

Επεξεργασία Εικόνας & Βίντεο



02. Εισαγωγή στην Εικόνα

Εισηγητής: Νικόλαος Γιαννακέας

Επίκουρος Καθηγητής, Σημάτων & Συστημάτων

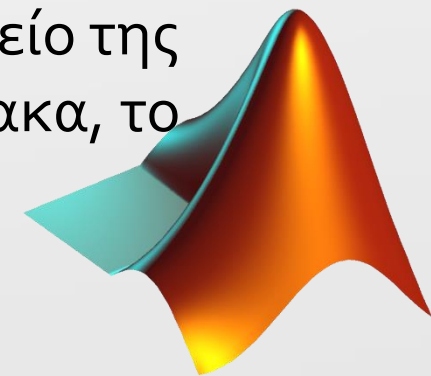




Εισαγωγή

Η βασικές εντολές στο MATLAB για την εκτέλεση πράξεων μεταξύ εικόνων, αλλά και για την διαχείριση των εικονοστοιχείων (pixels) μιας εικόνας είναι οι εντολές που άπτονται στην διαχείριση πινάκων που μελετήθηκαν στο προηγούμενο εργαστήριο. Όπως εντόνως αναδείχθηκε από το θεωρητικό μάθημα, **οι ψηφιοποιημένες εικόνες δεν είναι τίποτε άλλο παρά μεγάλοι πίνακες τιμών**. Στην αναπαράσταση μια εικόνας με πίνακα τιμών, η θέση ενός εικονοστοιχείου αντιστοιχεί στην θέση του στοιχείου του πίνακα (γραμμή, στήλη).

Για παράδειγμα σε εικόνα 512x512 το πάνω αριστερά εικονοστοιχείο της είναι το στοιχείο της πρώτης γραμμής και πρώτης στήλης του πίνακα, το $A(1,1)$.

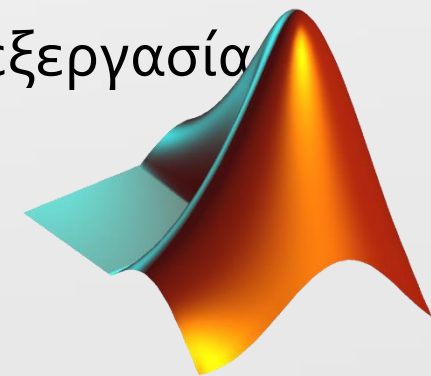


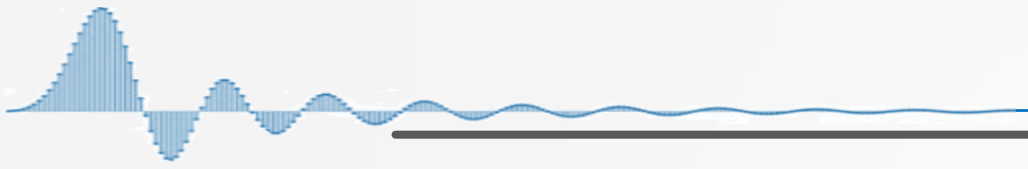


Εισαγωγή

Το κάτω αριστερά εικονοστοιχείο είναι το στοιχείο του πίνακα στην 512η γραμμή και 1η στήλη, το $A(512,1)$. Η **τιμή** του **στοιχείου** του πίνακα **αντιστοιχεί** στην **φωτεινότητα** του **εικονοστοιχείου** που βρίσκεται στην αντίστοιχη θέση της εικόνας.

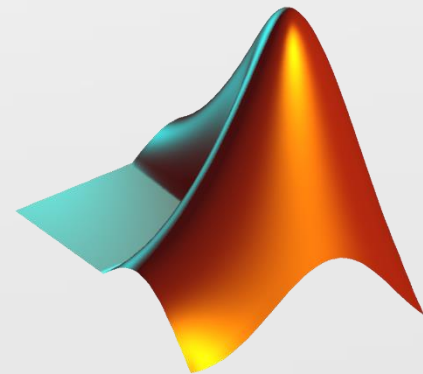
Πέραν του τρόπου διαχείρισης των εικονοστοιχείων των εικόνων ως στοιχεία πινάκων το MATLAB παρέχει μια σειρά **βιβλιοθηκών** (υλοποιημένων συναρτήσεων) εξειδικευμένων για επεξεργασία εικόνων. Οι συναρτήσεις αυτές συμπεριλαμβάνονται στην «εργαλειοθήκη επεξεργασία εικόνας» (Image Processing Toolbox).

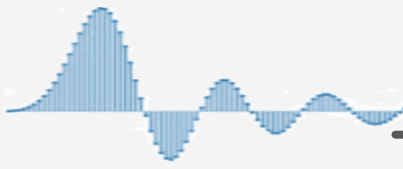




Εισαγωγή

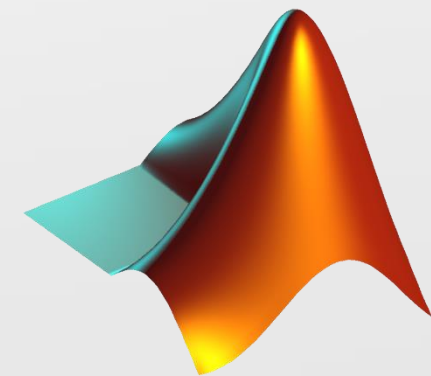
Η εργαλειοθήκη περιλαμβάνει από τις βασικές εντολές για **εισαγωγή, εξαγωγή και οπτικοποίηση των εικόνων, έως εξελιγμένες μεθόδους μετασχηματισμού, βελτίωσης, επεξεργασίας και ανάλυσης**. Παρακάτω παραθέτουμε ένα σύνολο σημαντικών εντολών από την εργαλειοθήκη επεξεργασίας εικόνων. Περισσότερες λεπτομέρειες μπορεί κάποιος να αναζητήσει είτε στην πλατφόρμα βοήθειας του MATLAB είτε στον ιστότοπο της Mathworks. (<http://www.mathworks.com/help/images/index.html>)

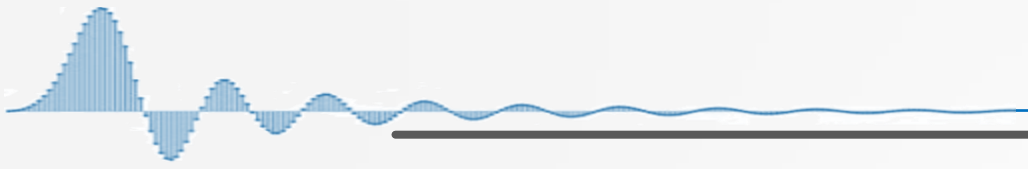




Εντολές Εισαγωγής-Εξαγωγής

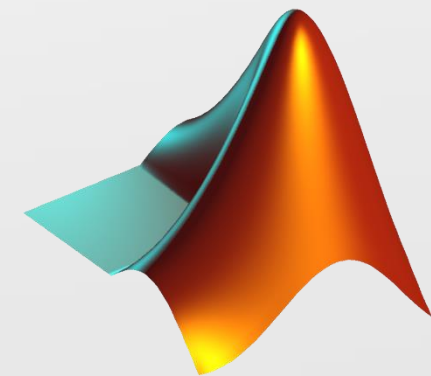
Εντολή	Περιγραφή
imread	Διαβάσει εικόνα από αρχείο και την εισάγει στο workspace
imwrite	Γράφει την εικόνα από το workspace σε αρχείο εικόνας (π.χ. .jpg)
imfinfo	Παρέχει πληροφορίες για ένα αρχείο εικόνας

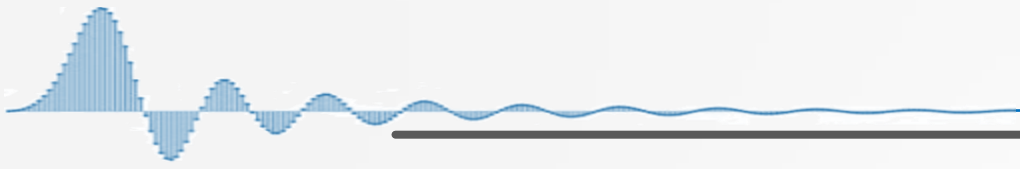




Εντολές Οπτικοποίησης

Εντολή	Περιγραφή
<code>imshow</code>	Παρουσίαση της εικόνας
<code>subimage</code>	Παρουσίαση πολλαπλών εικόνων σε ένα σχήμα

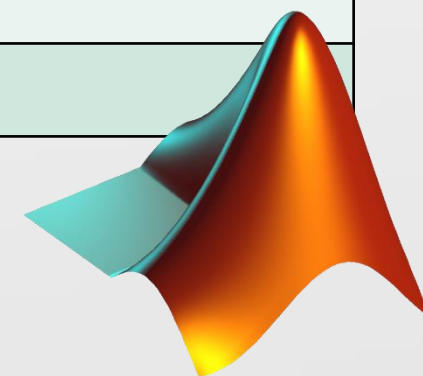


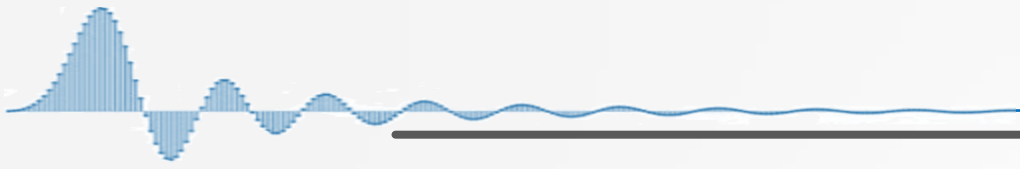


Εντολές Μετατροπής

Μετατροπές Επιπέδων Γκρι – Έγχρωμων – Δεικτών

Εντολή	Περιγραφή
<code>gray2ind</code>	Μετατροπή επιπέδων γκρι ή δυαδικής σε εικόνες δεικτών
<code>ind2gray</code>	Μετατροπή εικόνας δεικτών σε εικόνα επιπέδων γκρι
<code>mat2gray</code>	Μετατροπή πίνακα σε εικόνα επιπέδων γκρι
<code>rgb2gray</code>	Μετατροπή RGB σε επιπέδων γκρι
<code>ind2rgb</code>	Μετατροπή εικόνας δεικτών σε RGB

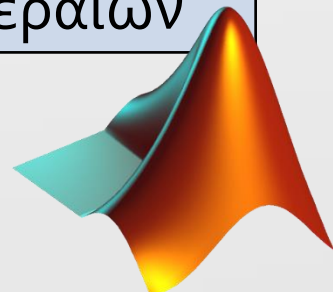


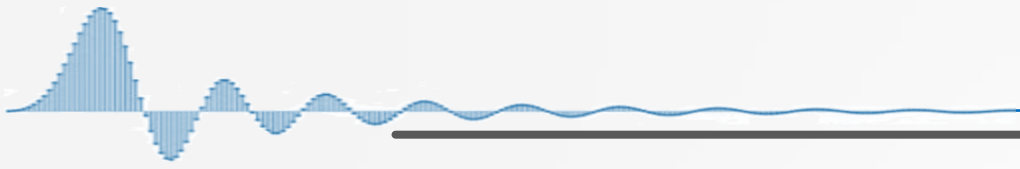


Εντολές Μετατροπής

Μετατροπές ψηφίων ανα εικονοστοιχείο

Εντολή	Περιγραφή
<code>im2double</code>	Μετατροπή σε εικόνα διπλής ακρίβειας
<code>im2int16</code>	Μετατροπή σε εικόνα 16-bit προσημασμένων ακεραίων
<code>im2single</code>	Μετατροπή σε εικόνα απλής ακρίβειας
<code>im2uint16</code>	Μετατροπή σε εικόνα 16-bit μη προσημασμένων ακεραίων
<code>im2uint8</code>	Μετατροπή σε εικόνα 8-bit μη προσημασμένων ακεραίων

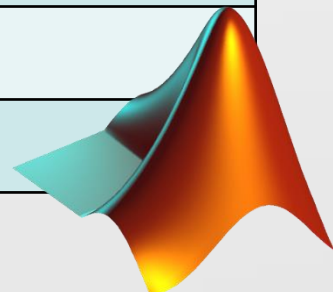


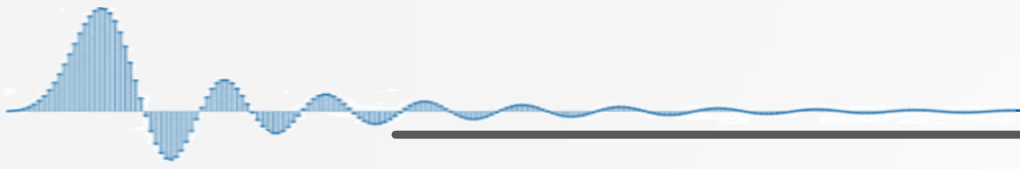


Εντολές Μετατροπής

Μετατροπές χρωματισμών

Εντολή	Περιγραφή
rgb2ntsc	Μετατροπή RGB σε NTSC χρωματισμό
rgb2xyz	Μετατροπή RGB σε CIE 1931 XYZ
rgb2ycbcr	Μετατροπή RGB σε YCbCr χρωματισμό
xyz2rgb	Μετατροπή CIE 1931 XYZ σε RGB
ycbcr2rgb	Μετατροπή YCbCr σε RGB χρωματισμό
rgb2hsv	Μετατροπή RGB σε HSV χρωματισμό
ntsc2rgb	Μετατροπή NTSC σε RGB χρωματισμό

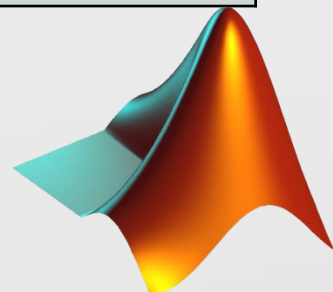


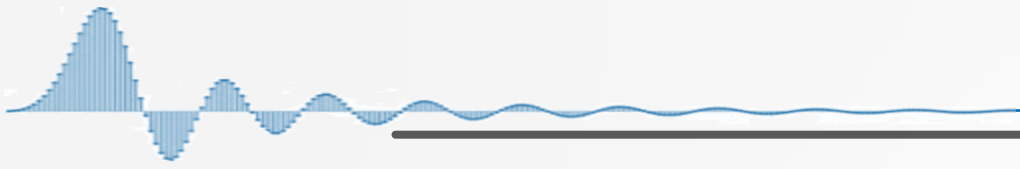


Εντολές Μετατροπής

Εντολές Μετασχηματισμών

Εντολή	Περιγραφή
bwdist	Μετασχηματισμός απόστασης δυαδικής εικόνας
graydist	Gray-weighted distance transform of grayscale image
hough	Μετασχηματισμός Hough
dct2	2-D Διακριτός Μετασχηματισμός Συνημιτόνου
dctmtx	Πίνακας Διακριτού Μετασχηματισμού Συνημιτόνου

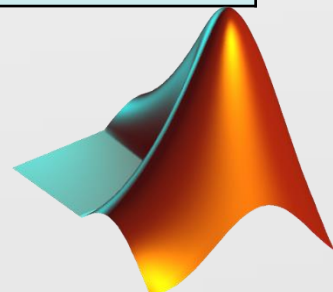


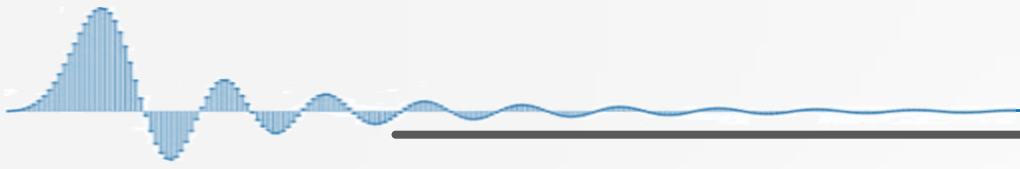


Εντολές Μετατροπής

Εντολές Μετασχηματισμών

Εντολή	Περιγραφή
<code>idct2</code>	2-D Αντίστροφος Διακριτός Μετασχηματισμός Συνημιτόνου
<code>iradon</code>	Αντίστροφος Μετασχηματισμός Radon
<code>radon</code>	Μετασχηματισμός Radon
<code>fft2</code>	2-D Γρήγορος Μετασχηματισμός Fourier (FFT)
<code>ifft2</code>	2-D Αντίστροφος Γρήγορος Μετασχηματισμός Fourier

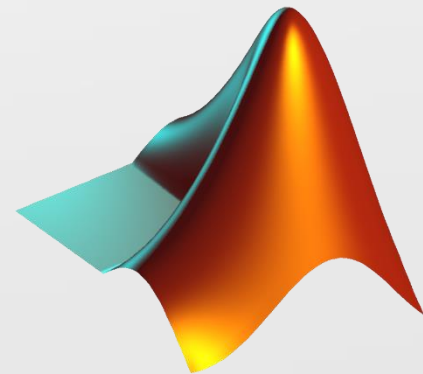


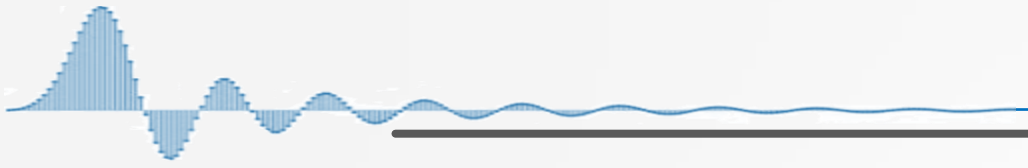


ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 1

Εισάγεται στο workspace του MATLAB την Λένα (grayscale), και λάβετε τον «καθρεφτισμό» της με την βοήθεια της συνάρτησης που υλοποιήσαμε στο προηγούμενο εργαστήριο.



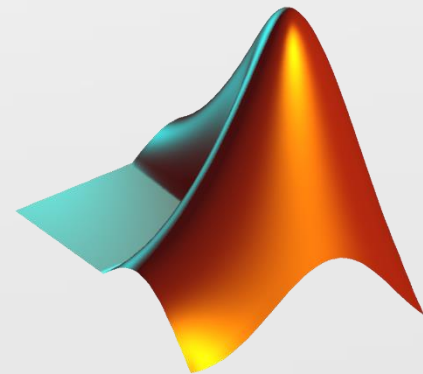


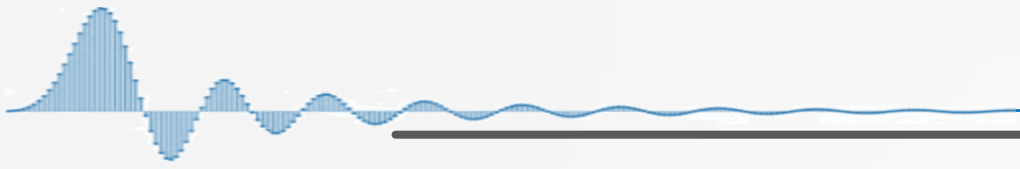
ΛΥΣΗ

Υλοποίηση Συνάρτησης:

```
function mirrored = mirroring(l)

    mirrored = zeros(size(l,1),size(l,2));
    count = 1;
    for i = size(l,2):-1:1
        mirrored(:,count) = l(:,i);
        count = count + 1;
    end
end
```





Κλήση Συνάρτησης:

ΛΥΣΗ

```
I = imread('lena.png');
```



Διάβασμα έγχρωμης εικόνας

```
I_gray = rgb2gray(I);
```



Μετατροπή εικόνας σε γκρι

```
I_mirror=mirroring(I_gray);
```



Κλήση συνάρτησης καθρεπτισμού

```
figure, imshow('lena.png')
```



Εμφάνιση έγχρωμης εικόνας

```
figure, imshow(I_gray)
```



Εμφάνιση γκρι εικόνας

```
figure, imshow(uint8(I_mirror))
```

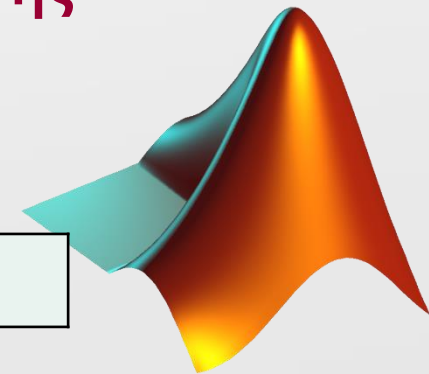


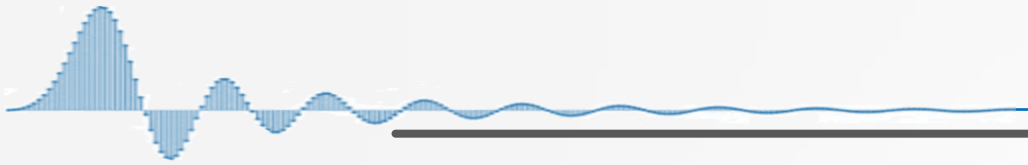
Εμφάνιση καθρεπτισμένης εικόνας

Μετατροπή σε 8bit απρόσημο ακέραιο πίνακα

rgb2gray

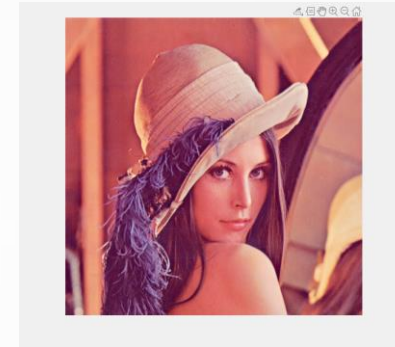
Μετατροπή RGB σε επιπέδων γκρι





Κλήση Συνάρτησης:

```
>> I = imread('lena.png');  
>> I_gray = rgb2gray(I);  
>> I_mirror=mirroring(I_gray);  
>> figure, imshow('lena.png')  
>> figure, imshow(I_gray)  
>> figure, imshow(uint8(I_mirror))
```

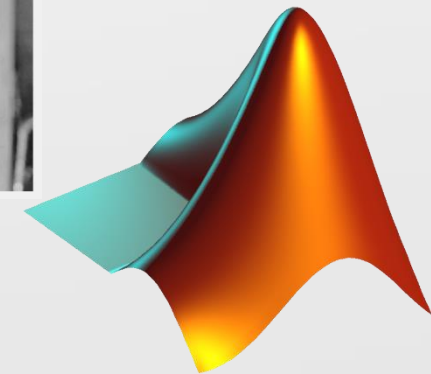
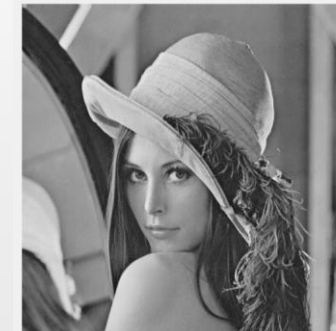


ΛΥΣΗ



Υλοποίηση Συνάρτησης:

```
function mirrored = mirroring(I)  
    mirrored = zeros(size(I,1),size(I,2));  
    count = 1;  
    for i = size(I,2):-1:1  
        mirrored(:,count) = I(:,i);  
        count = count + 1;  
    end  
end
```

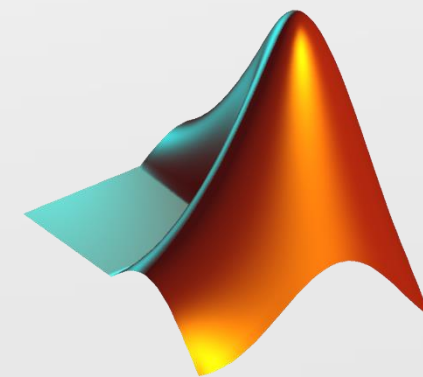


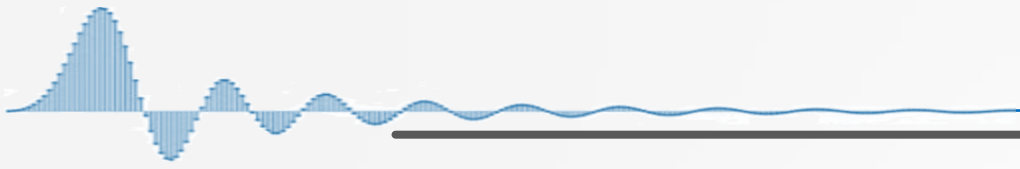


ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 2

Δημιουργήστε μια συνάρτηση η οποία να αντιστρέφει την φωτεινότητα της εικόνας. Η συνάρτηση θα μπορεί να δέχεται οποιαδήποτε εικόνα (έγχρωμη, ασπρόμαυρη). Εάν η εικόνα που εισάγεται είναι ασπρόμαυρη (grayscale) θα εξάγει το αρνητικό το ενός καναλιού, ενώ εάν είναι έγχρωμη θα εξάγει εικόνα με ανεστραμμένα όλα τα κανάλια.

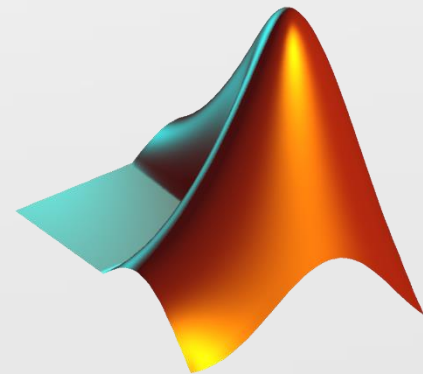


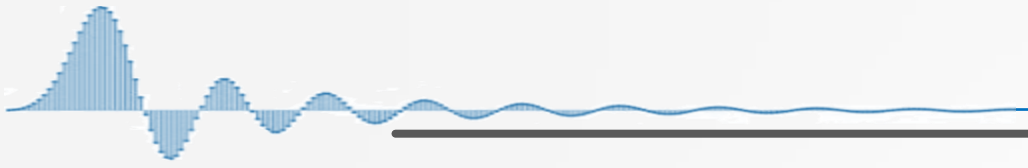


ΛΥΣΗ

```
function arnitiki = arnitiko(I)  
  
end
```

- ✓ Δημιουργία Συνάρτησης:
 - ✓ Η συνάρτηση μας θα επιστρέφει ένα πίνακα arnitiki
 - ✓ Η συνάρτηση μας θα δέχεται ένα πίνακα I

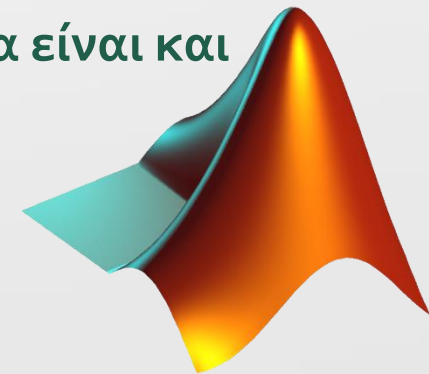


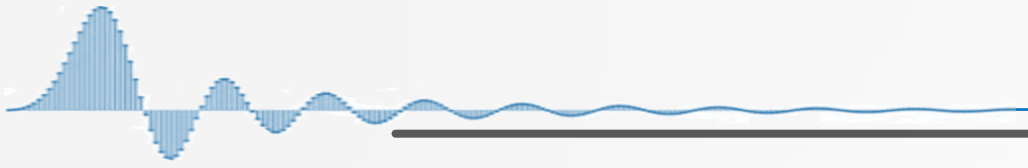


ΛΥΣΗ

```
function arnitiki = arnitiko(l)  
    arnitiki = zeros(size(l,1),size(l,2),size(l,3));  
  
end
```

- ✓ Υλοποίηση Συνάρτησης:
 - ✓ Αρχικοποιούμε ένα πίνακα arnitiki με μέγεθος το μέγεθος του πίνακα που δεχόμαστε ως όρισμα
 - ✓ Σε αυτήν την περίπτωση το όρισμα μπορεί να είναι και τρισδιάστατος πίνακας

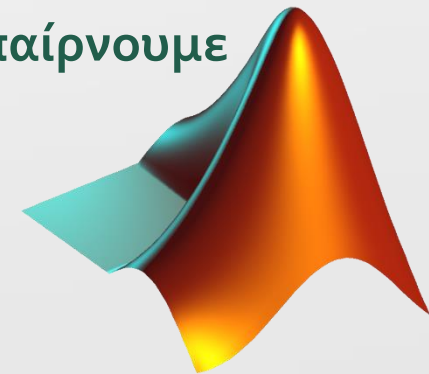


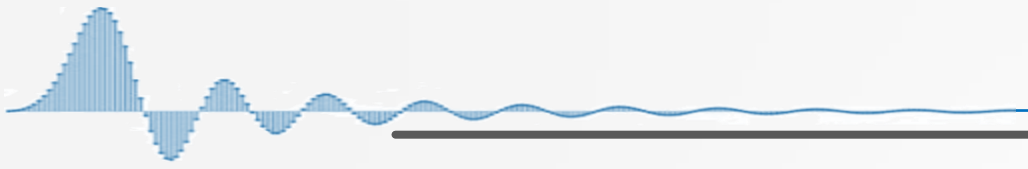


ΛΥΣΗ

```
function arnitiki = arnitiko(I)
    arnitiki = zeros(size(I,1),size(I,2),size(I,3));
    if size(I,3) == 1
end
```

- ✓ Υλοποίηση Συνάρτησης:
 - ✓ Ελέγχουμε αν η εικόνας μας είναι σε κλίμακα του γκρι
 - ✓ Δηλαδή ελέγχουμε αν η εικόνα μας δεν είναι τρισδιάστατη όπου σε αυτή την περίπτωση παίρνουμε σαν αποτέλεσμα την τιμή 1.





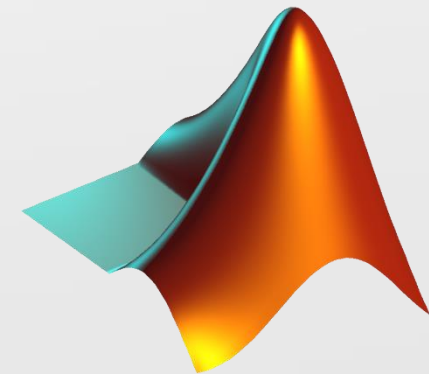
ΛΥΣΗ

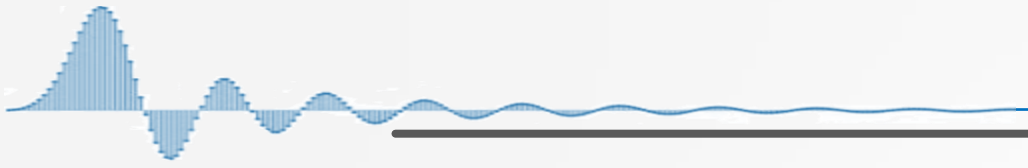
```
function arnitiki = arnitiko(I)
    arnitiki = zeros(size(I,1),size(I,2),size(I,3));
    if size(I,3) == 1
        arnitiki = 255 - I;
    end
```

Εξαγωγή του αρνητικού της εικόνας

- Αφαιρείται η τιμή του κάθε εικονοστοιχείου της εικόνας από την μέγιστη τιμή φωτεινότητας
- Π.χ. 255 αν πρόκειται για 8-bit

- ✓ **Υλοποίηση Συνάρτησης:**
 - ✓ Εξάγουμε το αρνητικό του ενός καναλιού
 - ✓ Δηλαδή αφαιρούμε 255 από το πίνακα I





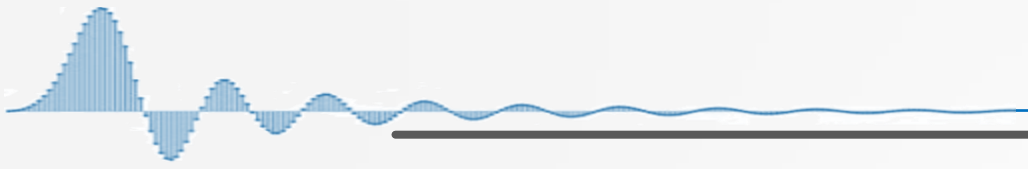
ΛΥΣΗ

```
function arnitiki = arnitiko(I)
    arnitiki = zeros(size(I,1),size(I,2),size(I,3));
    if size(I,3) == 1
        arnitiki = 255 - I;
    elseif size(I,3) == 3
    end
```

✓ Υλοποίηση Συνάρτησης:

- ✓ Ελέγχουμε την περίπτωση όπου έχουμε μια έγχρωμη εικόνα
 - ✓ Σε αυτή την περίπτωση έχουμε τρισδιάστατο πίνακα για αυτό ελέγχουμε αν το $\text{size}(I,3) == 3$





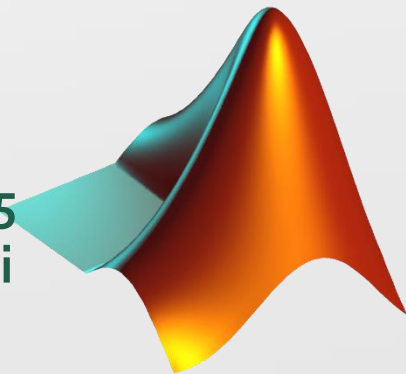
ΛΥΣΗ

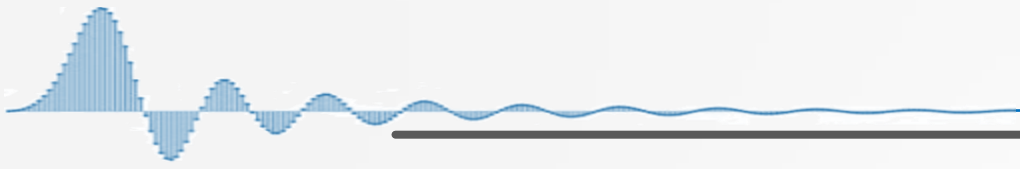
```
function arnitiki = arnitiko(I)
    arnitiki = zeros(size(I,1),size(I,2),size(I,3));
    if size(I,3) == 1
        arnitiki = 255 - I;
    elseif size(I,3) == 3
        arnitiki(:,:,1) = 255 - I(:,:,1);
        arnitiki(:,:,2) = 255 - I(:,:,2);
        arnitiki(:,:,3) = 255 - I(:,:,3);
    end
```

✓ Υλοποίηση Συνάρτησης:

✓ Αναστρέφουμε όλα τα κανάλια

- ✓ Για κάθε μια από τις 3 διαστάσεις παίρνουμε την αντίστοιχη διάσταση από το πίνακα I αφαιρούμε το 255 και το αποθηκεύουμε στο μηδενικό μας πίνακα arnitiki στην αντίστοιχη διάσταση





ΛΥΣΗ

Υλοποίηση Συνάρτησης:

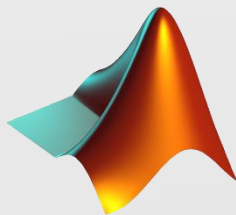
```
function arnitiki = arnitiko(I)
arnitiki = zeros(size(I,1),size(I,2),size(I,3));
if size(I,3) == 1
    arnitiki = 255 - I;
elseif size(I,3) == 3
    arnitiki(:, :, 1) = 255 - I(:, :, 1);
    arnitiki(:, :, 2) = 255 - I(:, :, 2);
    arnitiki(:, :, 3) = 255 - I(:, :, 3);
end
```

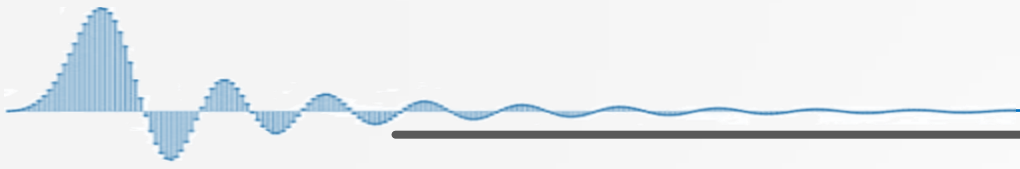
Κλήση Συνάρτησης:

```
I = imread('lena.png');
I_gray = rgb2gray(I);
k = arnitiko(I_gray);
figure, imshow(uint8(k))
```



```
I = imread('lena.png');
k = arnitiko(I);
figure, imshow(uint8(k))
```





ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 3

Υλοποιήστε συνάρτηση στο MATLAB η οποία να δέχεται μια RGB εικόνα και να μετατρέπει τον χρωματισμό της σε σύστημα HSV. Δίνονται οι μαθηματικοί τύποι για την μετατροπή

$$R' = \frac{R}{255}$$

$$G' = \frac{G}{255}$$

$$B' = \frac{B}{255}$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

Hue calculation:

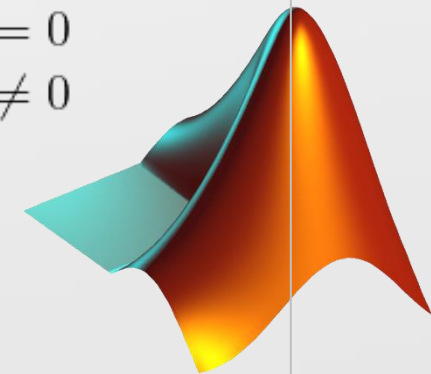
$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6\right) & , C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2\right) & , C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4\right) & , C_{\max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} & , C_{\max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{\max}$$

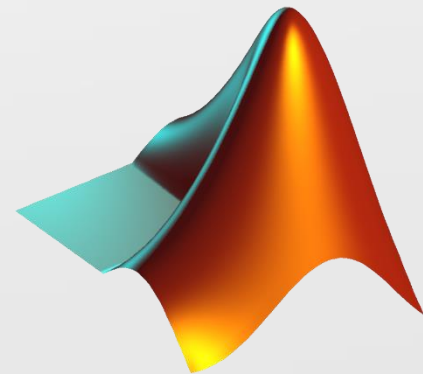


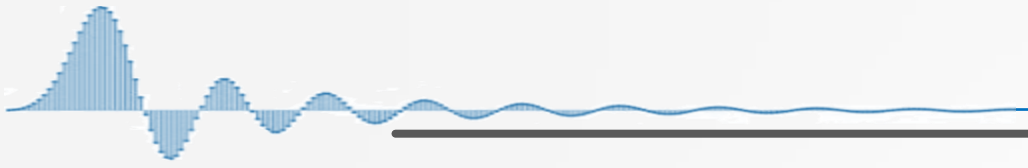


ΛΥΣΗ

```
function I_HSV = HSV(I_RGB)  
  
end
```

- ✓ Δημιουργία Συνάρτησης:
 - ✓ Η συνάρτηση μας θα επιστρέφει ένα πίνακα I_SVH
 - ✓ Η συνάρτηση μας θα δέχεται ένα πίνακα I_RGB

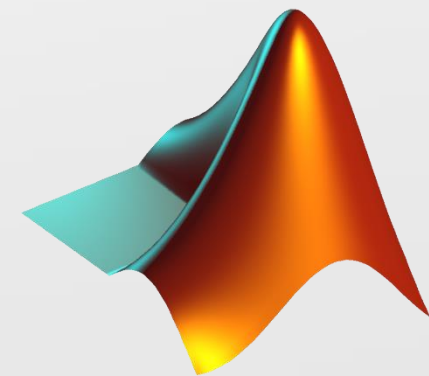


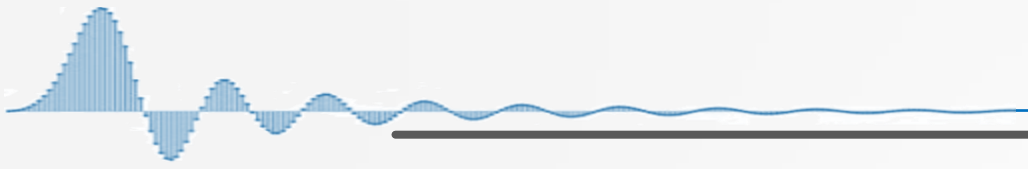


ΛΥΣΗ

```
function I_HSV = HSV(I_RGB)  
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));  
end
```

- ✓ Υλοποίηση Συνάρτησης:
 - ✓ Δημιουργώ ένα μηδενικό πίνακα I_HSV με μέγεθος το μέγεθος του πίνακα που δέχομαι σαν όρισμα





ΛΥΣΗ

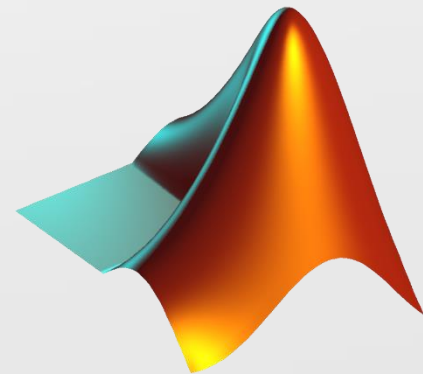
```
function I_HSV = HSV(I_RGB)
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));

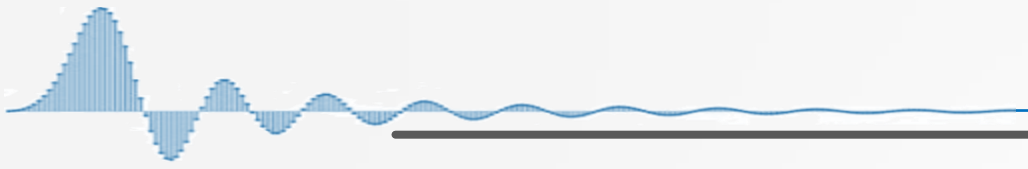
    for i=1:size(I_RGB,1)
        for j=1:size(I_RGB,2)
        end
    end
```

✓ Υπολογισμός HSV:

- ✓ Αρχικοποιούμε το i με τιμή 1
- ✓ Μέχρι το i να γίνει ίσο με το μέγεθος των γραμμών του πίνακα I_RGB

- ✓ Αρχικοποιούμε το j με τιμή 1
- ✓ Μέχρι το j να γίνει ίσο με το μέγεθος των στηλών του πίνακα I_RGB





```
function I_HSV = HSV(I_RGB)
```

```
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
```

```
    for i=1:size(I_RGB,1)
```

```
        for j=1:size(I_RGB,2)
```

```
            rgb = double([I_RGB(i,j,1) I_RGB(i,j,2) I_RGB(i,j,3)]);
```

```
            rgb_tonos = double(rgb./255);
```

ΛΥΣΗ

$$R' = \frac{R}{255}$$

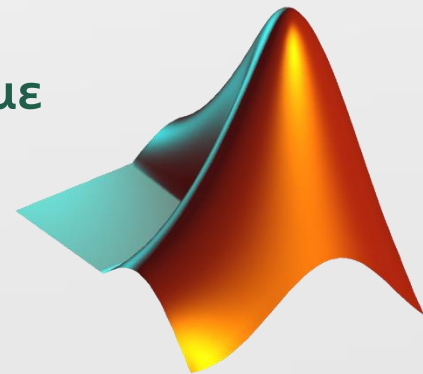
$$G' = \frac{G}{255}$$

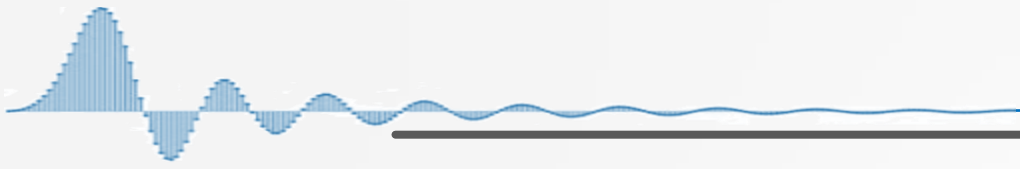
$$B' = \frac{B}{255}$$

✓ Υπολογισμός HSV:

- ✓ Στο rgb έχουμε το πίνακα μας με διπλής ακρίβειας τιμές
 - ✓ Μέσω της συνάρτησης double

- ✓ Για να βρούμε το RGB' θα πρέπει να διαιρέσουμε το RGB με το 255
 - ✓ Με διπλής ακρίβειας αποτέλεσμα





ΛΥΣΗ

```
function I_HSV = HSV(I_RGB)
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));

    for i=1:size(I_RGB,1)
        for j=1:size(I_RGB,2)
            rgb = double([I_RGB(i,j,1) I_RGB(i,j,2) I_RGB(i,j,3)]);
            rgb_tonos = double(rgb./255);
            Cmax = max(rgb_tonos);
            Cmin = min(rgb_tonos);
            Delta = Cmax - Cmin;
```

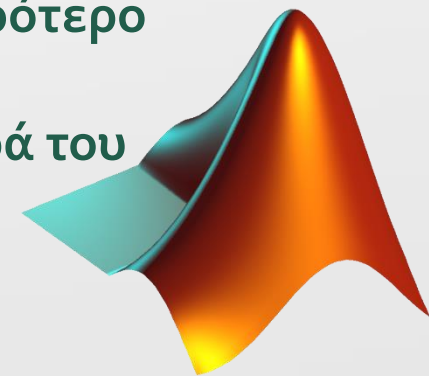
$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

✓ Υπολογισμός HSV:

- ✓ Μέσω της συνάρτησης max βρίσκουμε το μεγαλύτερο RGB
- ✓ Μέσω της συνάρτησης min βρίσκουμε το μικρότερο RGB
- ✓ Στην μεταβλητή Delta εκχωρούμε την διαφορά του $C_{\max} - C_{\min}$



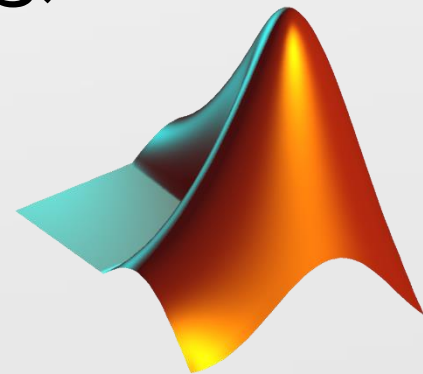


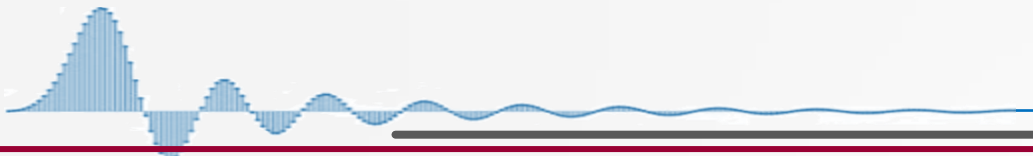
ΛΥΣΗ

$$R' = \frac{R}{255} \longrightarrow \text{rgb_tonos}(1)$$

$$G' = \frac{G}{255} \longrightarrow \text{rgb_tonos}(2)$$

$$B' = \frac{B}{255} \longrightarrow \text{rgb_tonos}(3)$$





```
function I_HSV = HSV(I_RGB)
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
    for i=1:size(I_RGB,1)
        for j=1:size(I_RGB,2)
            ...
            if Cmax == rgb_tonos(1)
                H = 60*(mod((rgb_tonos(2)-rgb_tonos(3))/Delta,6));
            elseif Cmax == rgb_tonos(2)
                H = 60*((rgb_tonos(3)-rgb_tonos(1))/Delta) + 2;
            elseif Cmax == rgb_tonos(3)
                H = 60*((rgb_tonos(1)-rgb_tonos(2))/Delta) + 4;
            elseif Delta == 0
                H = 0;
            ...
        end
    end
end
```

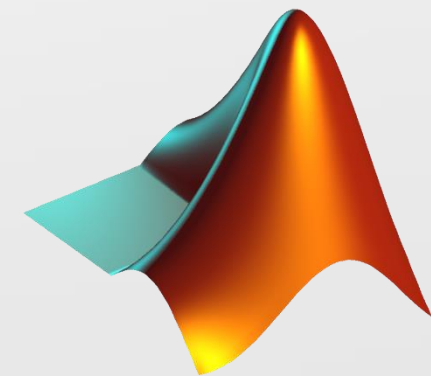
ΛΥΣΗ

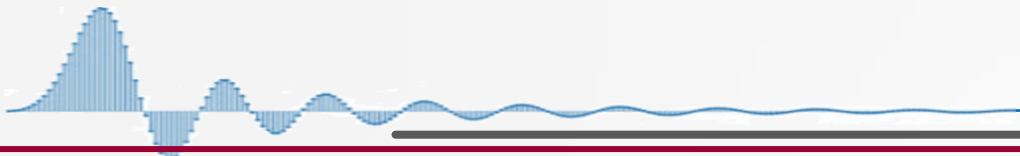
✓ Υπολογισμός του H:

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

- ✓ Ελέγχουμε αν το $C_{max} = R'$
 - ✓ Στο H εκχωρούμε το αποτέλεσμα της συνάρτησης $60^\circ * (G' - B') / \Delta \bmod 6$
- ✓ Ελέγχουμε αν το $C_{max} = G'$
 - ✓ Στο H εκχωρούμε το αποτέλεσμα της συνάρτησης $60^\circ * (B' - R') / \Delta + 2$





```
function I_HSV = HSV(I_RGB)
```

```
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
```

```
    for i=1:size(I_RGB,1)
```

```
        for j=1:size(I_RGB,2)
```

```
            ...
```

```
            if Cmax == rgb_tonos(1)
```

```
                H = 60*(mod((rgb_tonos(2)-rgb_tonos(3))/Delta,6));
```

```
            elseif Cmax == rgb_tonos(2)
```

```
                H = 60*((rgb_tonos(3)-rgb_tonos(1))/Delta) + 2;
```

```
            elseif Cmax == rgb_tonos(3)
```

```
                H = 60*((rgb_tonos(1)-rgb_tonos(2))/Delta) + 4;
```

```
            elseif Delta == 0
```

```
                H = 0;
```

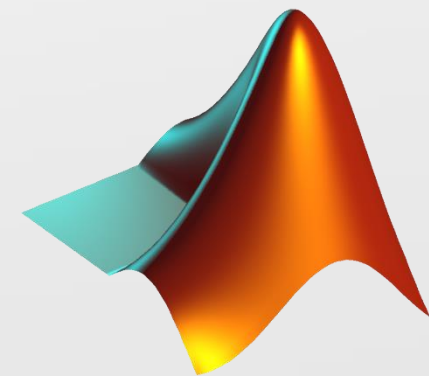
ΛΥΣΗ

✓ Υπολογισμός του H:

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \text{mod} 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

- ✓ Ελέγχουμε αν το $C_{max} = B'$
 - ✓ Στο H εκχωρούμε το αποτέλεσμα της συνάρτησης $60^\circ * (R' - G') / \Delta + 4$
- ✓ Ελέγχουμε αν το $\Delta = 0$
 - ✓ Στο H εκχωρούμε την τιμή 0





```
function I_HSV = HSV(I_RGB)
```

```
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
```

```
    for i=1:size(I_RGB,1)
```

```
        for j=1:size(I_RGB,2)
```

```
            ...
```

```
            if Cmax == rgb_tonos(1)
```

```
                H = 60*(mod((rgb_tonos(2)-rgb_tonos(3))/Delta,6));
```

```
            elseif Cmax == rgb_tonos(2)
```

```
                H = 60*((rgb_tonos(3)-rgb_tonos(1))/Delta) + 2;
```

```
            elseif Cmax == rgb_tonos(3)
```

```
                H = 60*((rgb_tonos(1)-rgb_tonos(2))/Delta) + 4;
```

```
            elseif Delta == 0
```

```
                H = 0;
```

```
            end
```

```
            I_HSV(i,j,1) = H;
```

ΛΥΣΗ

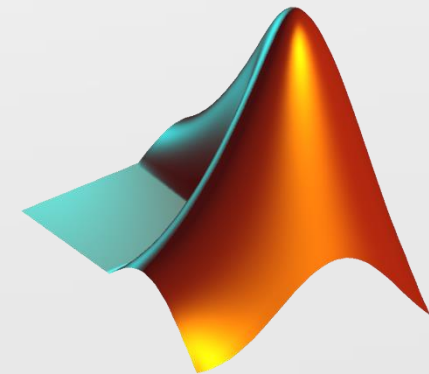
Hue calculation:

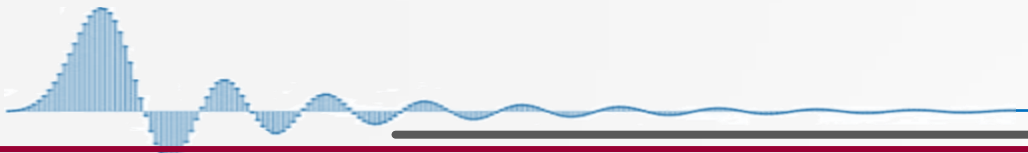
$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

✓ Υπολογισμός του H:

✓ Για κάθε i, j και για την 1^η διάσταση εκχωρώ

✓ στο I_HSV(i,j,1) = H;





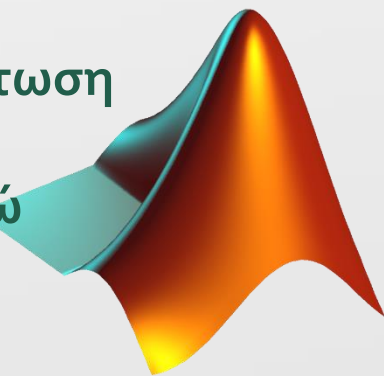
```
function I_HSV = HSV(I_RGB)
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
    for i=1:size(I_RGB,1)
        for j=1:size(I_RGB,2)
            ...
            if Cmax == 0
                S = 0;
            else
                S = double(Delta/Cmax);
            end
            I_HSV(i,j,2) = S;
```

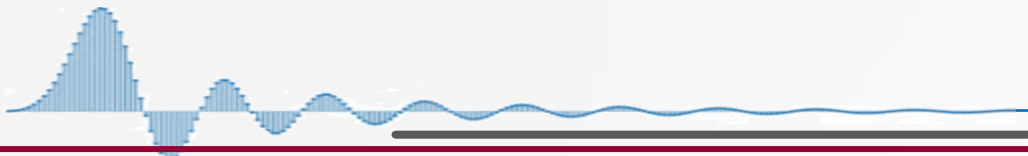
ΛΥΣΗ

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

- ✓ Υπολογισμός του **S**:
 - ✓ Ελέγχουμε την ισότητα ή όχι το C_{max} με το 0
 - ✓ Εφαρμόζουμε την εξίσωση για κάθε περίπτωση
 - ✓ Για κάθε i, j και για την 2^η διάσταση εκχωρώ
 - ✓ στο $I_HSV(i,j,2) = S$;



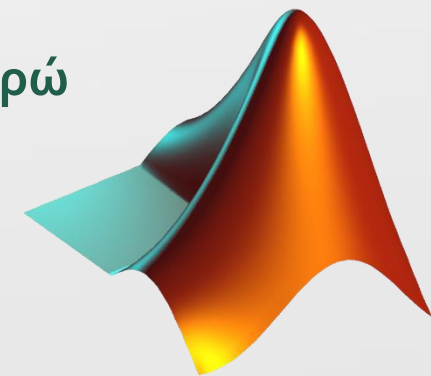


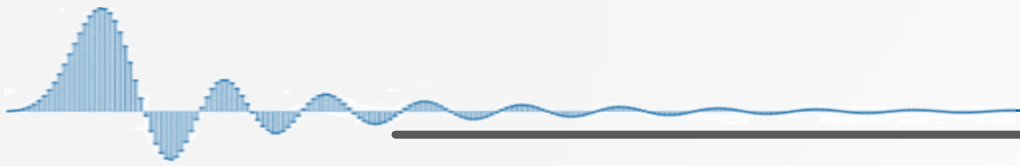
```
function I_HSV = HSV(I_RGB)
    I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));
    for i=1:size(I_RGB,1)
        for j=1:size(I_RGB,2)
            ...
            V = Cmax
            I_HSV(i,j,3) = Cmax;
```

ΛΥΣΗ

$$V = C_{max}$$

- ✓ Υπολογισμός του V:
 - ✓ Εκχωρούμε στο V το Cmax
 - ✓ Για κάθε i, j και για την 3^η διάσταση εκχωρώ
 - ✓ στο I_HSV(i,j,3) = Cmax;





ΛΥΣΗ

```
function I_HSV = HSV(I_RGB)

I_HSV = zeros(size(I_RGB,1),size(I_RGB,2),size(I_RGB,3));

for i=1:size(I_RGB,1)
    for j=1:size(I_RGB,2)
        rgb = double([I_RGB(i,j,1) I_RGB(i,j,2) I_RGB(i,j,3)]);
        rgb_tonos = double(rgb./255);
        Cmax = max(rgb_tonos);
        Cmin = min(rgb_tonos);
        Delta = Cmax - Cmin;

        if Cmax == rgb_tonos(1)
            H = 60*(mod((rgb_tonos(2)-rgb_tonos(3))/Delta,6));
        elseif Cmax == rgb_tonos(2)
            H = 60*((rgb_tonos(3)-rgb_tonos(1))/Delta) + 2;
        elseif Cmax == rgb_tonos(3)
            H = 60*((rgb_tonos(1)-rgb_tonos(2))/Delta) + 4;
        elseif Delta == 0
            H = 0;
        end
        I_HSV(i,j,1) = H;
```

```
if Cmax == 0
    S = 0;
else
    S = double(Delta/Cmax);
end
I_HSV(i,j,2) = S;

V = Cmax
I_HSV(i,j,3) = Cmax;
end
end
```

```
>> I = imread('lena.png');
>> I_hsv_custom = HSV(I_rgb);
>> figure, imshow(I_hsv_custom)
```

```
ans(:,:,1) =
    7.1287    7.1287    2.6667
    7.1287    7.1287    2.6667
    7.1287    7.1287    2.6667

ans(:,:,2) =
    0.4469    0.4469    0.4036
    0.4469    0.4469    0.4036
    0.4469    0.4469    0.4036

ans(:,:,3) =
    0.8863    0.8863    0.8745
    0.8863    0.8863    0.8745
    0.8863    0.8863    0.8745
```

