

# Επεξεργασία Εικόνας & Βίντεο

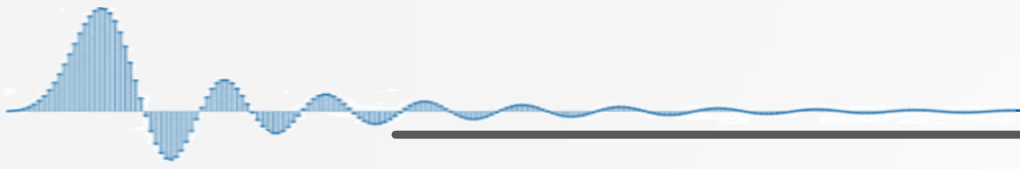


07. Κατάτμηση με Ανίχνευση  
Ακμών

Εισηγητής: Νικόλαος Γιαννακέας

*Επίκουρος Καθηγητής, Σημάτων & Συστημάτων*



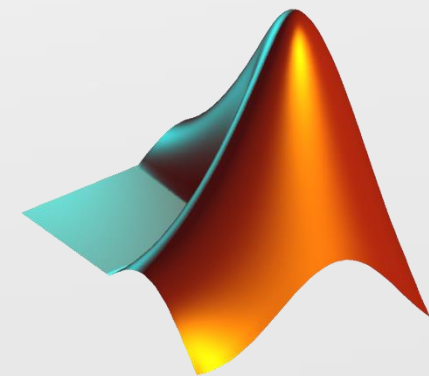


# Εισαγωγή

Η πιο βασική μέθοδος εντοπισμού των **ακμών** της εικόνας πραγματοποιείται με την **λήψη** της **παραγώγου** της **εικόνας**. Η **Ολική παράγωγος** ή **κλίση** (Gradient) της **εικόνας** (Ανάδελτα της  $f$  της Εξ. 1) συνθέτει ουσιαστικά την **κλίση της μεταβολής των φωτεινότητων** (αν δηλαδή η φωτεινότητες **αυξομειώνονται απότομα** σε **κοντινά** εικονοστοιχεία ή όχι. Ως προς την υλοποίηση της ολικής παραγώγου αυτή πραγματοποιείται με την

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix},$$

Εξ. 1



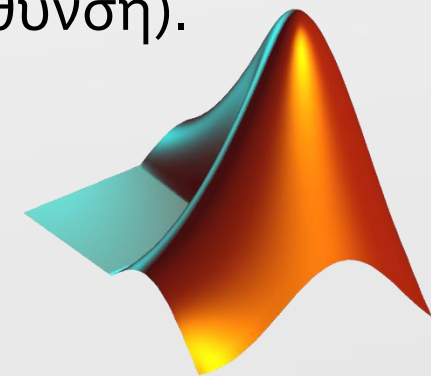


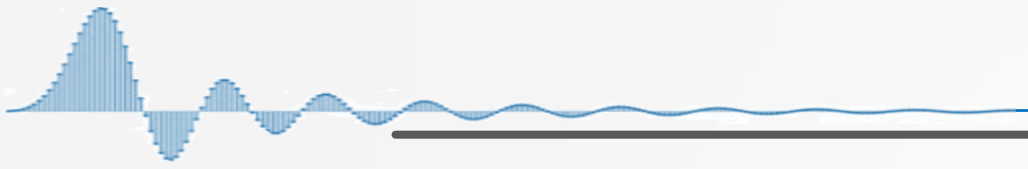
# Εισαγωγή

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix},$$

Εξ. 1

Όπου:  $g_x$  είναι η **μερική παράγωγος** ως προς  $x$  (Δηλαδή η **κλίση** της **φωτεινότητας** στην **οριζόντια** διεύθυνση), και  $g_y$  είναι η **μερική παράγωγος** ως προς  $y$  (Δηλαδή η **κλίση** της **φωτεινότητας** στην **κατακόρυφη** διεύθυνση).



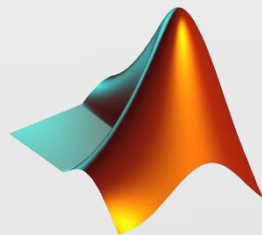


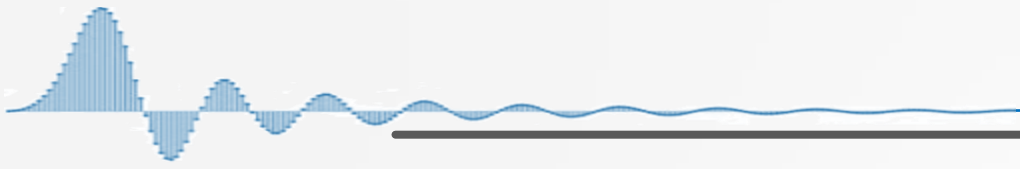
# Εισαγωγή

$$\nabla I_R = \sqrt{(I_R * G_x)^2 + (I_R * G_y)^2}$$

Εξ. 2

Δεδομένου ότι η **φωτεινότητα** μιας ψηφιακής εικόνας είναι **διακριτή συνάρτηση**, δεν μπορούν να οριστεί **παράγωγος** αυτής εκτός εάν υποθέσουμε ότι υπάρχει μια **συνεχής συνάρτηση φωτεινότητας**, η οποία έχει **δειγματοληφθεί** στα **σημεία** εικόνας. Με κάποιες επιπλέον υποθέσεις, η **παράγωγος** της συνάρτησης φωτεινότητας μπορεί να **υπολογιστεί** ως **συνάρτηση της ψηφιακής εικόνας**. Οι προσεγγίσεις αυτών των παραγώγων μπορούν να οριστούν με **διαφορετικούς βαθμούς ακρίβειας**. Ο πιο συνηθισμένος τρόπος προσέγγισης της **κλίσης** της εικόνας είναι η **συνέλιξη** της εικόνας με έναν **πυρήνα**, όπως η **Sobel** ή η **Prewitt**.

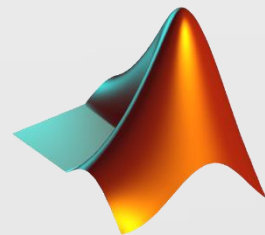


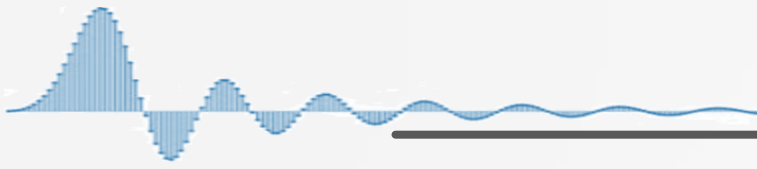


# Εισαγωγή

Ο πιο συνηθισμένος τρόπος προσέγγισης της **κλίσης** της εικόνας είναι η **συνέλιξη** της εικόνας με έναν **πυρήνα**, όπως η **Sobel** ή η **Prewitt**.

Prewitt			Sobel								
Vertical			Horizontal			Vertical			Horizontal		
-1	0	1	-1	-1	-1	-1	0	1	-1	-2	-1
-1	0	1	0	0	0	-2	0	2	0	0	0
-1	0	1	1	1	1	-1	0	1	1	2	1





# ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

## Άσκηση 1

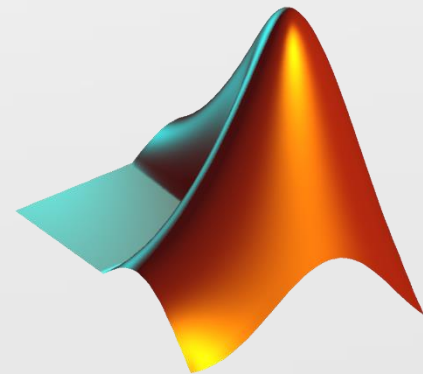
Δημιουργείστε ένα νέο script, το οποίο να εισάγει την εικόνα του 'pool.png' από τον φάκελο datasets στο workspace του λογισμικού MATLAB και αρχικά να εξάγει μόνο το «**κόκκινο**» **κανάλι** της εικόνας.

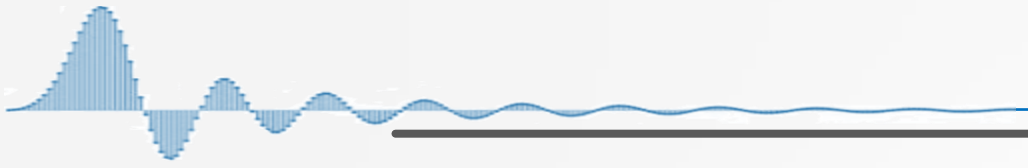
Εν συνεχεία υλοποιείστε τις **δύο μάσκες** του **Sobel** για την **οριζόντια και την κατακόρυφη** διεύθυνση αντιστοίχως.

Πραγματοποιήστε **συνέλιξη** της εικόνας μια με την μία φορά με την **πρώτη μάσκα** και μια φορά με την **δεύτερη** (βλ. υπόδειξη για την εντολή).

Για να υπολογίσετε το **ολικό διαφορικό** (ολική παράγωγο) της εικόνας υλοποιήστε της **εξίσωση** (2).

**Εμφανίστε** σε τρία διαφορετικά παράθυρα την **συνέλιξη** της εικόνας μια **δύο διαφορετικές μάσκες**, καθώς επίσης και την **τελική ολική παράγωγο**.





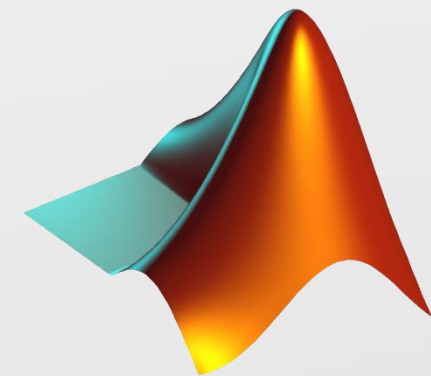
# ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

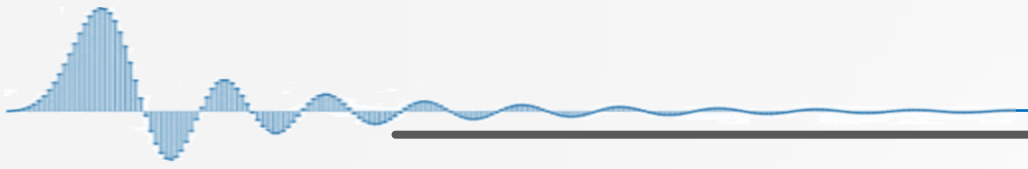
## Άσκηση 1

Τέλος, επιβεβαιώστε τα αποτελέσματα κάνοντας χρήση την **συνάρτηση** που υπάρχει υλοποιημένη στην MATLAB για τον υπολογισμό της κλίσης (**Gradient**) με μάσκα Sobel. Εμφανίστε το αποτέλεσμα σε ένα **τέταρτο** παράθυρο.

### Υπόδειξη:

- i) Για την **συνέλιξη** της εικόνας με τις δύο μάσκες χρησιμοποιήστε την συνάρτηση για δυοδιάστατη συνέλιξη **conv2( )**.
- ii) Η **συνάρτηση** της MATLAB που υλοποιεί **[Gmag, Gdir] = imgradient(IR, 'sobel');**





# ΛΥΣΗ

```
I_rgb = imread('pool.png');
```

```
IR = I_rgb(:,:,1);
```

```
Gx = [1 0 -1; +2 0 -2; 1 0 -1];
```

```
Gy = [1 2 1; 0 0 0; -1 -2 -1];
```

→ Διάβασμα εικόνας

→ Για ευκολία θα επεξεργαστούμε  
μόνο το κόκκινο κανάλι

Δύο μάσκες του Sobel για την οριζόντια και  
την κατακόρυφη διεύθυνση αντιστοίχως.

Vertical

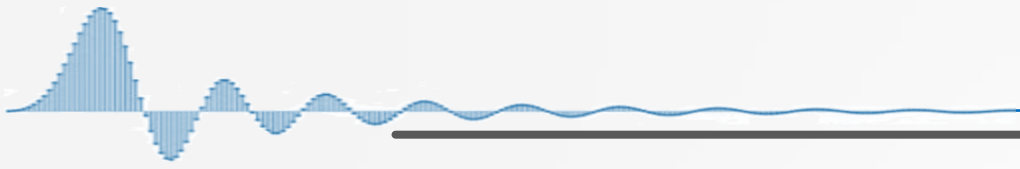
-1	0	1
-2	0	2
-1	0	1

Sobel

Horizontal

-1	-2	-1
0	0	0
1	2	1





# ΛΥΣΗ

```
I_rgb = imread('pool.png');
```

```
IR = I_rgb(:,:,1);
```

```
Gx = [1 0 -1; +2 0 -2; 1 0 -1];
```

```
Gy = [1 2 1; 0 0 0; -1 -2 -1];
```

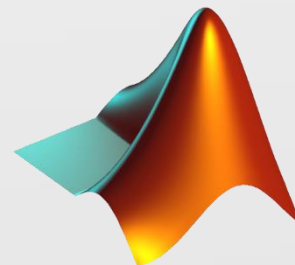
```
I_grad_x = conv2(IR, Gx);
```

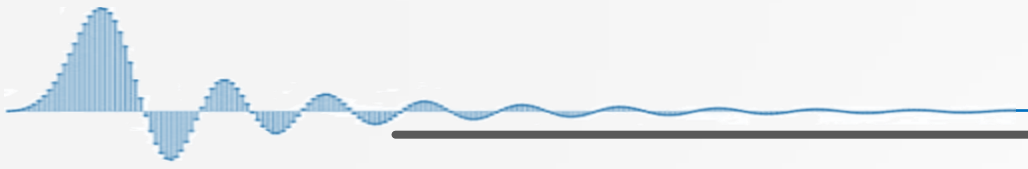
```
I_grad_y = conv2(IR, Gy);
```

→ **Vertical Direction (μερική παράγωγος)**

→ **Horizontal Direction (μερική παράγωγος)**

Συνέλιξη της εικόνας με κάθε μια από της μάσκες του Sobel



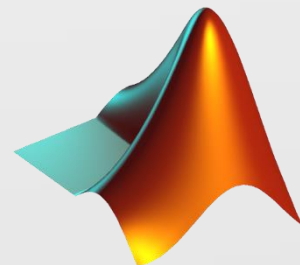


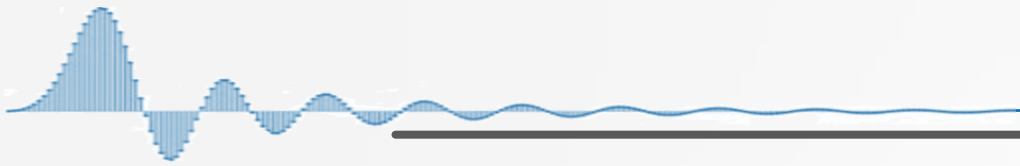
# ΛΥΣΗ

```
I_rgb = imread('pool.png');  
IR = I_rgb(:,:,1);  
Gx = [1 0 -1; +2 0 -2; 1 0 -1];  
Gy = [1 2 1; 0 0 0; -1 -2 -1];  
I_grad_x = conv2(IR, Gx);  
I_grad_y = conv2(IR, Gy);  
  
I_grad = sqrt(I_grad_x.^2 + I_grad_y.^2);
```

Συγχώνευση (πληρες διαφορικό)

$$\nabla I_R = \sqrt{(I_R * G_x)^2 + (I_R * G_y)^2}$$





# ΛΥΣΗ

```
I_rgb = imread('pool.png');  
IR = I_rgb(:, :, 1);  
Gx = [1 0 -1; +2 0 -2; 1 0 -1];  
Gy = [1 2 1; 0 0 0; -1 -2 -1];  
I_grad_x = conv2(IR, Gx);  
I_grad_y = conv2(IR, Gy);  
I_grad = sqrt(I_grad_x.^2 + I_grad_y.^2);
```

```
figure, imshow(uint8(I_grad_x))
```

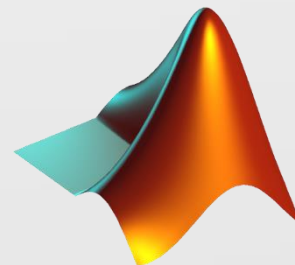
```
figure, imshow(uint8(I_grad_y))
```

```
figure, imshow(uint8(I_grad))
```

→ Εμφάνιση εικόνας κατακόρυφης παραγώγου

→ Εμφάνιση εικόνας οριζόντιας παραγώγου

→ Εμφάνιση εικόνας πλήρες διαφορικό

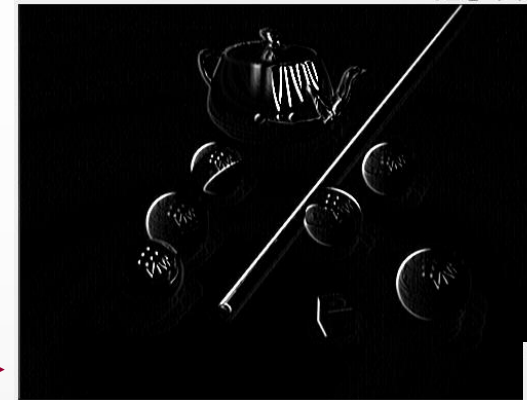




## ΛΥΣΗ

```
I_rgb = imread('pool.png');  
IR = I_rgb(:,:,1);  
Gx = [1 0 -1; +2 0 -2; 1 0 -1];  
Gy = [1 2 1; 0 0 0; -1 -2 -1];  
I_grad_x = conv2(IR, Gx);  
I_grad_y = conv2(IR, Gy);  
I_grad = sqrt(I_grad_x.^2 + I_grad_y.^2);
```

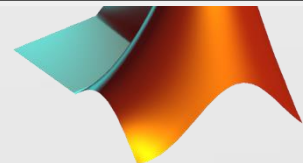
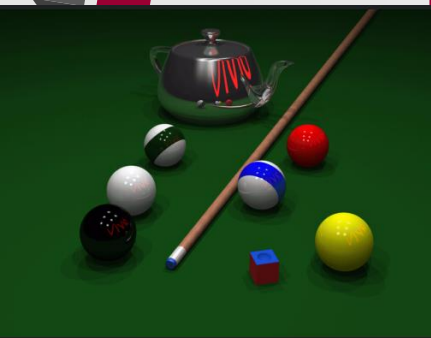
```
figure, imshow(uint8(I_grad_x))
```

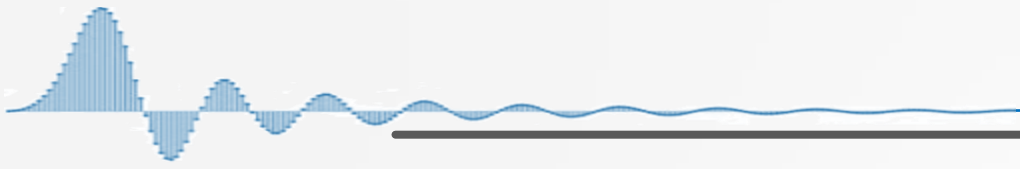


```
figure, imshow(uint8(I_grad_y))
```



```
figure, imshow(uint8(I_grad))
```

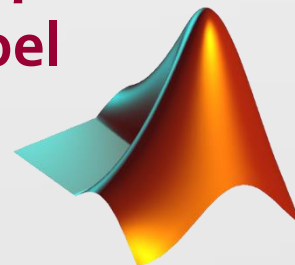


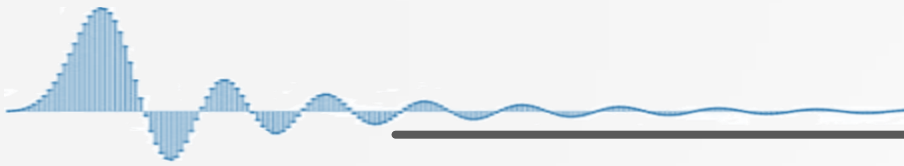


```
I_rgb = imread('pool.png');  
IR = I_rgb(:, :, 1);  
Gx = [1 0 -1; +2 0 -2; 1 0 -1];  
Gy = [1 2 1; 0 0 0; -1 -2 -1];  
I_grad_x = conv2(IR, Gx);  
I_grad_y = conv2(IR, Gy);  
I_grad = sqrt(I_grad_x.^2 + I_grad_y.^2);  
figure, imshow(uint8(I_grad_x))  
figure, imshow(uint8(I_grad_y))  
figure, imshow(uint8(I_grad))  
  
[Gmag, Gdir] = imgradient(IR, 'sobel');  
figure, imshow(uint8(Gmag))
```



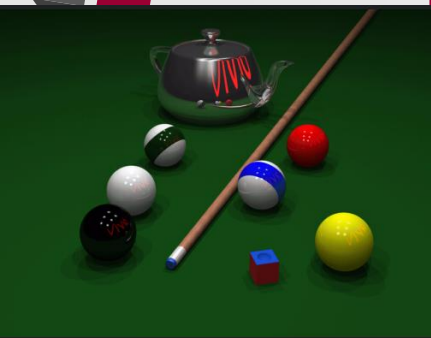
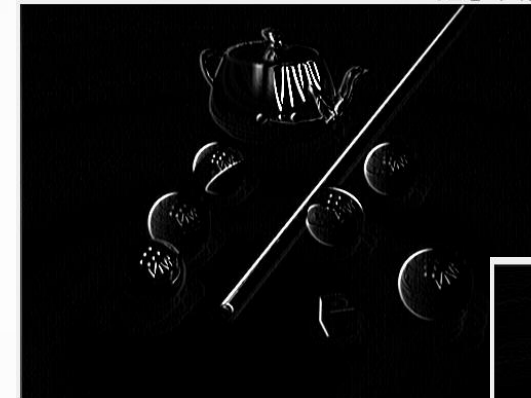
**Υλοποίηση της MATLAB για  
εντοπισμό ακμών με Sobel**

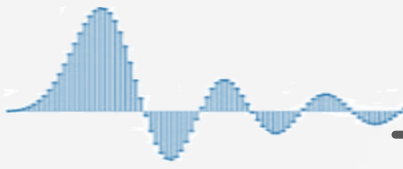




```
I_rgb = imread('pool.png');  
IR = I_rgb(:,:,1);  
Gx = [1 0 -1; +2 0 -2; 1 0 -1];  
Gy = [1 2 1; 0 0 0; -1 -2 -1];  
I_grad_x = conv2(IR, Gx);  
I_grad_y = conv2(IR, Gy);  
I_grad = sqrt(I_grad_x.^2 + I_grad_y.^2);  
figure, imshow(uint8(I_grad_x))  
figure, imshow(uint8(I_grad_y))  
figure, imshow(uint8(I_grad))  
  
[Gmag, Gdir] = imgradient(IR, 'sobel');  
figure, imshow(uint8(Gmag))
```

ΛΥΣΗ





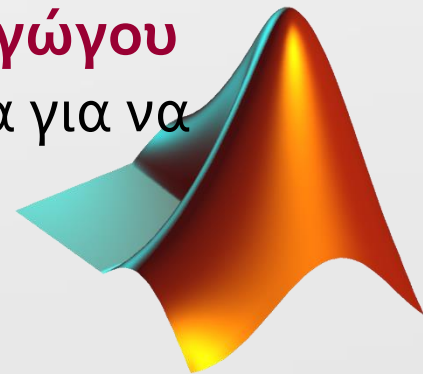
# ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

## Άσκηση 2

Εισάγετε στο workspace την εικόνα της **Lena**, και κρατήστε μόνο το **κόκκινο** κανάλι για να έχετε μια εικόνα επιπέδων του **γκρι**. Αξιοποιήστε την εντολή **edge()**, η οποία συγκεντρώνει όλες τις μεθόδους **ανίχνευσης ακμών** που διδαχτήκατε στην θεωρία.

Δείτε την σύνταξη της εντολής είτε από το help της Matlab είτε από τον δικτυακό τόπο της Mathworks.

Εκτελέστε για το κόκκινο κανάλι της **Lena** ανίχνευση ακμών με τις μεθόδους **1ης παραγώγου** (Roberts, Sobel, Prewitt), **2ης Παραγώγου** (LoG), και την μέθοδο του **Canny**. Εμφανίστε τα αποτελέσματα για να συγκρίνετε τα αποτελέσματα.





## ΛΥΣΗ

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

```
Ired = I(:,:,1);
```

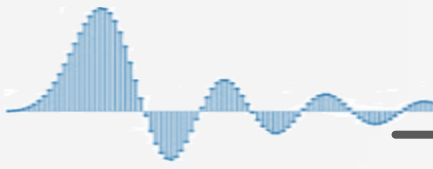


**Διάβασμα εικόνας**



**Για ευκολία θα επεξεργαστούμε  
μονο το κόκκινο κανάλι =  
grayscale**





# Ανίχνευση Ακμών με Παραγωγή 1<sup>η</sup> Παράγωγος

Μέθοδος Prewitt

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Μέθοδος Sobel

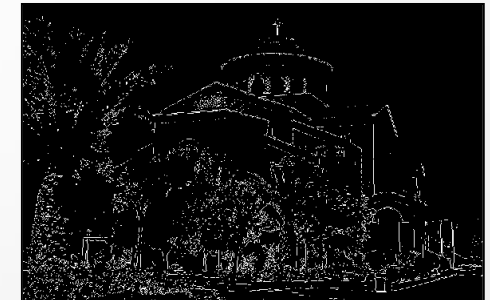
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Μέθοδος Roberts

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$



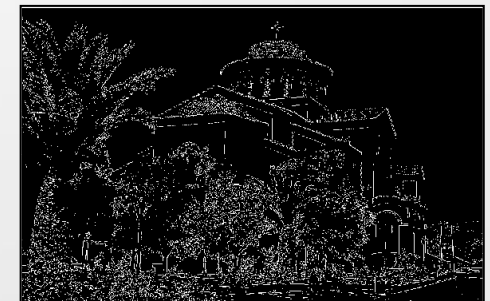
Αρχική εικόνα



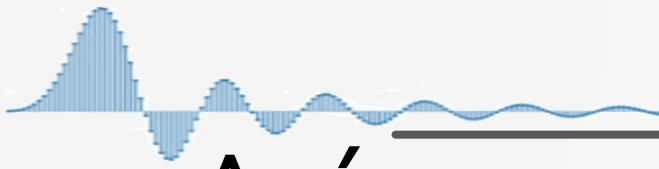
Μέθοδος Prewitt



Μέθοδος Sobel

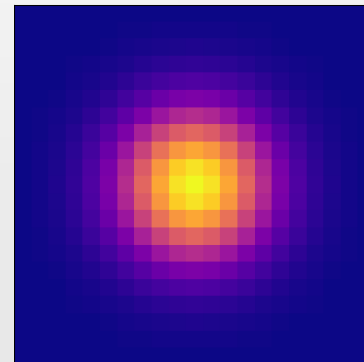
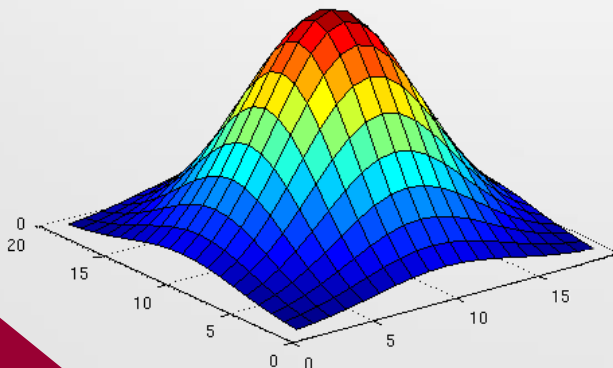
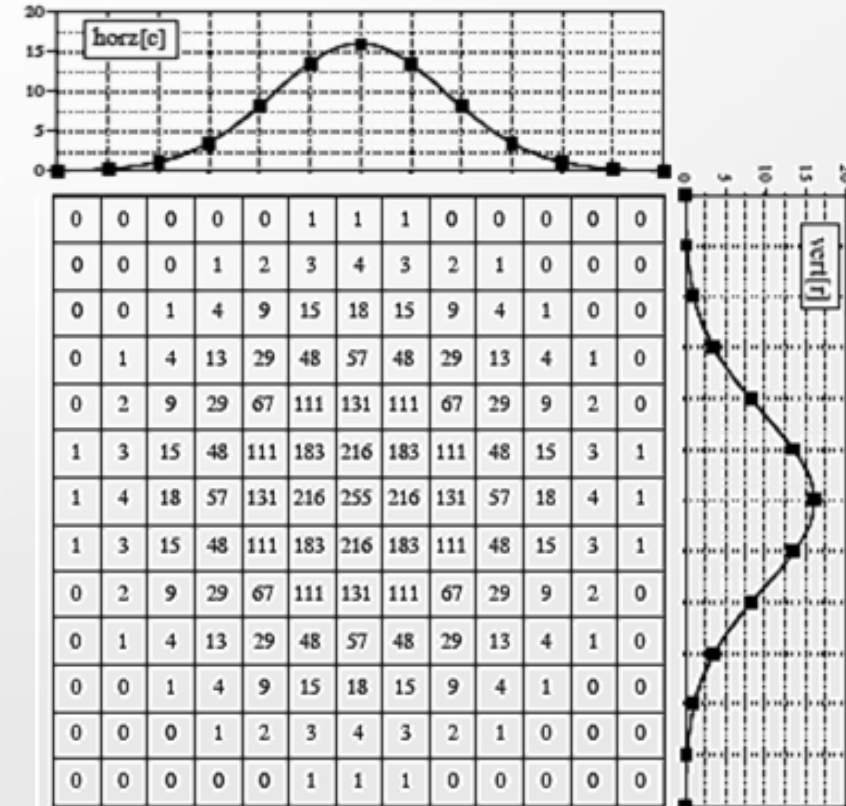


Μέθοδος Roberts



# Ανίχνευση Ακμών με Παραγωγή Λαπλασιανή της Γκαουσιανής - LoG

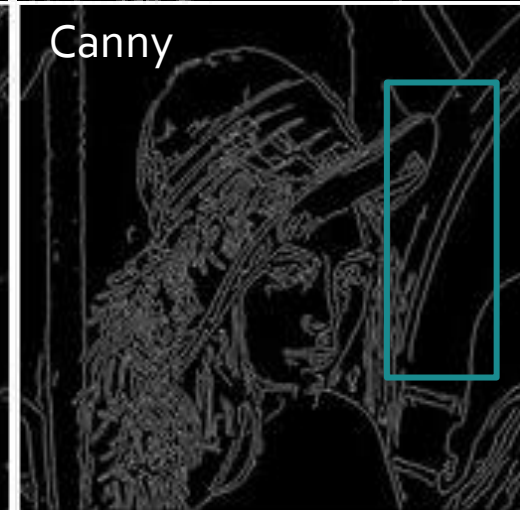
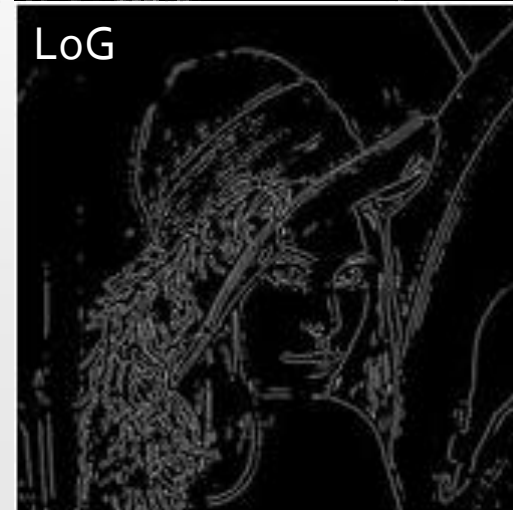
Ο συνδυασμός του Γκαουσιανού φίλτρου με τον Λαπλασιανό τελεστή της δεύτερης παραγωγού αναφέρεται στην βιβλιογραφία ως Λαπλασιανή της Γκαουσιανής (Laplacian of Gaussian – LoG)

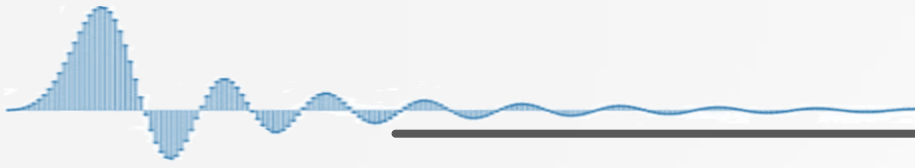




# Σύγκριση μεθόδων Ανίχνευσης Ακμών

- Η μέθοδος Canny Υπερτερεί καθώς και έχει απαλείψει τον περισσότερο θόρυβο ενώ έχει διατηρήσει όλες τις περισσότερες σημαντικές ακμές





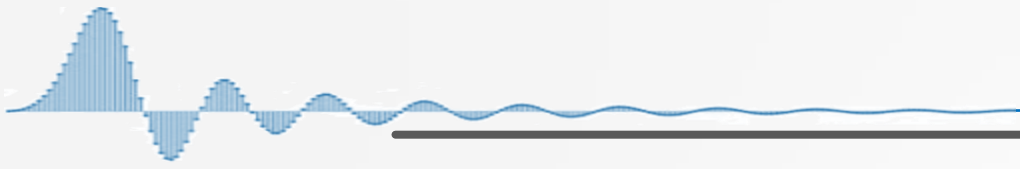
# Σύγκριση μεθόδων Ανίχνευσης Ακμών

edge

Βρείτε άκρες σε εικόνα 2-D σε κλίμακα του γκρι

$BW = \text{edge}(I, \text{method})$

Ανιχνεύει άκρες στην εικόνα  $I$  χρησιμοποιώντας τον αλγόριθμο ανίχνευσης ακμών που καθορίζεται από το `method`.

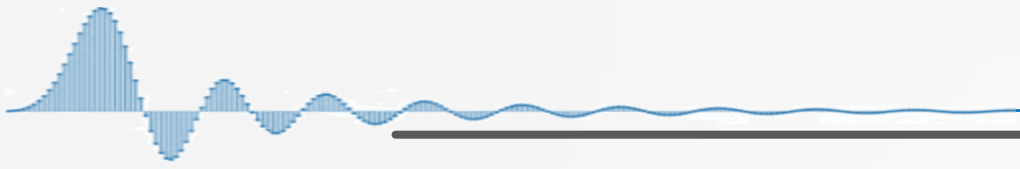


# Σύγκριση μεθόδων Ανίχνευσης Ακμών

$$BW = \text{edge}(I, \text{method})$$

Ανιχνεύει άκρες στην εικόνα  $I$  χρησιμοποιώντας τον αλγόριθμο ανίχνευσης ακμών που καθορίζεται από το  $\text{method}$ .

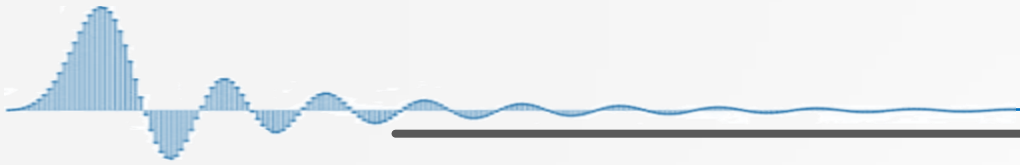
Μέθοδος	Περιγραφή
"sobel"	Βρίσκει ακμές σε εκείνα τα σημεία όπου η κλίση της εικόνας είναι μέγιστη, χρησιμοποιώντας την προσέγγιση Sobel στην παράγωγο.
"Prewitt"	Βρίσκει ακμές σε εκείνα τα σημεία όπου η διαβάθμιση του $I$ είναι μέγιστη, χρησιμοποιώντας την προσέγγιση Prewitt στην παράγωγο.
"Roberts"	Βρίσκει ακμές σε εκείνα τα σημεία όπου η διαβάθμιση του $I$ είναι μέγιστη, χρησιμοποιώντας την προσέγγιση Roberts στην παράγωγο.
"log"	Βρίσκει άκρες αναζητώντας μηδενικές διασταυρώσεις μετά το φιλτράρισμα $I$ με ένα φίλτρο Laplacian of Gaussian (LoG).
"zerocross"	Βρίσκει άκρες αναζητώντας μηδενικές διασταυρώσεις μετά το φιλτράρισμα $I$ με ένα φίλτρο που καθορίζετε, $h$
"Canny"	Βρίσκει ακμές αναζητώντας τοπικά μέγιστα της κλίσης του $I$ . ο $\text{edgeH}$ συνάρτηση υπολογίζει τη διαβάθμιση χρησιμοποιώντας την παράγωγο ενός φίλτρου Gauss. Αυτή η μέθοδος χρησιμοποιεί δύο κατώφλια για την ανίχνευση ισχυρών και αδύναμων άκρων, συμπεριλαμβανομένων των αδύναμων άκρων στην έξοδο, εάν συνδέονται με ισχυρές ακμές. Χρησιμοποιώντας δύο κατώφλια, η μέθοδος Canny είναι λιγότερο πιθανό από τις άλλες μεθόδους να ξεγελαστεί από το θόρυβο και πιο πιθανό να ανιχνεύσει πραγματικά αδύναμα άκρα.
"approxcanny"	Βρίσκει άκρες χρησιμοποιώντας μια κατά προσέγγιση έκδοση του αλγόριθμου ανίχνευσης άκρων Canny που παρέχει ταχύτερο χρόνο εκτέλεσης σε βάρος του λιγότερο ακριβούς εντοπισμού. Οι εικόνες κινητής υποδιαστολής αναμένεται να κανονικοποιηθούν στο εύρος [0, 1].



# ΛΥΣΗ

```
I = imread('lena.png');  
Ired = I(:,:,1);  
  
I_bin_roberts = edge(Ired,'Roberts');  
I_bin_sobel = edge(Ired,'Sobel');  
I_bin_prewitt = edge(Ired,'Prewitt');  
I_bin_log = edge(Ired,'log');  
I_bin_canny = edge(Ired,'Canny');
```

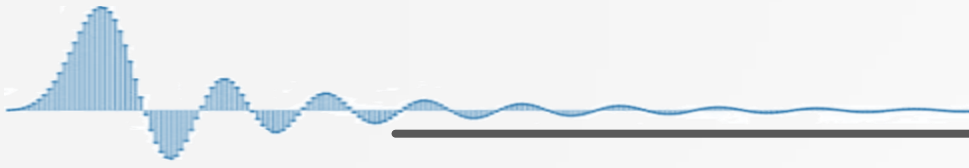
**Μέθοδοι ανίχνευσης ακμών**



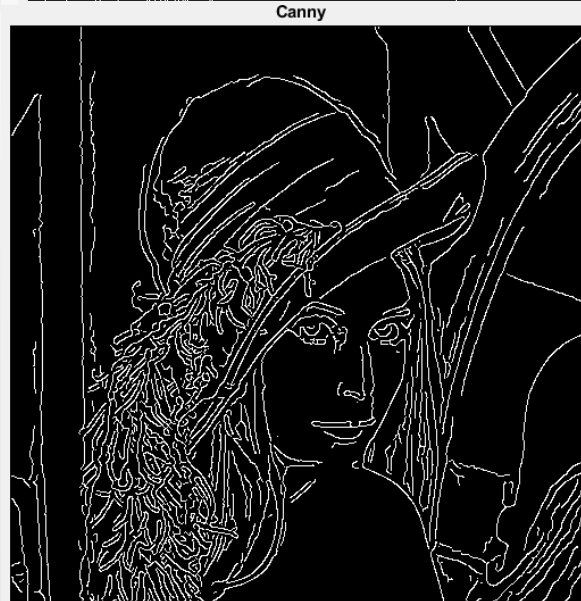
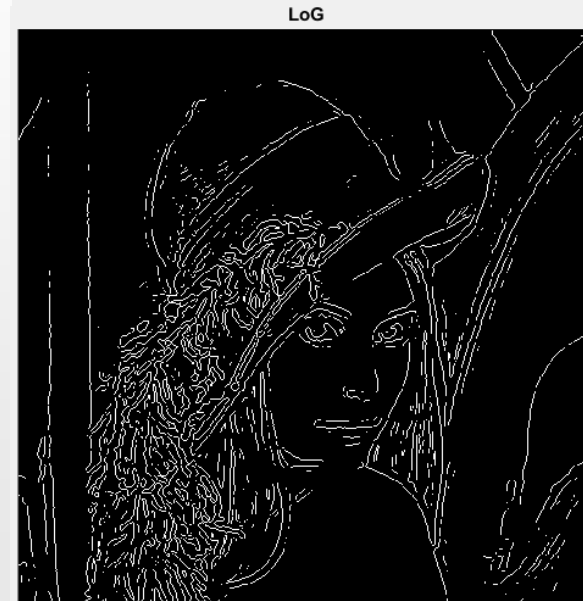
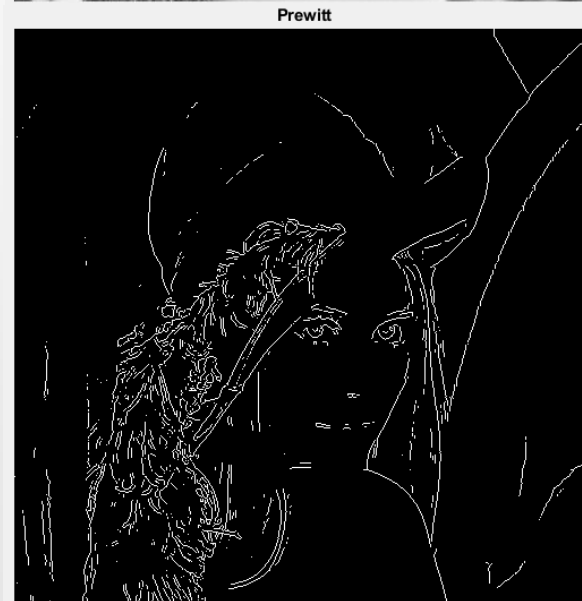
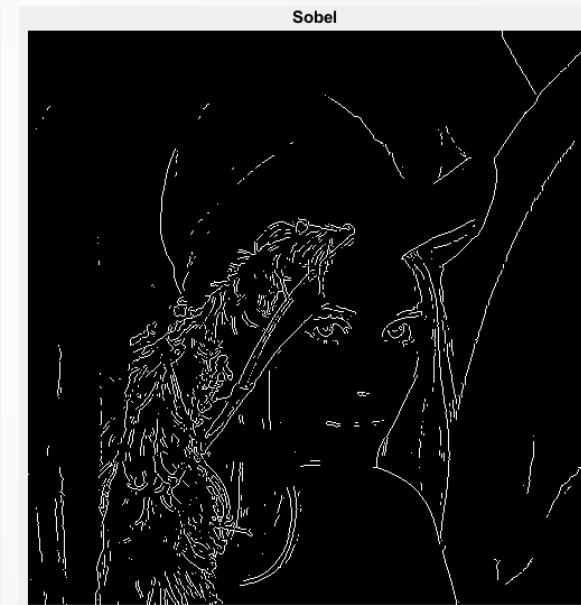
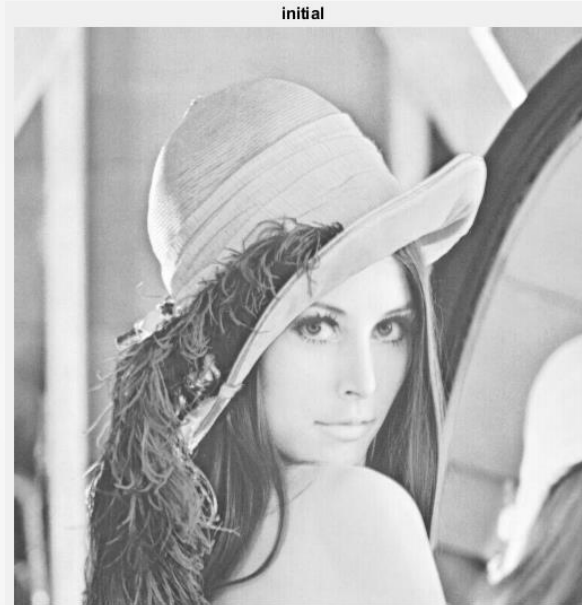
```
figure, imshow(Ired)
title('initial')
figure, imshow(I_bin_roberts)
title('Roberts')
figure, imshow(I_bin_sobel)
title('Sobel')
figure, imshow(I_bin_prewitt)
title('Prewitt')
figure, imshow(I_bin_log)
title('LoG')
figure, imshow(I_bin_canny)
title('Canny')
```

# ΛΥΣΗ

**Εμφάνιση εικόνων σε  
ξεχωριστά παράθυρα με τίτλο  
στο κάθε ένα**



```
figure, imshow(Ired)
title('initial')
figure, imshow(I_bin_roberts)
title('Roberts')
figure, imshow(I_bin_sobel)
title('Sobel')
figure, imshow(I_bin_prewitt)
title('Prewitt')
figure, imshow(I_bin_log)
title('LoG')
figure, imshow(I_bin_canny)
title('Canny')
```





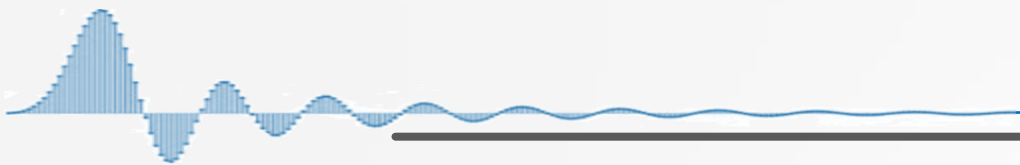


```
figure,  
subplot(3,2,1), imshow(Ired)  
subplot(3,2,2), imshow(I_bin_roberts)  
subplot(3,2,3), imshow(I_bin_sobel)  
subplot(3,2,4), imshow(I_bin_prewitt)  
subplot(3,2,5), imshow(I_bin_log)  
subplot(3,2,6), imshow(I_bin_canny)
```



## ΛΥΣΗ

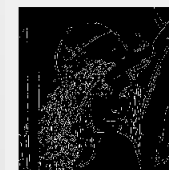
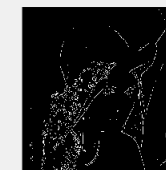
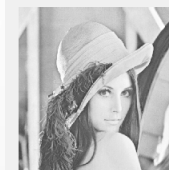
**Εμφάνιση εικόνων σε κοινό παράθυρο**

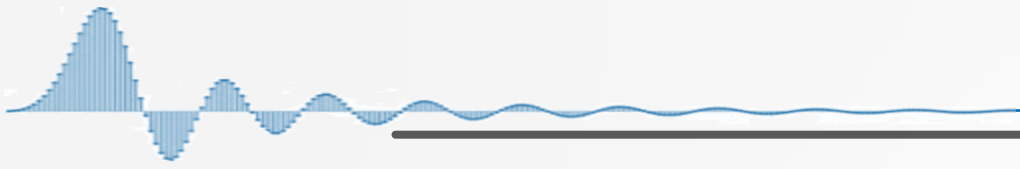


```
figure,  
subplot(3,2,1), imshow(Ired)  
subplot(3,2,2), imshow(I_bin_roberts)  
subplot(3,2,3), imshow(I_bin_sobel)  
subplot(3,2,4), imshow(I_bin_prewitt)  
subplot(3,2,5), imshow(I_bin_log)  
subplot(3,2,6), imshow(I_bin_canny)
```

## ΛΥΣΗ

### Εμφάνιση εικόνων σε κοινό παράθυρο





# ΛΥΣΗ

```
I = imread('lena.png');  
Ired = I(:,:,1);  
  
I_bin_roberts = edge(Ired, 'Roberts');  
I_bin_sobel = edge(Ired, 'Sobel');  
I_bin_prewitt = edge(Ired, 'Prewitt');  
I_bin_log = edge(Ired, 'log');  
I_bin_canny = edge(Ired, 'Canny');
```

```
figure,  
subplot(3,2,1), imshow(Ired)  
subplot(3,2,2), imshow(I_bin_roberts)  
subplot(3,2,3), imshow(I_bin_sobel)  
subplot(3,2,4), imshow(I_bin_prewitt)  
subplot(3,2,5), imshow(I_bin_log)  
subplot(3,2,6), imshow(I_bin_canny)
```

