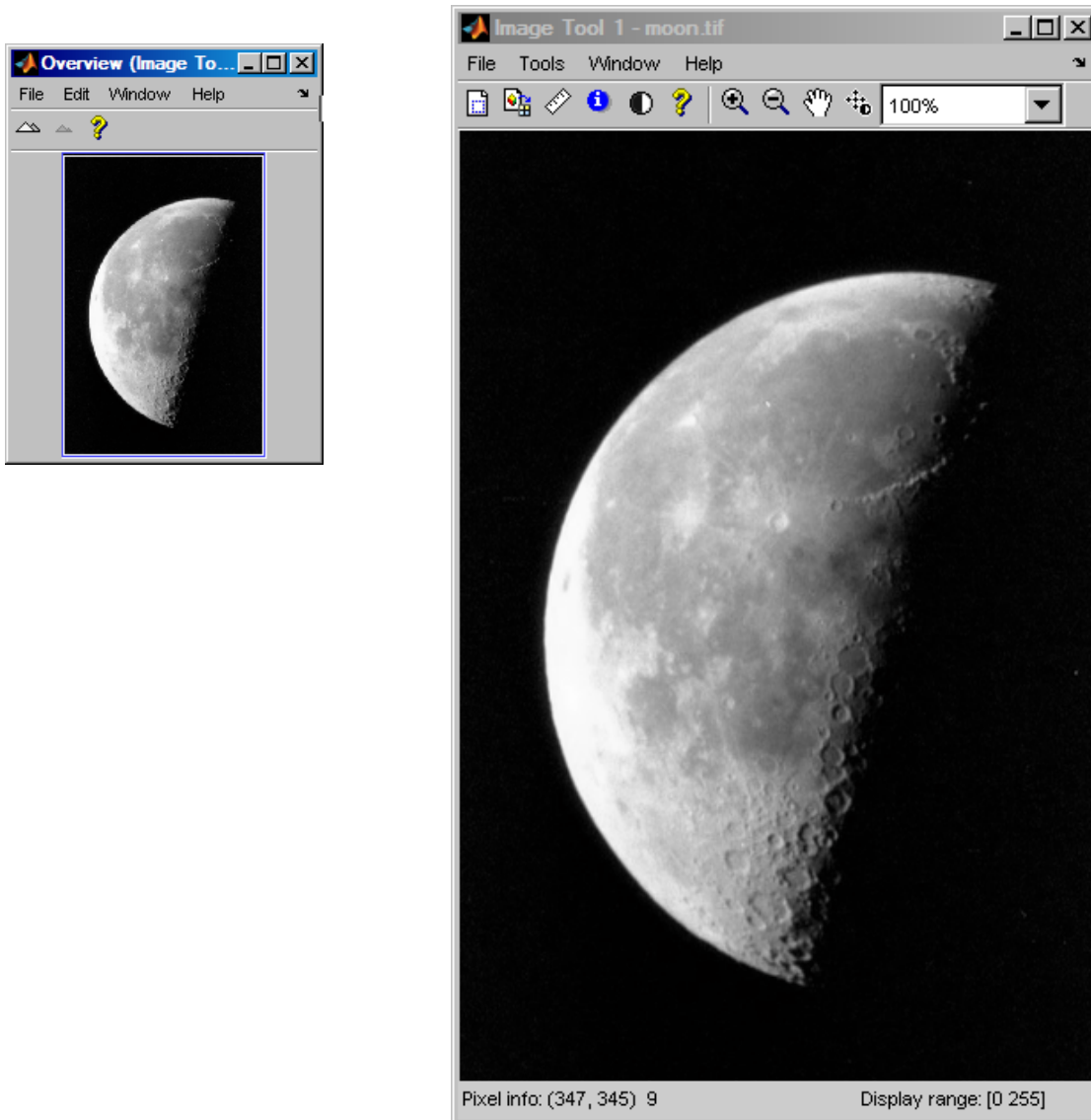
<p><code>A=imread('filename')</code></p> <p><code>[X, map]=imread('filename')</code></p> <p><code>imshow('filename')</code></p>	<p>გამოიძახებს ორფეროვანი(შავ-თეთრი) ან ფერად გამონასახის შემცველ ფაილს და წარმოადგენს მას A მატრიცის სახით. თუ გამონასახი შავ-თეთრია, მივიღებთ ორგანზომილებიან მატრიცას, თუ ფერადია, მივიღებთ სამგანზომილებიან მატრიცას. მატრიცის კლასი დამოკიდებულია გამოძახებული ფაილის ფორმატზე. გამოიძახებს ფერადი, ინდექსირებული გამონასახის შემცველ ფაილს და წარმოადგენს მას X მატრიცის სახით, რომლის შესაბამის ინფორმაციას ფერთა შესახებ წარმოდგენილია map მატრიცის სახით წარმოადგენს გამონასახის შემცველ ფაილს გრაფიკულ ფანჯარაში</p>
---	--

იმისათვის, რომ ვნახოთ თუ როგორი სახით წარმოადგენს გამონასახს ფუნქცია `imshow` მივმართოთ ფუნქციას `whos`:

```
imread('moon.tif');
whos
  Name      Size      Bytes  Class
  ans      537x358    192246  uint8 array
Grand total is 192246 elements using 192246 bytes
```

- ფუნქცია **imtool** გამონასახის გრაფიკულ ფანჯარაში წარმოადგენსთან ერთად საშუალებას გვაძლევს გამოვიყენოთ ასევე Pixel Region tool, the Image Information tool, და the Adjust Contrast tool:

```
imtool('moon.tif')
```



ნახ. 15.5 imtool

15.4 არითმეტიკული ოპერაციები გამონასახზე

მას შემდეგ, რაც გამონასახი გამოძახებულ იქნება **MATLAB** გარემოში ბრძანებით **imread**, შესაძლებელია მიღებულ ინდექსირებულ გამონასახზე არითმეტიკული ოპერაციების წარმოება. გამონასახები ერთიდაიგივე ზომის და კლასის უნდა იყოს. მათემატიკური ოპერაციები სრულდება შესაბამის ელემენტებზე.

`imadd`

`imdivide`

`imsubtract`

`immultiply`

`imabsdiff`

გამონასახების შეკრება

გამონასახების გაყოფა

გამონასახების გამოკლება

გამონასახების გამრავლება

გვაძლევს ახალ მატრიცას, რომლის ელემენტებიც არგუმენტთა შესაბამისი ელემენტების სხვაობათა

აბსოლუტურ სიდიდეს წარმოადგენს

ამ ფუნქციითა არგუმენტები ერთნაირი ზომის და კლასის მატრიცებია, ან მეორე ელემენტი შესაძლოა იყოს double scalar კლასის.

მაგალითად:

```
I = imread('rice.png');
J = imread('cameraman.tif');
K = imadd(I,J);
figure, imshow(K)
```

ან:

```
I = imread('moon.tif');
J = immultiply(I,0.5);
figure, imshow(I), figure, imshow(J)
```

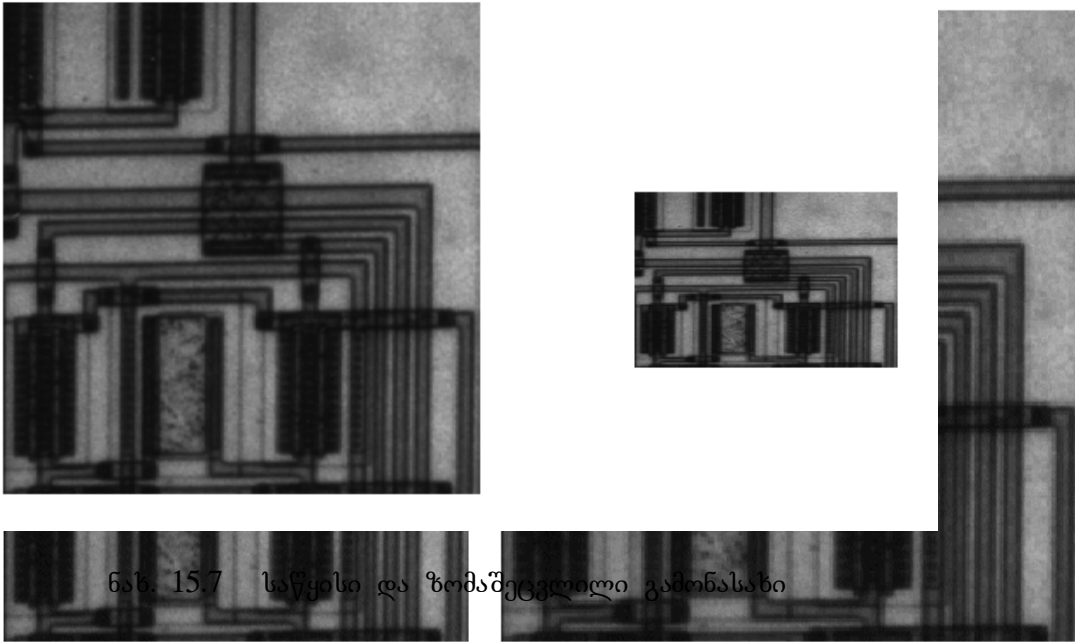
15.5 გამონასახის ზომის შეცვლა

MATLAB საშუალებით შესაძლებელია გამონასახის ზომის შეცვლა ფუნქციის **imresize**. განვიხილოთ ამ ფუნქციის გამოყენების რამდენიმე მაგალითი.

- გამონასახის ზომის შეცვლა გადიდების ფაქტორის მითითებით

```
I = imread('circuit.tif');
J = imresize(I,1.25);
imshow(I)
figure, imshow(J)
```

ბრძანებათა ეს მწკრივი გამოიძახებს **MATLAB** გარემოში გამონასახს და მოგვცემს საწყისი გამონასახის ინდექსირებულ მატრიცას **I**, 1.25 ჯერ გადიდებული ახალი გამონასახის ინდექსირებულ მატრიცას **J**, საწყისი და ახალი გამონასახის გრაფიკულ გამოსახულებებს ნახ. 15.6.



ნახ. 15.7 საწყისი და ზომაშეცვლილი გამონასახი

ნახ. 15.6 საწყისი და გადიდებული გამონასახი

- გამონასახის ზომის შეცვლა სასურველი ზომის მითითებით

```
I = imread('circuit.tif');
J = imresize(I,[100 150]);
imshow(I)
figure, imshow(J)
```

მივიღებთ საწყისი და ზომაშეცვლილ გამონასახებს ნახ. 15.7

- გამონასახის ზომის შეცვლა ინტერპოლირების მეთოდის მითითებით

```
I = imread('circuit.tif');
J = imresize(I,[100 300],'bilinear');
imshow(I)
figure, imshow(J)
```

ფუნქციის მესამე არგუმენტი მიუთითებს ინტერპოლირების მეთოდს, რომელიც უნდა იქნას გამოყენებული ახალი გამონასახის შესაქმნელად. თუ არგუმენტი მითითებული არ არის ინტერპოლირება ხდება მეზობელი ელემენტების მეთოდით, 'bilinear' - ინტერპოლირება ორი წრფივი ფუნქციის გამოყენებით, 'bicubic' - ინტერპოლირება ორი კუბური ფუნქციის გამოყენებით.

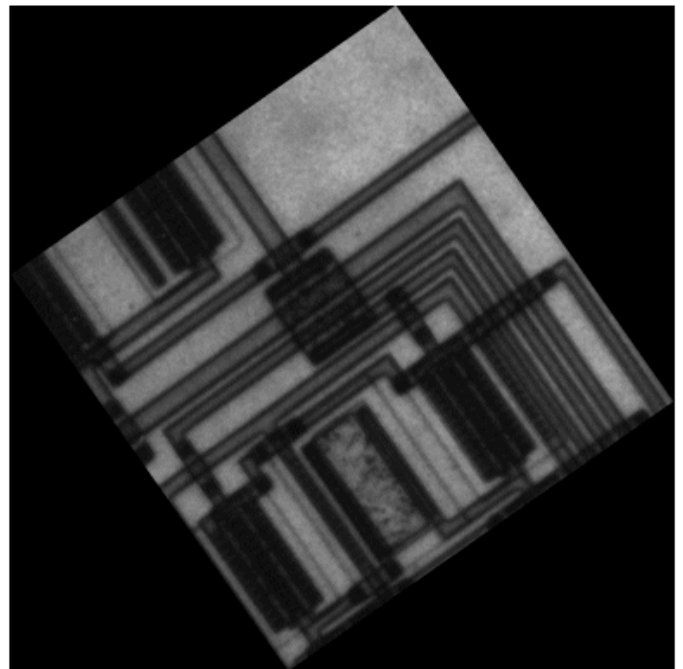
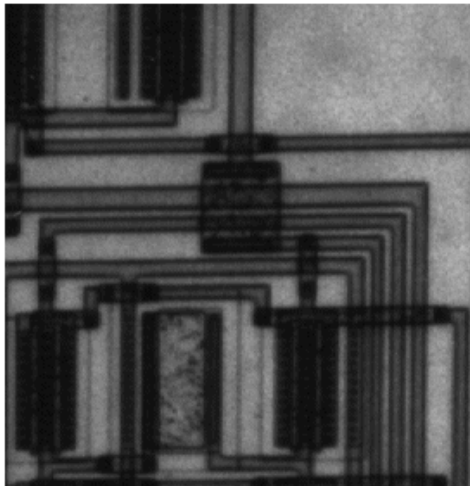
15.6 გამონასახის მობრუნება

MATLAB საშუალებით შესაძლებელია გამონასახის მობრუნება ფუნქციის `imrotate`. განვიხილოთ ამ ფუნქციის გამოყენების მაგალითი.

- გამონასახის მობრუნება


```
I = imread('circuit.tif');
J = imrotate(I,35,'bilinear');
imshow(I)
figure, imshow(J)
```

ბრძანებთა ეს მწკრივი გამოიძახებს MATLAB გარემოში გამონასახს და მოგვცემს საწყისი გამონასახის ინდექსირებულ მატრიცას I, 35 გრადუსით მობრუნებული ახალი გამონასახის ინდექსირებულ მატრიცას J, საწყისი და ახალი გამონასახის გრაფიკულ გამოსახულებებს ნახ. 15.8.



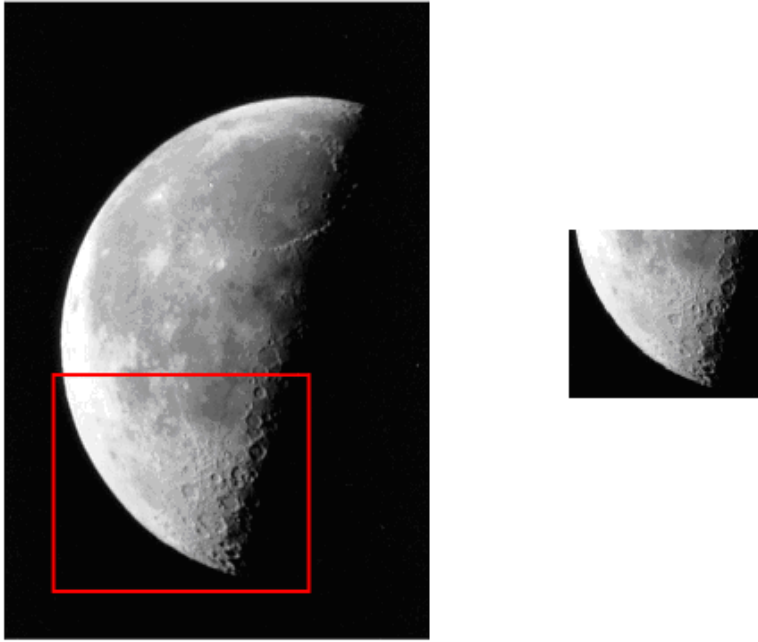
ნახ. 15.8 საწყისი და შემობრუნებული გამონასახი

15.7 დეტალის ამოჭრა გამონასახიდან

MATLAB-ში დეტალის ამოსაჭრელად გამოიყენება ფუნქცია `imcrop`. განვიხილოთ ამ ფუნქციის გამოყენების მაგალითი.

```
imshow moon.tif
I = imcrop;
imshow(I);
```

ფუნქცია `imcrop` ეკრანზე მოიხმობს გამონასახს და დაელოდება მომხმარებელს მაუზის საშუალებით მონიშნოს სასურველი დეტალი გამონასახიდან. მივიღებთ ამოკვეთილი დეტალის გრაფიკულ გამოსახულებას ნახ. 15.9.



ნახ. 15.9 საწყისი გამონასახი და ამოკვეთილი ლეტალი

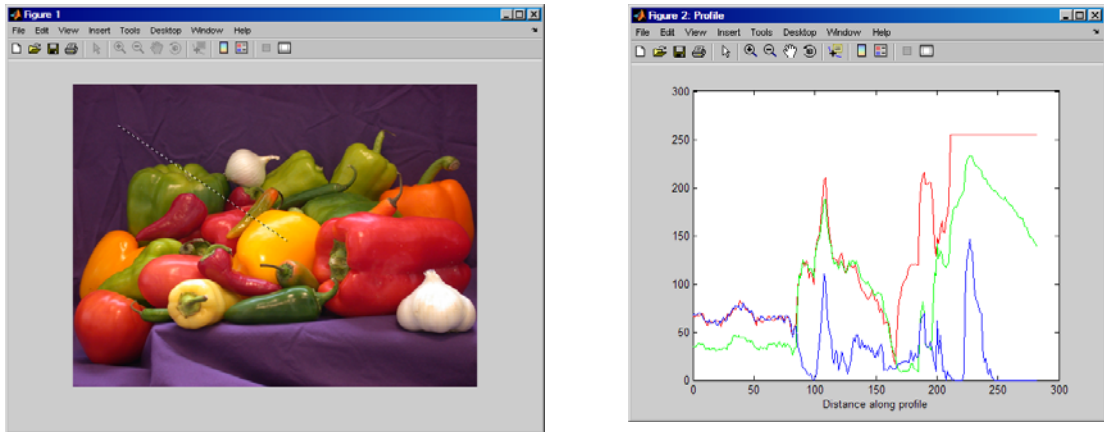
15.8 ინფორმაცია გამონასახის შესახებ და სტატისტიკური ანალიზი

MATLAB გვაძლევს საშუალებას მივიღოთ ინფორმაცია იმ სიდედეთა შესახებ, რაც ქმნის გამონასახს და შევასრულოთ ამ მონაცემთა სტატისტიკური ანალიზი.

- ინფორმაცია გამონასახის შემადგენელი პიქსელის შესახებ. ფუნქცია **impixel** იმისათვის, რომ მივიღოთ ინფორმაცია პიქსელის შესახებ, უნდა გამოვიძახოთ გამონასახი `imshow canoe.tif`. ამის შემდეგ ვისარგებლოთ ბრძანებით: `vals = impixel`, რაც საშუალებას მოგვცემს გამონასახზე მოვნიშნოთ ჩვენთვის საინტერესო პიქსელები. შერჩევის პროცესი სრულდება `return` კლავისის საშუალებით. ინფორმაცია მონიშნული პიქსელების შესახებ შეინახება ცვლადის- `vals` სახით.
- **improfile** – გამოითვლის და ააგებს ინტენსივობათა განაწილებას გამონასახზე მითითებული მონაკვეთის გასწვრივ. მონაკვეთი შეგვიძლია განვსაზღვროთ მაუსის საშუალებით, ან მივუთითოთ კოორდინატები, როგორც შემავალი არგუმენტები. ფერადი გამონასახის შემთხვევაში ფუნქცია ააგებს ინტენსივობების მნიშვნელობებს შესაბამის ფერებში.

შემდეგი ბრძანებები **MATLAB**-ში საშუალებას მოგვცემს გამონასახი გამოვიტანოთ გრაფიკულ ფანჯარაში და მოვნიშნოთ მონაკვეთი, რომლის გასწვრივაც გვინტერესებს პროფილის აგება:

```
imshow peppers.png
improfile
```

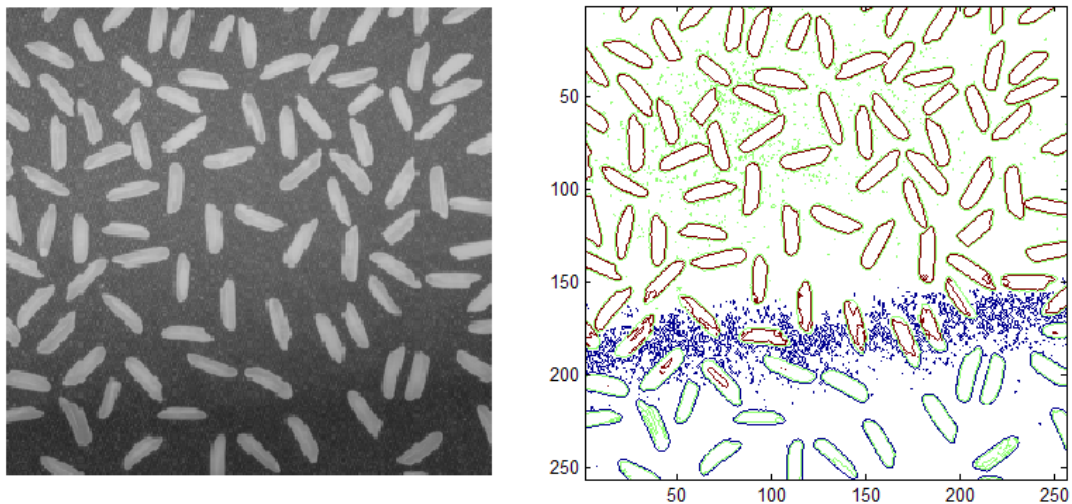


ნახ. 15.10 improfile

- გამონასახის კონტურული გრაფიკი. ფუნქცია **imcontour**

```
I = imread('rice.png');
imshow(I)
figure, imcontour(I,3)
```

ბრძანებთა მწკრივი გამოიძახებს, გრაფიკულ ფანჯარაში გამოიტანს გამონასახს და ახალ გრაფიკულ ფანჯარაში მოგვცემს გამონასახის კონტურულ გრაფიკს.

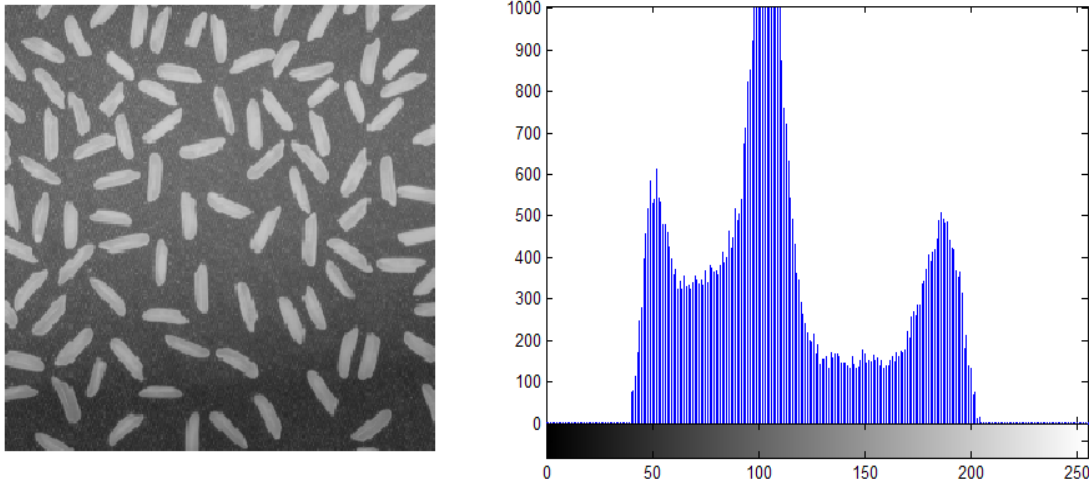


ნახ. 15.11 საწყისი გამონასახი და კონტურული გრაფიკი

- გამონასახის ჰისტოგრამა. ფუნქცია **imhist**

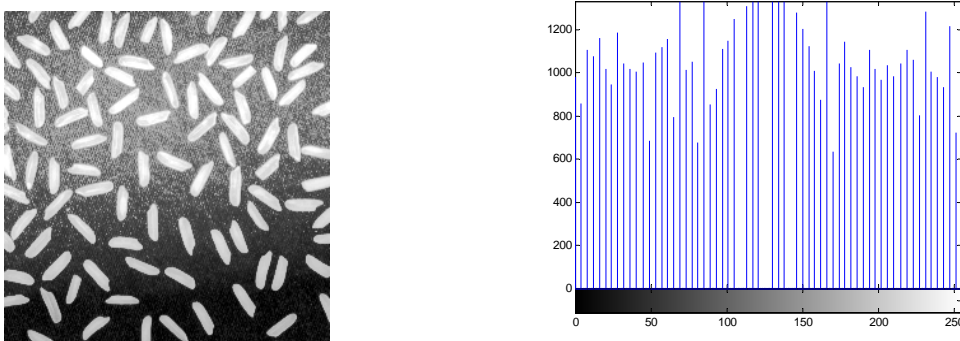
```
I = imread('rice.png');
imshow(I)
figure, imhist(I)
```

ბრძანებთა მწკრივი გამოიძახებს, გრაფიკულ ფანჯარაში გამოიტანს გამონასახს და ახალ გრაფიკულ ფანჯარაში მოგვცემს გამონასახის ჰისტოგრამას.



ნახ. 15.12 საწყისი გამონასახი და ჰისტოგრამა

არსებობს რამდენიმე გზა გამონასახის კონტრასტის შესაცვლელად. ერთ-ერთი მათგანია ფუნქცია **histeq** რაც საშუალებას იძლევა განვაგრძოთ ინტენსივობა თანაბრად მთელ გამონასახზე. ამ პროცესს ჰისტოგრამის გათანაბრებას უწოდებენ (histogram equalization).



ნახ. 15.13 გათანაბრებული გამონასახი და ჰისტოგრამა

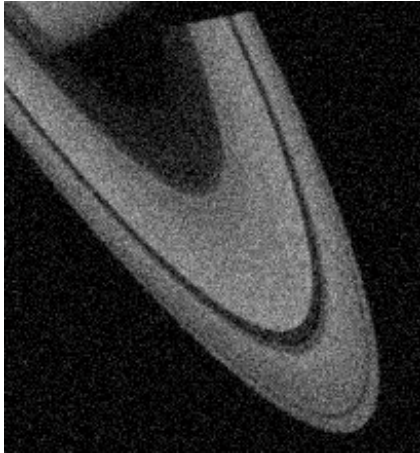
- გამონასახის სტატისტიკური ანალიზი
ფუნქციები **mean2**, **std2** გამოითვლის გამონასახის შემქმნელი მატრიცის ელემენტების მნიშვნელობათა საშუალოსა და სტანდარტულ გადახრას, ხოლო **corr2** ერთნაირი ზომის ორი მატრიცის კორელაციის კოეფიციენტს გამოითვლის.
- ციფრული გამონასახი შესაძლოა დამახინჯებული იყო სხვადასხვა მიზეზით გამოწვეული ხმაურით. **MATLAB** გააჩნია ხმაურის გამორიცხვის სხვადასხვა საშუალება, როგორცაა წრფივი, მდლიანური და ადაპტური ფილტრაცია. ფუნქცია **wiener2** ანალიზებს გამონასახს და ხმაურის ფილტრაციას იმის მიხედვით შეასრულებს. ბრძანებათა შემდეგი მწკრივი გამოიძახებს გამონასახს, გადაიყვანს მას შავ-თეთრ გამონასახში, დაუმატებს ხმაურს. (გაუსის განაწილება) და შემდეგ გაასუფთავებს გამონასახს ხმაურისაგან **wiener2** ფუნქციის საშუალებით.

```

RGB = imread('saturn.png');
I = rgb2gray(RGB);
J = imnoise(I,'gaussian',0,0.005);
K = wiener2(J,[5 5]);
imshow(J)
figure, imshow(K)

```

•



ნახ. 15.14 გამონასახის გასუფთავება ხმაურისაგან

- გამონასახის ჩაწერა დისკზე

ფუნქცია **imwrite** ჩაწერს გამონასახს მითითებულ ადგილზე.

```
imwrite(I2, 'rice2.png');
```

ხოლო ფუნქცია **iminfo** გვიჩვენებს ინფორმაციას ახლადშექმნილი გამონასახის შესახებ:

```
iminfo('rice2.png ')
```

```
ans =
```

```

Filename: 'rice2.png '
FileModDate: '08-Nov-2010 18:04:56'
FileSize: 49552
Format: 'png'
FormatVersion: []
Width: 256
Height: 256
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: [137 80 78 71 13 10 26 10]
olormap: []
Histogram: []
InterlaceType: 'none'
Transparency: 'none'
SimpleTransparencyData: []

```



```

BackgroundColor: []
RenderingIntent: []
Chromaticities: []
    Gamma: []
    XResolution: []
    YResolution: []
ResolutionUnit: []
    XOffset: []
    YOffset: []
    OffsetUnit: []
SignificantBits: []
ImageModTime: '8 Nov 2010 14:04:56 +0000'
Title: []
Author: []
Description: []
Copyright: []
CreationTime: []
Software: []
Disclaimer: []
Warning: []
Source: []
Comment: []
OtherText: []

```

15.9 MATLAB ბრძანებები და ფუნქციები

<code>A=imread('filename')</code>	გამოიძახებს ორფეროვანი(შავ-თეთრი) გამონასახის შემცველ ფაილს და წარმოადგენს მას A მატრიცის სახით (ინდექსირებული გამონასახი)
<code>[X,map]=imread('filename')</code>	გამოიძახებს ფერად გამონასახის შემცველ ფაილს და წარმოადგენს მას A მატრიცის სახით, რომლის შესაბამის ინფორმაციას ფერთა შესახებ წარმოადგენს map მატრიცის სახით
<code>imshow('filename')</code>	წარმოადგენს გამონასახის შემცველ ფაილს გრაფიკულ ფანჯარაში
<code>B=imresize(A,m)</code>	გვაძლევს A ინდექსირებული გამონასახის m-ჯერ გადიდებულ გამონასახს (თუ $0 < m < 1$, გამონასახი შემცირდება)
<code>imrotate(A,angle)</code>	აბრუნებს A გამონასახს 'angle' გრადუსით საათის ისრის საწინააღმდეგო მიმართულებით
<code>imadd</code>	გამონასახების შეკრება
<code>imdivide</code>	გამონასახების გაყოფა
<code>imadd</code>	გამონასახების შეკრება
<code>imdivide</code>	გამონასახების გაყოფა
<code>imsubtract</code>	გამონასახების გამოკლება
<code>immultiply</code>	გამონასახების გამრავლება
<code>imabsdiff</code>	გვაძლევს ახალ მატრიცას, რომლის ელემენტებიც არგუმენტთა შესაბამისი ელემენტების სხვაობათა

	აბსოლუტურ სიდიდეს წარმოადგენს
imcrop	ამოჭრის დეტალს გამონასახიდან
impxel	იძლევა ინფორმაციას მონიშნული პიქსელის შესახებ
imcontour	გამონასახის კონტურული გრაფიკი
imhist	აგებს გამონასახის ჰისტოგრამას
mean2	mean2 გამოითვლის გამონასახის შემქმნელი მატრიცის ელემენტების მნიშვნელობათა საშუალოს
std2	std2 გამოითვლის გამონასახის შემქმნელი მატრიცის ელემენტების მნიშვნელობათა სტანდარტულ გადახრას,
corr2	გამითვლის ერთნაირი ზომის ორი მატრიცის კორელაციის კოეფიციენტს.
wiener	გამონასახის ხმაურისაგან გასუფთავება
improfile	აგებს ინტენსივობათა პროფილს გამონასახზე მითითებული მონაკვეთის გასწვრივ
dither	გარდაქმნის შავ-თეთრი გამონასახს ბინარულში, ან რეალურს - ინდექსირებულში
gray2ind	გარდაქმნის შავ-თეთრ გამონასახს ინდექსირებულში
im2bw	გარდაქმნის შავ-თეთრ, ინდექსირებულ ან რეალურ გამონასახს ორობით გამონასახში
ind2gray	გარდაქმნის ინდექსირებულ გამონასახს შავ-თეთრში
ind2grb	გარდაქმნის ინდექსირებულ გამონასახს რეალურ გამონასახში
mat2gray	მონაცემთა მატრიცას წარმოადგენს შავ-თეთრ გამონასახის სახით მონაცემთა სკალირების საფუძველზე
rgb2gray	გარდაქმნის რეალურ გამონასახს შავ-თეთრში
rgb2ind	გარდაქმნის რეალურ გამონასახს ინდექსირებულში
imwrite	ჩაწერს გამონასახს მითითებულ ადგილზე
iminfo	გვიჩვენებს ინფორმაციას გამონასახის შესახებ