Εργαστήριο ADICV1-3

Matlab Image Basics, neighbours and boundaries

Κώστας Μαριάς

Octave

File Edit Debug Window Help News

Current Directory: C:\Users\kmarias

File Browser

C:/Users/kmarias

Name

.config
.idlerc
.matplotlib
3D Objects
Contacts
Desktop
Documents
Downloads
Dropbox
Favorites
Google Drive
Links

Workspace

Filter

Name | Class | Dimension | Value

Command History

Filter

y=x^2+1
y=x.^2+1
plot(x,y)
y=x.^2+exp(x);
plot(x,y)
exit
Octave 4.2.1, Thu Feb 22 12:45:17 2018 GMT <unknow

Command Window

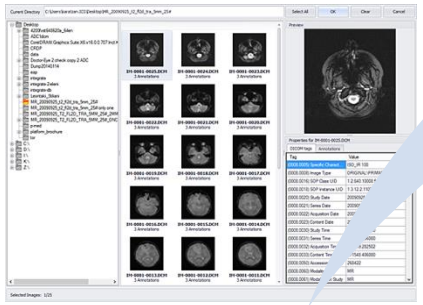
```
GNU Octave, version 4.2.1
Copyright (C) 2017 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-w64-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

>> pkg load image
```

Command Window | Editor | Documentation

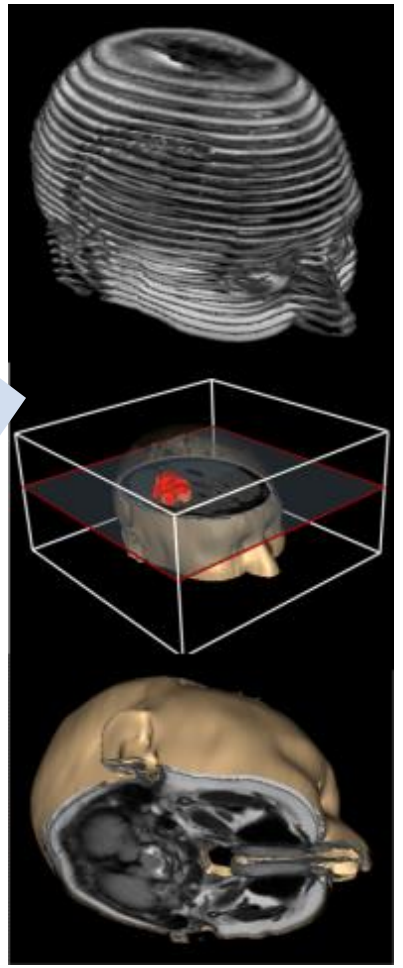
Store

Type here to search

ENG 7:30 PM 2/25/2018



Basic
Matlab
Image
Processing



DrEye

http://biomodeling.ics.forth.gr/

Εικόνες σε MATLAB

- The basic data structure in MATLAB® is the *array*, an ordered set of real or complex elements. This object is naturally suited to the representation of *images*, real-valued ordered sets of color or intensity data.
- MATLAB stores most images as two-dimensional arrays (i.e., matrices), in which each element of the matrix corresponds to a single *pixel* in the displayed image. (Pixel is derived from *picture element* and usually denotes a single dot on a computer display.)
- For example, an image composed of 200 rows and 300 columns of different colored dots would be stored in MATLAB as a 200-by-300 matrix.

https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

Εικόνες σε MATLAB

- Η βασική δομή δεδομένων στο MATLAB® είναι ο πίνακας, ένα ταξινομημένο σύνολο πραγματικών ή σύνθετων στοιχείων. Αυτό είναι φυσικά κατάλληλο για την απεικόνιση εικόνων με πραγματικές τιμές χρώματος ή έντασης.
- Το MATLAB αποθηκεύει τις περισσότερες εικόνες ως δισδιάστατους Πίνακες, όπου κάθε στοιχείο της μήτρας αντιστοιχεί σε ένα μόνο εικονοστοιχείο στην απεικονιζόμενη εικόνα.
- (Το Pixel προέρχεται από το στοιχείο εικόνας και συνήθως υποδηλώνει μια μόνο κουκκίδα σε μια οθόνη υπολογιστή.) Για παράδειγμα, μια εικόνα που αποτελείται από 200 σειρές και 300 στήλες από διαφορετικές έγχρωμες κουκίδες θα αποθηκεύεται σε MATLAB ως πίνακας 200 x 300.
  https://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

Η ψηφιακή εικόνα…

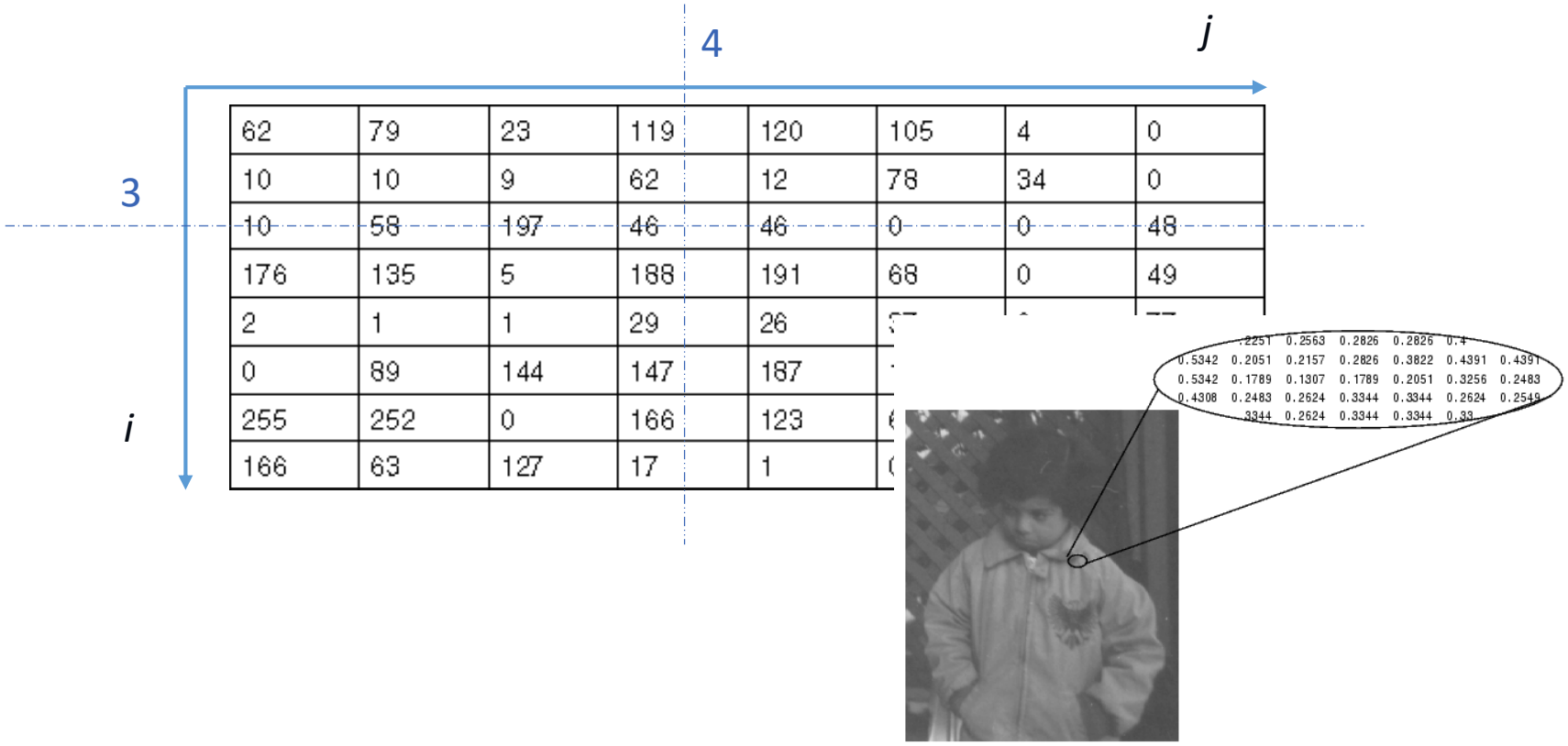
- Μπορούμε να σκεφτούμε μια εικόνα ως συνάρτηση f, από R^2 -> R :
  - $f(x, y)$ δίνει την ένταση στο (x, y)
  - Ρεαλιστικά , περιμένουμε την εικόνα για να οριστεί μόνο σε ένα ορθογώνιο , με έναν πεπερασμένο εύρος:

- Μια έγχρωμη εικόνα είναι μια διανυσματική συνάρτηση τριών εικόνων R, G, B

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Η ψηφιακή εικόνα…

- Χρησιμοποιούμε διακριτές τιμές στις εικόνες (π.χ. δειγματοληψία)
- Η εικόνα μπορεί τώρα να αναπαρασταθεί ως ένας πίνακας με ακέραιες τιμές
- Η Ένταση στο *i*=3, *j*=4 είναι $f(3, 4) = 46$



Display Εικόνων

Για να εμφανιστεί μια εικόνα στην MATLAB χρησιμοποιούμε τη συνάρτηση imshow, με την σύνταξη:

>imshow(f, G)

f είναι ένας πίνακας εικόνας και G είναι ο αριθμός των επιπέδων έντασης (intensity levels) για το display. Αν παραλειφθεί το G το default είναι 256 levels.

Με την σύνταξη:

>imshow(f, [low high])

- Όλες οι τιμές < από το low, εμφανίζονται με μαύρο
- Όλες οι τιμές > από το high, εμφανίζονται με λευκό
- Οι ενδιάμεσες τιμές με αποχρώσεις του γκρί (8bit->256 αποχρώσεις)
- Με αυτό τον τρόπο μπορούμε να απεικονίσουμε εικόνες με χαμηλό εύρος τιμών (dynamic range) ή με θετικές και αρνητικές τιμές πίνακα.

Read and write image data, get information about contents of image files

- Οι βασικές εντολές της Matlab για να διαβάσουμε/αποθηκεύσουμε εικόνες είναι:

Functions

`imread`	Read image from graphics file
`imwrite`	Write image to graphics file
`imfinfo`	Information about graphics file

Διαβάζοντας εικόνες από διαφορετικά format

- Ας χρησιμοποιήσουμε την συνάρτηση imread. Με το παρακάτω παράδειγμα διαβάζουμε μια εικόνα **truecolor στο** workspace **του** MATLAB® **και την αναθέτουμε στην μεταβλητή RGB**.

```
RGB = imread('football.jpg');
figure, imshow(RGB)
```

- Εάν η μορφή αρχείου εικόνας χρησιμοποιεί εικονοστοιχεία 8-bit, το imread αποθηκεύει τα δεδομένα στον χώρο εργασίας ως πίνακα uint8. Για μορφές αρχείων που υποστηρίζουν δεδομένα 16 bit, όπως PNG και TIFF, το imread δημιουργεί ένα πίνακα uint16.

Διαβάζοντας εικόνες από διαφορετικά format

- Το imread χρησιμοποιεί δύο μεταβλητές για να αποθηκεύσει μια ευρετηριασμένη εικόνα (indexed image) στον χώρο εργασίας: μία για την εικόνα και μία για το αντίστοιχο colormap.
- Το imread διαβάζει πάντα το colormap σε μια μήτρα double, παρόλο που η ίδια η συστοιχία εικόνων μπορεί να είναι της κλάσης uint8 ή uint16.
- [X,map] = imread(‘kids.tif');
- Το imread υποστηρίζει πολλές μορφές κοινών αρχείων γραφικών, όπως Bitmap Microsoft® Windows® (BMP), Graphic Interchange Format (GIF), JPEG (Joint Photographic Experts Group), Portable Network Graphics (PNG) και μορφές αρχείου ετικετών ετικετών (TIFF) .

Αποθηκεύοντας εικόνες από MATLAB® workspace σε graphics file

Για να εξάγετε δεδομένα εικόνας από τον χώρο εργασίας MATLAB® σε ένα αρχείο γραφικών σε μία από τις υποστηριζόμενες μορφές αρχείων γραφικών, χρησιμοποιήστε τη συνάρτηση εγγραφής imwrite .

Αυτό το παράδειγμα φορτώνει την ευρετηριωμένη εικόνα X από ένα αρχείο MAT μαζί με τον σχετικό χάρτη χρωμάτων και στη συνέχεια εξάγει την εικόνα ως αρχείο bitmap (BMP).

```
[X,map] = imread('kids.tif');
imwrite(X,map,‘kids.bmp')
```

Αποθηκεύοντας εικόνες από MATLAB® workspace σε graphics file

Κατά την εγγραφή δυαδικών αρχείων, το MATLAB ορίζει το πεδίο ColorType σε 'grayscale'.

Καθορίστε την κατηγορία αποθήκευσης των αρχείων εξόδου

Το imwrite χρησιμοποιεί τους ακόλουθους κανόνες για να καθορίσει την κλάση αποθήκευσης που χρησιμοποιείται στην εικόνα εξόδου.

Storage Class of Image	Storage Class of Output Image File
`logical`	If the output image file format supports 1-bit images, `imwrite` creates a 1-bit image file. If the output image file format specified does not support 1-bit images, `imwrite` exports the image data as a `uint8` grayscale image.
`uint8`	If the output image file format supports unsigned 8-bit images, `imwrite` creates an unsigned 8-bit image file.
`uint16`	If the output image file format supports unsigned 16-bit images (PNG or TIFF), `imwrite` creates an unsigned 16-bit image file. If the output image file format does not support 16-bit images, `imwrite` scales the image data to class `uint8` and creates an 8-bit image file.
`int16`	Partially supported; depends on file format.
`single`	Partially supported; depends on file format.
`double`	MATLAB scales the image data to `uint8` and creates an 8-bit image file, because most image file formats use 8 bits.

Δημιουργώντας έναν πίνακα-εικόνα σε Matlab

- A=[1 2 3 4;5 6 7 8; 9 10 11 12;13 14 15 16]
- figure, imshow(A,[])
- B=A+10
- C=A-10
- Image=[A B C; C B A; B A C]
- min(min(A))
- min(min(Image))
- max(max(Image))
- Image2=uint8(Image)
- figure, imshow(Image,[]), title('my first image')
- figure, imshow(Image2,[]), title('my second image')

Δημιουργώντας έναν πίνακα-εικόνα σε Matlab

```
A=[1 2 3 4;5 6 7 8; 9 10 11 12;13 14 15 16]
figure, imshow(A,[])
B=A+10
C=A-10
Image=[A B C; C B A; B A C]
min(min(A))
min(min(Image))
max(max(Image))
Image2=uint8(Image)
figure, imshow(Image,[]), title('my first image')
figure, imshow(Image2,[]), title('my second image')
imwrite(Image2,'1.gif')
A=imread('1.gif')
```

Διαβάζοντας εικόνες σε διαφορετικά format

```
RGB = imread('football.jpg');
I = imread('cameraman.tif');
[X,map] = imread('trees.tif');
figure, imshow(X)
figure, imshow(X,map)
figure, imshow(X,map), colorbar
```

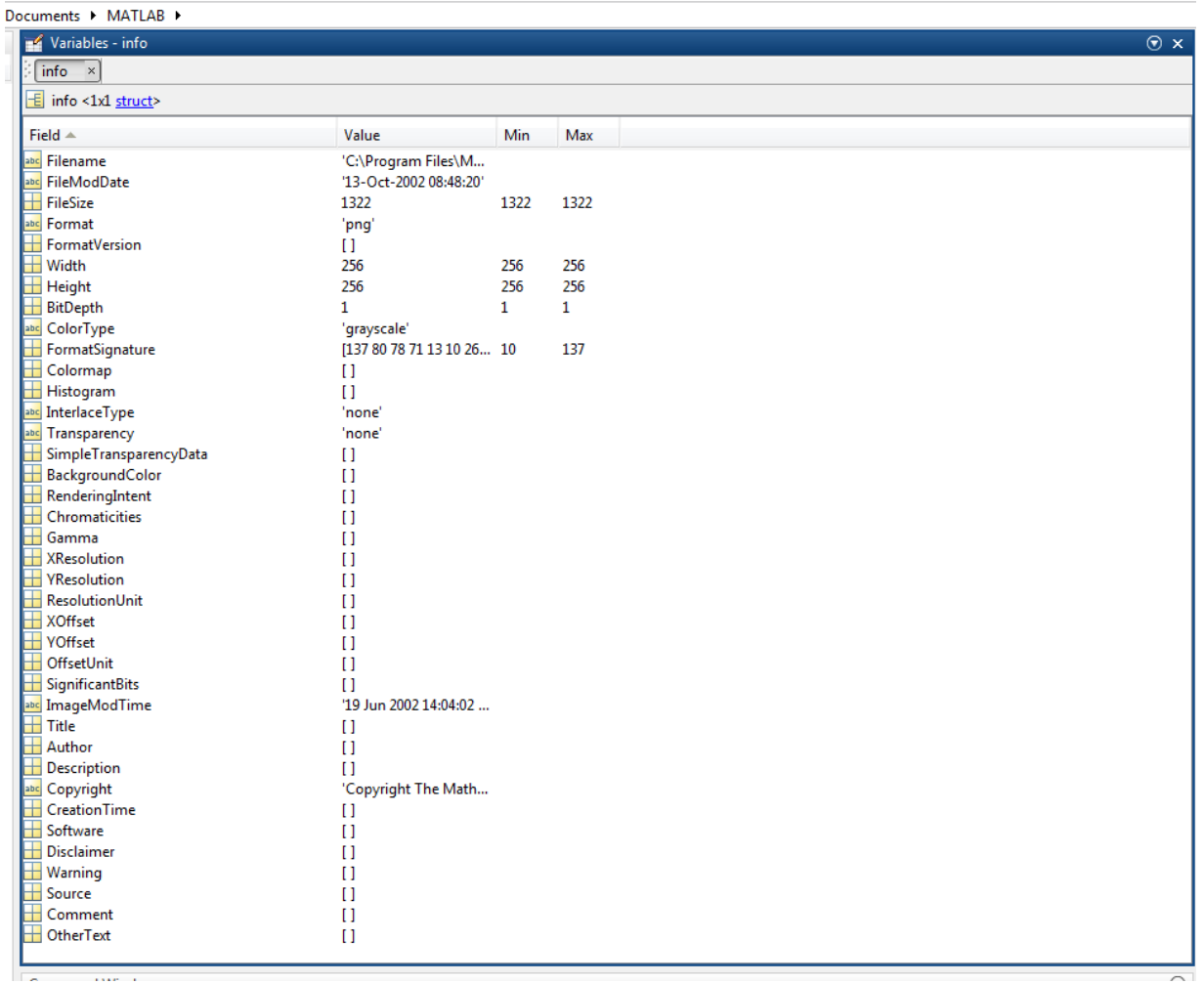
Bits binary εικόνας

```
 I=imread('text.png');
figure, imshow(I)
info=imfinfo('text.png');
 info.BitDepth
```

Bits grayscale εικόνας

```
info=imfinfo('cameraman.tif');
info.BitDepth
info
```

Βρείτε τα bits για την εικόνα 'cameraman.tif'



Bits grayscale εικόνας

```
BW = imread('text.png');
whos
figure, imshow(BW)

info = imfinfo('text.png');
info.BitDepth

imwrite(BW,'test.tif');
info2 = imfinfo('test.tif');
info2.BitDepth
```

Διαβάστε και οπτικοποιήστε την εικόνα text.png βρείτε τα bits του εικονοστοιχείου. Στη συνέχεια σώστε την ως αρχείο .tif και δείτε αν τα bits παραμένουν τα ίδια καθώς και τι έχει αλλάξει???

Άσκηση

```
I=imread('clown.bmp');
imwrite(I,'clown.tif')
clear
I2=imread('clown.tif');
figure, imshow(I2)
info=imfinfo('clown.tif');
info.BitDepth
```

```
[X,map] = imread('clown.bmp');
imwrite(X,map,'clown.tif')
[I,map]=imread('clown.tif');
figure, imshow(I,map)
info=imfinfo('clown.tif');
info.BitDepth
```

Read clown.bmp save it as tif image, read it again and visualize the images. See if they still have the same bits

```
info = imfinfo('…tif');
info.BitDepth
```

Images in MATLAB

- Ορισμένες εικόνες, όπως οι εικόνες truecolor, απαιτούν μια τρισδιάστατη διάταξη πινάκων, όπου ο πρώτος πίνακας αντιπροσωπεύει τις εντάσεις κόκκινων εικονοστοιχείων, ο δεύτερος τις εντάσεις των πράσινων εικονοστοιχείων ο τρίτοσ αντιπροσωπεύει τις εντάσεις των μπλε εικονοστοιχείων.
- Αυτή η συνθήκη κάνει την εργασία με εικόνες σε MATLAB παρόμοια με την εργασία με οποιοδήποτε άλλο τύπο δεδομένων matrix, και κάνει την πλήρη ισχύ του MATLAB διαθέσιμη για εφαρμογές επεξεργασίας εικόνας.

Image Types σε Matlab

• **Gray-scale Imagew**: Είναι πίνακες των οποίων οι τιμές αντικατοπτρίζουν τις αποχρώσεις του γκρίζου Όταν τα στοιχεία μιας gray-scale εικόνας είναι class uint8 or uint16, έχουν ακέραιες τιμές στα διαστήματα [0, 255] ή [0, 65535], αντίστοιχα. Αν οι τιμές είναι class double ή Single, οι τιμές είναι αριθμοί στο σύστημα κινητής υποδιαστολής . Οι τιμές Values σε double και single gray-scale εικόνες συνήθως ανακλιμακώνονται στο εύρος τιμών [0,1]

• **Binary images**: Οι δυαδικές εικόνες MATLAB είναι *logical* πίνακες τιμών 0 και 1. Αν ο πίνακας A έχει τιμές μόνο 0 και 1 αλλά είναι data class π.χ. uint8 δεν είναι binary image στη MATLAB και για να γίνει:

B = logical(A) Αν ο A έχει και άλλες τιμές εκτός από 0,1 τότε η συνάρτηση logical ολες τις μη μηδενικές τιμές σε logical 1s και όλες τις μηδενικές σε logical 0s. Με σχεσιακούς και λογικούς τελεστές πετυχένουμε τα ίδια αποτελέσματα.

Με τη συνάρτηση : islogical(C) Αν ο C είναι logical array, επιστρέφει 1, αλλιώς 0. Δοκιμάστε:

A=[0 0 5;1 0 1; 3 2 1]

B = logical(A)

islogical(A)

islogical(B)

Επίσης έχουμε **Indexed images** και **RGB** images που θα ασχοληθούμε αργότερα.

Διαβάζοντας εικόνες από διαφορετικά format Truecolor εικόνες

- Use the imread function. This example reads **a truecolor image** into the MATLAB® workspace **as the variable RGB**.
- RGB = imread('football.jpg');
- figure, imshow(RGB)
- If the image file format uses 8-bit pixels, imread stores the data in the workspace as a uint8 array. For file formats that support 16-bit data, such as PNG and TIFF, imread creates a uint16 array.

- https://www.mathworks.com/company/newsletters/articles/how-matlab-represents-pixel-colors.html

Διαβάζοντας εικόνες από διαφορετικά format: Indexed εικόνες

- imread uses two variables to store an indexed image in the workspace: one for the image and another for its associated colormap. imread always reads the colormap into a matrix of class double, even though the image array itself may be of class uint8 or uint16.
- [X,map] = imread('trees.bmp');
- imread supports many common graphics file formats, such as Microsoft® Windows® Bitmap (BMP), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG), and Tagged Image File Format (TIFF) formats. For the latest information concerning the bit depths and/or image formats supported, see imread and imformats.

- https://www.mathworks.com/company/newsletters/articles/how-matlab-represents-pixel-colors.html

Αποθηκεύοντας εικόνες από MATLAB® workspace σε graphics file

Note When writing binary files, MATLAB sets the ColorType field to 'grayscale'.

- Determine Storage Class of Output Files
- imwrite uses the following rules to determine the storage class used in the output image.

Storage Class of Image	Storage Class of Output Image File
`logical`	If the output image file format supports 1-bit images, `imwrite` creates a 1-bit image file. If the output image file format specified does not support 1-bit images, `imwrite` exports the image data as a `uint8` grayscale image.
`uint8`	If the output image file format supports unsigned 8-bit images, `imwrite` creates an unsigned 8-bit image file.
`uint16`	If the output image file format supports unsigned 16-bit images (PNG or TIFF), `imwrite` creates an unsigned 16-bit image file. If the output image file format does not support 16-bit images, `imwrite` scales the image data to class `uint8` and creates an 8-bit image file.
`int16`	Partially supported; depends on file format.
`single`	Partially supported; depends on file format.
`double`	MATLAB scales the image data to `uint8` and creates an 8-bit image file, because most image file formats use 8 bits.

Data types in Matlab

- Elements in Matlab matrices may have a number of different numeric data types; the most common are listed in the next table

Data type	Description	Range
`int8`	8-bit integer	-128 — 127
`uint8`	8-bit unsigned integer	0 — 255
`int16`	16-bit integer	-32768 — 32767
`uint16`	16-bit unsigned integer	0 — 65535
`double`	Double precision real number	Machine specific

Data classes

double	Double-precision, floating-point number is in the approximate range $\pm 10^{308}$ (8 bytes per element)
single	Single-precision floating-point numbers with values in the approximate range $\pm 10^{38}$ (4 bytes per element).
uint8	Unsigned 8-bit Integers in the range [0, 255] (1 byte per element)
uint16	Unsigned 16-bit Integers In the range [0, 65535] (2 bytes per element)
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element)
int8	Signed 8-bit integers in the range [-128,127] (1 byte per element)
int16	Signed 16-bit integers in the range [-32768,32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648,2147483647] (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

Converting between Classes

Η γενική σύνταξη για μετατροπή κλάσεων είναι **B = class_name(A)**

Π.χ. αν ο πίνακας **A είναι class uint8** μπορούμε να φτιάξουμε τον αντίστοιχο double-precision array, B με την εντολή B = double (A).

Αν ο C είναι ένας πίνακας class double με τιμές στο διάστημα [0, 255] , μπορεί να μετατραπεί σε uint8 με την εντολή D = uint8 (C) .

Αν ένας πίνακας έχει τιμές έξω από το διάστημα [0,255] και μετατραπεί σε class uint8 όπως πριν, η MATLAB μετατρέπει σε 0 όλες τις τιμές < 0, σε 255 όλες τις τιμές >255. Οι ενδιάμεσες τιμές στρογγυλοποιούνται στον κοντινότερο ακέραιο.

Οπότε πρέπει να γίνει σωστή μετατροπή τιμών (scaling) ενός πίνακα double array έτσι ώστε όλα τα στοιχεία του να είναι το διάστημα [0, 255] ΠΡΙΝ ΜΕΤΑΤΡΑΠΕΙ ΣΕ uint8!!!!

Matlab εργαλειοθήκη για τη μετατροπή εικόνας από ένα class σε άλλο

Name	Converts Input to:	input
im2uint8	uint8	logical, uint8, uint16, int16, single, and double
im2uint16	uint16	logical, uint8, uint16, int16, single, and double
im2double	double	logical, uint8, uint16, int16, single, and double
im2single	single	logical, uint8, uint16, int16, single, and double
mat2gray	double in the range [0, 1]	logical, uint8, int8, uint16, int16, uint32, int32, single, and double
im2bw	logical	uint8, uint16,int16, single, and double

Αριθμητικοί τελεστές πινάκων

Τελεστής	Όνομα	Εξηγήσεις
+	Array and matrix addition	a + b, A + B, or a + A.
-	Array and matrix subtraction	a - b, A - B, A - a, or a - A.
.*	Array multiplication	Cv=A.*B,C(I, J) =A(I, J)*B(1, J) .
*	Matrix multiplication	A*B (πολλαπλασιασμός πινάκων)
. /	Array right division	C=A./B, C(1, J) =A(i, j)/B(i, j)
. \	Array left division	C=A.\B, C(1, J) =B(j, j)/A(i, j)
/	Matrix right division	A/ B is the preferred way to compute A* inv (B).
\	Matrix left division	A\B is the preferred way to compute inv(A) *B.
.^	Array power	IfC=A.'B,thenC(I, J) =A(I, J) ^B(I, J) .
.’	Transpose	A.' , standard vector and matrix transpose.

Πολλαπλασιασμός πινάκων στη matlab

- A=[a1 a2; a3 a4]
- and B=[b1 b2; b3 b4]
- The *array product* of A and B:
- A. *B = [a1b1 a2b2; a3b3 a4b4]
- whereas the *matrix product:*
- A*B=[a1b1+a2b3 a1b2+a2b4 ; a3b1 + a4b3 a3b2 + a4b4]

- Διάταξη (στοιχείων, αριθμών): array
- Matrix: πίνακας

Εικόνες σεMatlab

- Gray-scale Εικόνες: Είναι πίνακες των οποίων οι τιμές αντικατοπτρίζουν τις αποχρώσεις του γκρίζου. Όταν τα στοιχεία μιας gray-scale εικόνας είναι class uint8 or uint16, έχουν ακέραιες τιμές στα διαστήματα [0, 255] ή [0, 65535], αντίστοιχα.
- Αν οι τιμές είναι class double ή Single, οι τιμές είναι αριθμοί στο σύστημα κινητής υποδιαστολής .
- Οι τιμές Values σε double και single gray-scale εικόνες συνήθως ανακλιμακώνονται στο εύρος τιμών [0,1]

Εικόνες σεMatlab

• Binary images: Οι δυαδικές εικόνες MATLAB είναι *logical* πίνακες τιμών 0 και 1. Αν ο πίνακας A έχει τιμές μόνο 0 και 1 αλλά είναι data class π.χ. uint8 θα πρέπει να γίνει logical image στη MATLAB και για να γίνει γράφουμε:

B = logical(A)

Αν ο A έχει και άλλες τιμές εκτός από 0,1 τότε η συνάρτηση logical κάνει:

1) Όλες τις μη μηδενικές τιμές σε logical 1s και

2) ‘Ολες τις μηδενικές σε logical 0s.

Με τη συνάρτηση :

islogical(C)

Αν ο C είναι logical array, επιστρέφει 1, αλλιώς 0.

Επίσης έχουμε **Indexed images** και **RGB** images που θα ασχοληθούμε αργότερα.

Άλλοι χρήσιμοι τελεστές

Operator	Name
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Operator	Description
&	Elementwise AND
|	Elementwise OR
~	Elementwise and scalar NOT
&&	Scalar AND
||	Scalar OR

& (AND operator) και | (operator OR) μπορεί να λειτουργήσει σε πίνακες στοιχείο προς στοιχείο.
&& και || είναι εκδόσεις «βραχυκυκλώματος» για τις οποίες το δεύτερο σκέλος αξιολογείται μόνο όταν το αποτέλεσμα δεν καθορίζεται πλήρως από το πρώτο. Αυτά μπορούν να λειτουργούν μόνο σε π.χ. σε έναν integer ή double , όχι σε πίνακες.
A & B (A and B are evaluated)
A && B (B is only evaluated if A is true)

Προγραμματισμός - Εντολές

Statement	Description
`if`	`if`, together with `else` and `elseif`, executes a group of statements based on a specified logical condition.
`for`	Executes a group of statements a fixed (specified) number of times.
`while`	Executes a group of statements an indefinite number of times, based on a specified logical condition.
`break`	Terminates execution of a `for` or `while` loop.
`continue`	Passes control to the next iteration of a `for` or `while` loop, skipping any remaining statements in the body of the loop.
`switch`	`switch`, together with `case` and `otherwise`, executes different groups of statements, depending on a specified value or string.
`return`	Causes execution to return to the invoking function.
`try...catch`	Changes flow control if an error is detected during execution.

Δουλεύοντας με πίνακες-εικόνες

```
%Δημιουργούμε έναν πίνακα 100x100 με τυχαίες
τιμές
A=rand(100,100);
figure, imshow(A)
min(min(A))
max(max(A))
imwrite(A,'2.gif'); Z=imread('2.gif');
B=uint8(A);
figure, imshow(B)
figure, imshow(A)
% Τι παρατηρείτε?
```

imshow will display a matrix of type double as a greyscale image as long as the matrix elements are between 0 and 1.

help imdemos

Αλλαγή εύρους τιμών

- Υπάρχει η γενική σχέση

$$\frac{max_{new} - min_{new}}{max_{old} - min_{old}} \cdot (v - min_{old}) + min_{new}$$

- Θέλουμε π.χ. να πάμε από 0-1 σε 0-255

Αλλαγή εύρους τιμών

%Create a random image 100x100 and rescale it 0-255 (assign it to B). Then write it as a gif file and read again in the variable B2. Are the values the sam=

A=rand(100,100);

B=(A-min(min(A)))*255/(max(max(A))-min(min(A)));

$$\frac{max_{new} - min_{new}}{max_{old} - min_{old}} \cdot (A - min_{old}) + min_{ne}$$

max(max(B))

B=uint8(B);

figure, imshow(B)

figure, imshow(A)

imwrite(B,'B.gif')

B2=imread('B.gif');

sum(sum(B-B2))

Άσκηση: Φτιάξτε έναν πίνακα C με for loops που να κάνει rescale τις τιμές του A από 0-255 και επιβεβαιώστε ότι είναι ο ίδιος με τον πίνακα B του προηγούμενου παραδείγματος

```
for i=1:100
        for j=1:100
        C(i,j)=(A(i,j)-min(min(A)))*255/(max(max(A))-min(min(A)));
        end
end
sum(sum(B-C))
```

Αλλάζοντας data class

- Η λειτουργία im2uint8 ορίζει σε 0 όλες τις τιμές εισόδου που είναι μικρότερες από 0, ορίζει σε 255 όλες τις τιμές στην είσοδο που είναι μεγαλύτερες από 1 και πολλαπλασιάζει όλες τις άλλες τιμές κατά 255. Η στρογγυλοποίηση των αποτελεσμάτων του πολλαπλασιασμού στον πλησιέστερο ακέραιο ολοκληρώνει μετατροπή.
- Η λειτουργία im2double μετατρέπει μια είσοδο στην κλάση double. Εάν η είσοδος είναι κλάσης uint8, uint16 ή logical, η λειτουργία im2double τη μετατρέπει σε κλάση double με τιμές στην περιοχή [0, 1]. Αν η είσοδος είναι μονής τάξης ή είναι ήδη double , το im2double επιστρέφει έναν πίνακα που είναι double , αλλά είναι αριθμητικά ίσος με την είσοδο.

```
h = uint8([25 50; 128 200]);
c=double(h)/255
g = im2double(h)
```

- η μετατροπή όταν η είσοδος είναι κλάσης uint8 γίνεται απλά διαιρώντας κάθε τιμή με 255. Εάν η είσοδος είναι της κλάσης uint16 η διαίρεση είναι με 65535.

ΑΣΚΗΣΗ: Αλλάξτε το data class του A, δημιουργώντας έναν νέο πίνακα B3 με την εντολή im2uint8 και συγκρινετέ τον με τους πίνακες B καιC. Αν δεν είναι ίδιοι τι έχει συμβεί?

```
B3=im2uint8(A);


sum(sum(B-B3))
 sum(sum(uint8(C)-B3))
sum(sum(uint8(A*255)-B3))
```

Μετατροπή Κλάσεων-Πίνακες logical και binary

- Ή παρακάτω εντολή
- g = f > T
- Δημιουργεί ένα πίνακα logical που περιέχει 1 όπου τα στοιχεία του f είναι μεγαλύτερα από T (κατώφλι) και 0 σε κάθε άλλη περίπτωση.
- Η εντολή του Toolbox im2bw δίνεται με την παρακάτω σύνταξη :
- g = im2bw(f, T)
- **Οι τιμές που καθορίζονται για το όριο T πρέπει να είναι στην περιοχή [0, 1], ανεξάρτητα από την κλάση της εικόνας εισόδου.**
- **Η λειτουργία κλιμακώνει αυτόματα την τιμή κατωφλίου σύμφωνα με την κατηγορία εικόνας εισόδου. Για παράδειγμα, αν το f είναι uint8 και το T είναι 0.4, τότε το im2bw ορίζει τα εικονοστοιχεία σε f, συγκρίνοντάς τα με την τιμή 255 * 0.4 = 102.**

Δουλεύοντας με πίνακες-εικόνες

```
A=rand(100,100);

B=(A-min(min(A)))*255/(max(max(A))-min(min(A)));

B=uint8(B);
```

G=B>102;

G2=im2bw(B,102) Λάθος! Ποια είναι η σωστή σύνταξη ώστε G2=G?

```
G2=im2bw(B,0.4);
sum(sum(G-G2))
G2N=im2double(G2);
```

Βασικοί, έτοιμοι Standard Πίνακες του Matlab

Οποιαδήποτε από τις ακόλουθες λειτουργίες-> το αποτέλεσμα είναι ένας τετράγωνος πίνακας

- zeros (M, N) generates an M x N matrix of 0s of class double.
- ones (M, N) generates an M x N matrix of 0s of class double.
- true (M, N) generates an M x N logical matrix of 1s.
- false (M, N) generates an M x N logical matrix of 0s.
- **magic (M) generates an M x M "magic square." This is a square array in which the sum along any row, column, or main diagonal, is the same. Magic squares are useful arrays for testing purposes because they are easy to generate and their numbers are integers.**
- eye (M) generates an M x M identity matrix.
- rand (M, N) generates an M x N matrix whose entries are uniformly distributed random numbers in the interval [0 , 1].
- randn (M, N) generates an M x N matrix whose numbers are normally distributed (i.e., Gaussian) random numbers with mean 0 and variance 1.

Άσκηση

- Δημιουργείστε έναν ‘μαγικό πίνακα’ 1000x1000
- Βρείτε το max και το min του πίνακα αυτού
- Αποδείξτε με απλές εντολές matlab ότι το άθροισμα των γραμμών του είναι το ίδιο με το άθροισμα των στηλών του.
- Αποδείξτε ότι το άθροισμα των στοιχείων της κυρίας διαγωνίου είναι ίδιο με το άθροισμα οποιασδήποτε γραμμής ή στήλης.
- Απεικονίστε τον πίνακα ως εικόνα αφού γίνει η κατάλληλη μετατροπή τιμών και χρησιμοποιώντας όσα μάθαμε στο 2ο εργαστήριο ώστε να σωθεί σε εικόνα gif. Ξαναδιαβάστε το gif αρχείο και βεβαιωθείτε ότι οι τιμές είναι οι ίδιες.

```
A=magic(1000);
sum(sum(A)-sum(A'))
max(max(A))
min(min(A))
figure, imshow(mat2gray(A))
figure, imagesc(mat2gray(A))
B=uint8(mat2gray(A))*255;
total=0
for i=1:1000
        for j=1:1000
                if i==j
                total=total+A(i,j);
                end
        end
end
sum(A(1,:))-sum(sum(total))
```

Boundary Detection

Γείτονες και περίγραμμα εικόνας

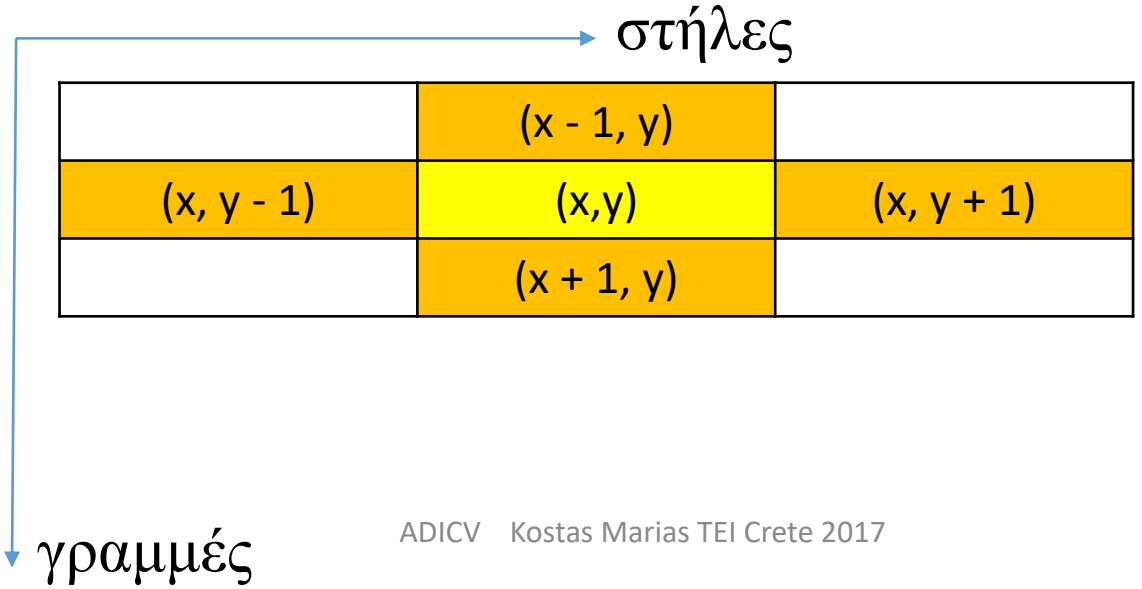
- Ορίζουμε ως V το σύνολο των τιμών εντάσεων εικόνας για να ορίσουμε γειτνίαση.
- Στην δυαδική εικόνα (binary) V = {1}

Basic

Βασικές σχέσεις ανάμεσα σε pixels

- Γείτονες του pixel p: 4-γείτονες $N_4(p)$

- Είναι το σύνολο από τους Τέσσερεις οριζόντιους και κάθετους: (x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)

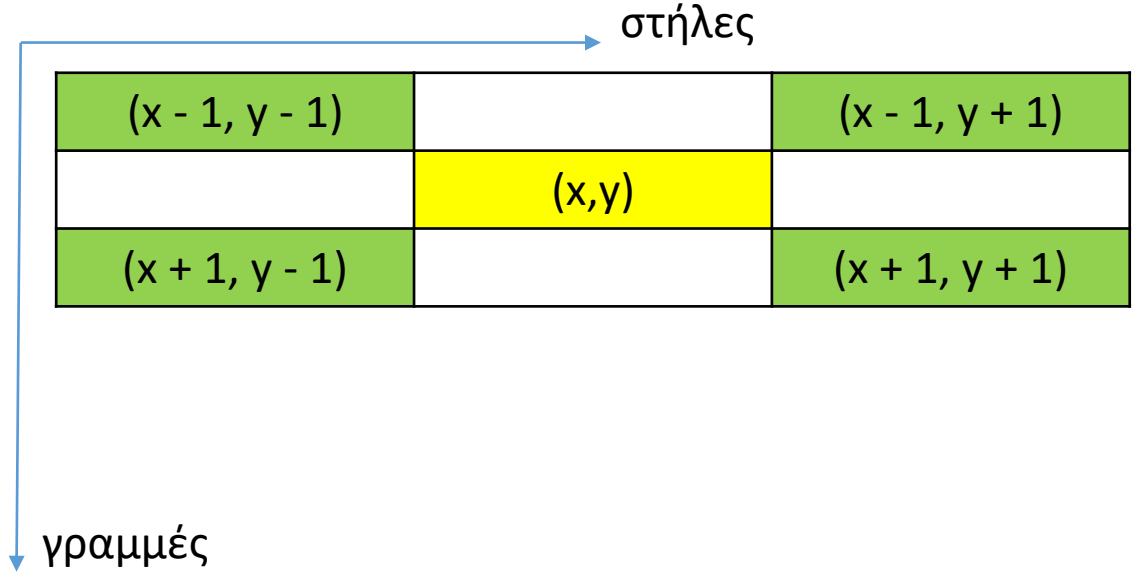


Basic

Βασικές σχέσεις ανάμεσα σε pixels

- Γείτονες του pixel p: 4- διαγώνιοι γείτονες $N_D(p)$

- Είναι το σύνολο από τους Τέσσερεις διαγώνιους:
  (x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)



Βασικές σχέσεις ανάμεσα σε pixels

- Γείτονες του pixel p: 8-γείτονες $N_8(p)=N_4(p)+ N_D(p)$

στήλες

(x - 1, y - 1)	(x - 1, y)	(x - 1, y + 1)
(x, y - 1)	(x,y)	(x, y + 1)
(x + 1, y - 1)	(x + 1, y)	(x + 1, y + 1)

γραμμές

- Σε όλες τις περιπτώσεις αν το (x,y) είναι στο περίγραμμα της εικόνας οι γείτονες ενδέχεται να είναι έξω από την εικόνα!!!

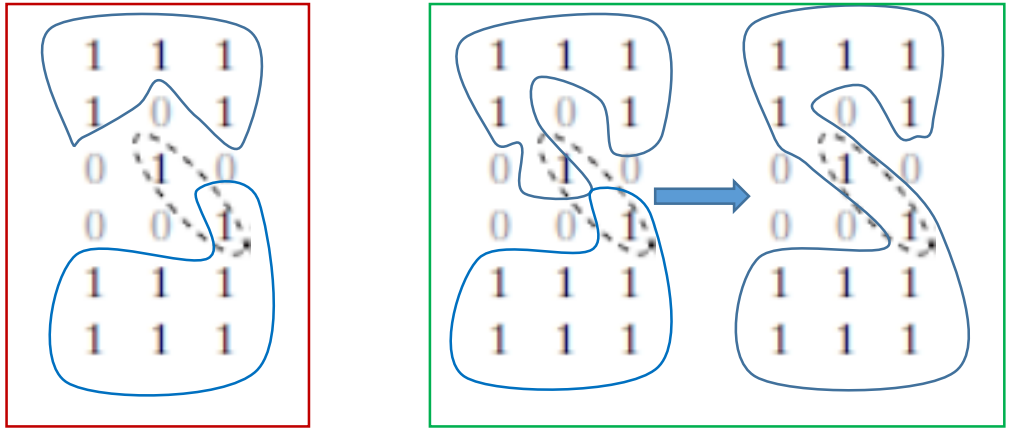
Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

Basic

- Έστω ότι *R είναι ένα υποσύνολο των* pixels μιας εικόνας.
- Το *R είναι μια περιοχή (region) της εικόνας αν είναι ένα συνδεδεμένο σύνολο* (connected set) δηλ. όλα τα pixels του είναι συνδεδεμένα (*υπάρχει ένα μονοπάτι γειτνίασης αναμεσά τους με pixels που ανήκουν αποκλειστικά στο R)*
- Δύο περιοχές είναι γειτονικές αν η ένωσή τους σχηματίζει *ένα συνδεδεμένο σύνολο*
- *Διαφορετικά οι περιοχές είναι ξεχωριστές (disjoint)*.
- Για να ορίσουμε όλα τα παραπάνω πρέπει να έχουμε ορίσει πρώτα αν μιλάμε για 4- ή -8, m γειτνίαση περιοχών.

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

- Στο παρακάτω παράδειγμα αριστερά οι δύο περιοχές που έχουν 4-γειτνίαση δεν γειτονεύουν. Αν χρησιμοποιήσουμε όμως 8-γειτνίαση (δεξιά) οι περιοχές γειτονεύουν γιατί η ένωσή τους είναι ένα συνδεδεμένο σύνολο!



Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

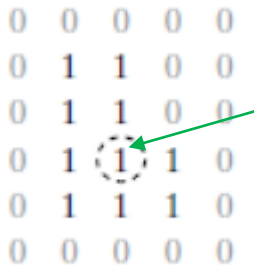
- Έστω ότι μια εικόνα περιέχει *K ξεχωριστές περιοχές* R_k, *k=1..K.*
- *Η ένωσή τους* R_u *ορίζεται ως προσκήνιο-foreground*.
- Η συμπληρωματική περιοχή {όλα όσα ΔΕΝ είναι στο R_u *δηλ. ανήκουν στο* $(R_u)^c$} είναι το φόντο-*background* της εικόνας.
- Το σύνορο ή περίγραμμα (*boundary, border* ή *contour*) μιας περιοχής *R είναι το σύνολο των σημείων της περιοχής που γειτονεύουν με το συμπλήρωμα της περιοχής R*.
- Σύνορο R = σύνολο pixels R που έχουν τουλάχιστον 1 γείτονα από το background.

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

- Έστω ότι μια εικόνα περιέχει *K ξεχωριστές περιοχές* R_k, *k=1..K.*
- *Η ένωσή τους* R_u *ορίζεται ως προσκήνιο-foreground*.
- Η συμπληρωματική περιοχή {όλα όσα ΔΕΝ είναι στο R_u *δηλ. ανήκουν στο* $(R_u)^c$} είναι το φόντο-*background* της εικόνας.
- Το σύνορο ή περίγραμμα (*boundary, border* ή *contour*) μιας περιοχής *R είναι το σύνολο των σημείων της περιοχής που γειτονεύουν με το συμπλήρωμα της περιοχής R*.
- Σύνορο R = σύνολο pixels R που έχουν τουλάχιστον 1 γείτονα από το background.

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

- Σχετικά με το περίγραμμα πρέπει και πάλι να ορίσουμε αν αναφερόμαστε σε 4-,8-, m- γειτνίαση.
- Για να αποφύγουμε ασάφεια (βλέπε παρακάτω σχήμα) χρησιμοποιούμε 8-γειτνίαση/συνδεσιμότητα για σημεία μιας περιοχής και του background.



Το pixel αυτό ΔΕΝ ανήκει στο περίγραμμα της περιοχής με τιμές ‘1’ αν χρησιμοποιήσουμε 4-γειτνίαση με το φόντο!

Γειτνίαση, Συνδεσιμότητα, Περιοχές , και Όρια

- Ο προηγούμενος ορισμός είναι το εσωτερικό περίγραμμα σε αντίθεση με το εξωτερικό που είναι το αντίστοιχο του background.
- Αυτό είναι σημαντικό για ανάπτυξη αλγορίθμων που εντοπίζουν το περίγραμμα της εικόνας γιατί το περίγραμμα είναι πάντα ένα κλειστό μονοπάτι!

0	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	0

- ✓ Το εσωτερικό περίγραμμα της περιοχής εικόνας με τιμές – 1 είναι η ίδια περιοχή ΑΛΛΑ δεν είναι κλειστό μονοπάτι!!!
- ✓ Το εξωτερικό όμως είναι!!!

Δημιουργία εικόνας και εύρεση περιγράμματος

```
A=zeros(100);
A(30:50,30:50)=1;
B=zeros(100); %Πως θα βρούμε το περίγραμμα? Ιδέες?
for i=1:100
for j=1:100
if A(i,j)==1 && (A(i,j-1)==0 | A(i,j+1)==0  |A(i-1,j)==0|A(i+1,j)==0)
B(i,j)=A(i,j);
end
end
end
```

	(i - 1, j)	
(i, j - 1)	(i,j)	(i, j + 1)
	(i + 1, j)	

Δημιουργία εικόνας και εύρεση περιγράμματος

```
%Ας φτιάξουμε ένα δεύτερο σχήμα που να ακουμπάει με το πρώτο.
A(51:70,51:70)=1;
figure, imshow(A)
B=zeros(100);
for i=1:100
for j=1:100
if A(i,j)==1 & (A(i,j-1)==0 | A(i,j+1)==0  |A(i-1,j)==0|A(i+1,j)==0)
B(i,j)=A(i,j);
end
end
end
```

- Φτιάχτε μια καινούρια εικόνα που να έχει 255 GreyLevel τιμή στα δύο τετράγωνα και 155 στο περίγραμμα.
- C=A*255-B*100;

figure, imshow(uint8(C))

Δημιουργία εικόνας και εύρεση περιγράμματος στο γράμμα ‘$\mathscr{B}$’

```
A=imread('B.gif');
figure, imshow(A)
[rows,columns]=size(A);
B=zeros(rows,columns);
for i=1:rows
for j=1:columns
if A(i,j)==255 & (A(i,j-1)==0 | A(i,j+1)==0  |A(i-1,j)==0|A(i+1,j)==0)
B(i,j)=A(i,j);
end
end
end
figure, imshow(B)
```

Δημιουργία εικόνας και εύρεση περιγράμματος με 8-γείτονες

```
B2=zeros(rows,columns);
for i=1:rows
for j=1:columns
if A(i,j)==255 & (A(i,j-1)==0 |A(i,j+1)==0 |A(i-1,j)==0|A(i+1,j)==0 |A(i-1,j-1)==0|A(i+1,j-1)==0|A(i-1,j+1)==0|A(i+1,j+1)==0)
B2(i,j)=A(i,j);
end
end
end
figure, imshow(B2)
%κατευθείαν μπορούμε να βρούμε τους διαγώνιους γείτονες
```

Δημιουργία εικόνας και εύρεση περιγράμματος με 8-γείτονες

```
%κατευθείαν μπορούμε να βρούμε τους διαγώνιους γείτονες
C=B2-B;
figure, imshow(uint8(C))

%Πάνω στην εικόνα Α αναδείξτε το περίγραμμά της
C2=A-uint8(B/2);
 figure, imshow(C2)
```

Boundary detection of ‘trees.bmp’

- Να βρεθεί το περίγραμμα της εικόνας trees.bmp κρατώντας μόνο τις τιμές του γκρίζου που είναι μεγαλύτερες του 70.

```
[X,map] = imread('trees.bmp');
figure, imshow(X);
figure, imshow(X>70);
 A=uint8((X>70)*255);
figure, imshow(A);
```

Boundary detection of ‘trees.bmp’

```
[rows,columns]=size(A);
B=zeros(rows,columns);
for i=2:rows-1
for j=2:columns-1
if A(i,j)==255 & (A(i,j-1)==0 | A(i,j+1)==0  |A(i-1,j)==0|A(i+1,j)==0)
B(i,j)=A(i,j);
end
end
end
figure, imshow(B)
```

Create a myboundary function

```
function [Bo Bi] = myboundary(A)
[m n]=size(A);
Bo=zeros(m,n);
Bi=zeros(m,n);
for i=2:m-1
  for j=2:n-1
    if A(i,j)==0 && (A(i,j-1)==1 | A(i,j+1)==1  |A(i-1,j)==1|A(i+1,j)==1)
    Bo(i,j)=1;
    end
    if A(i,j)==1 && (A(i,j-1)==0 | A(i,j+1)==0  |A(i-1,j)==0|A(i+1,j)==0)
    Bi(i,j)=1;
    end
  end
end
figure, imshow(Bo,[]), title('outer boundary of the image')
figure, imshow(Bi,[]), title('internal boundary of the image')
endfunction
```