

# ΟΡΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

8ο ΕΞΑΜΗΝΟ 2019-2020

## 1η Εργαστηριακή Άσκηση

Καθηγητής: Πέτρος Μαραγκός

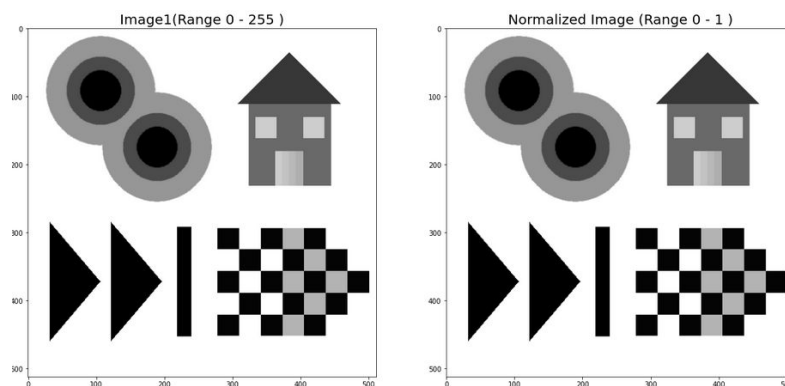
Φοιτητές: Γεώργιος-Ταξιάρχης Γιαννιός, Α.Μ.:03116156

Δημήτριος Κρανιάς, Α.Μ.:03116030

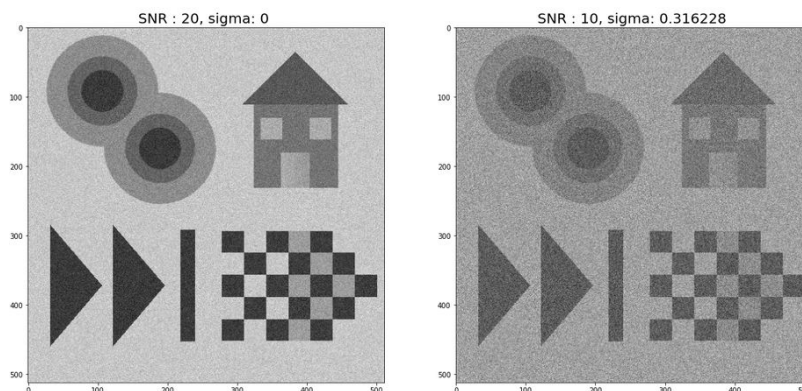
### Μέρος 1: Ανίχνευση Ακμών σε Γκριζες Εικόνες

#### 1.1 Δημιουργία Εικόνων Εισόδου

1.1.1. Αρχικά διαβάσαμε την εικόνα, και την κανονικοποιήσαμε στο διάστημα  $[0,1]$



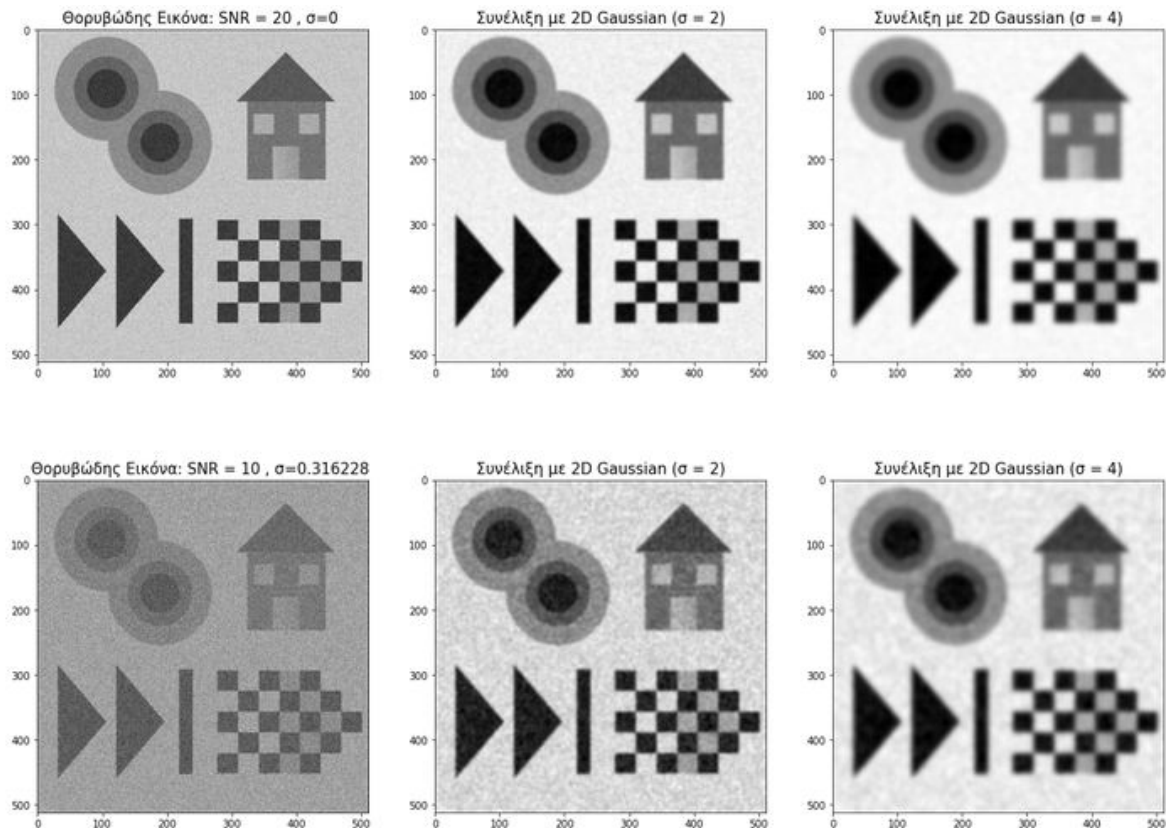
1.1.2. Στην συνέχεια προσθέσαμε λευκό *gaussian* θόρυβο με μηδενική μέση τιμή και δημιουργήσαμε δύο εικόνες με  $\sigma_1$  ώστε  $PSNR = 20dB$  (Εικόνα  $I_{\sigma1}$ ) και  $\sigma_2$  ώστε  $PSNR = 10dB$  (Εικόνα  $I_{\sigma2}$ )



Όπως είναι αναμενόμενο: Μικρότερο SNR -> Πιο θορυβώδης εικόνα

## 1.2. Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών

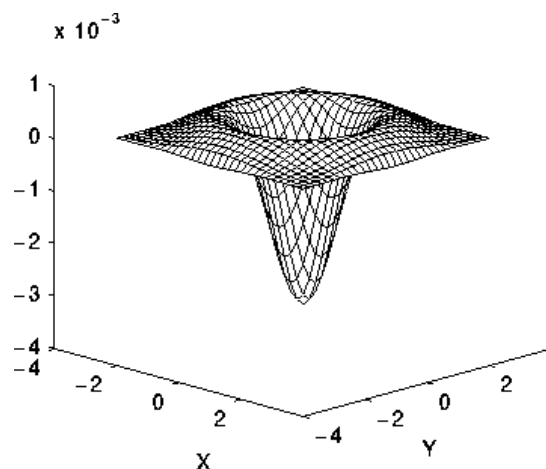
1.2.1 Στο σημείο αυτό υπολογίσαμε την συνέλιξη των εικόνων Ισ1 με Δισδιάστατη Gaussian με τιμές ( $\sigma = 2$  και  $\sigma = 4$ ). Την ίδια διαδικασία ακολουθήσαμε για την Ισ2:



Γίνεται αντιληπτό ότι η συνέλιξη της θορυβώδους εικόνας με Δισδιάστατη Gaussian με τυπική απόκλιση  $\sigma$  κοντά στο 2, εξομαλύνει ικανοποιητικά και τις δύο εικόνες,

1.2.2 Έπειτα προσεγγίσαμε την Laplacian L της εξομαλυμένης εικόνας χρησιμοποιώντας 2 διαφορετικές εναλλακτικές (Μη γραμμικά - Γραμμικά)

Γραμμικά

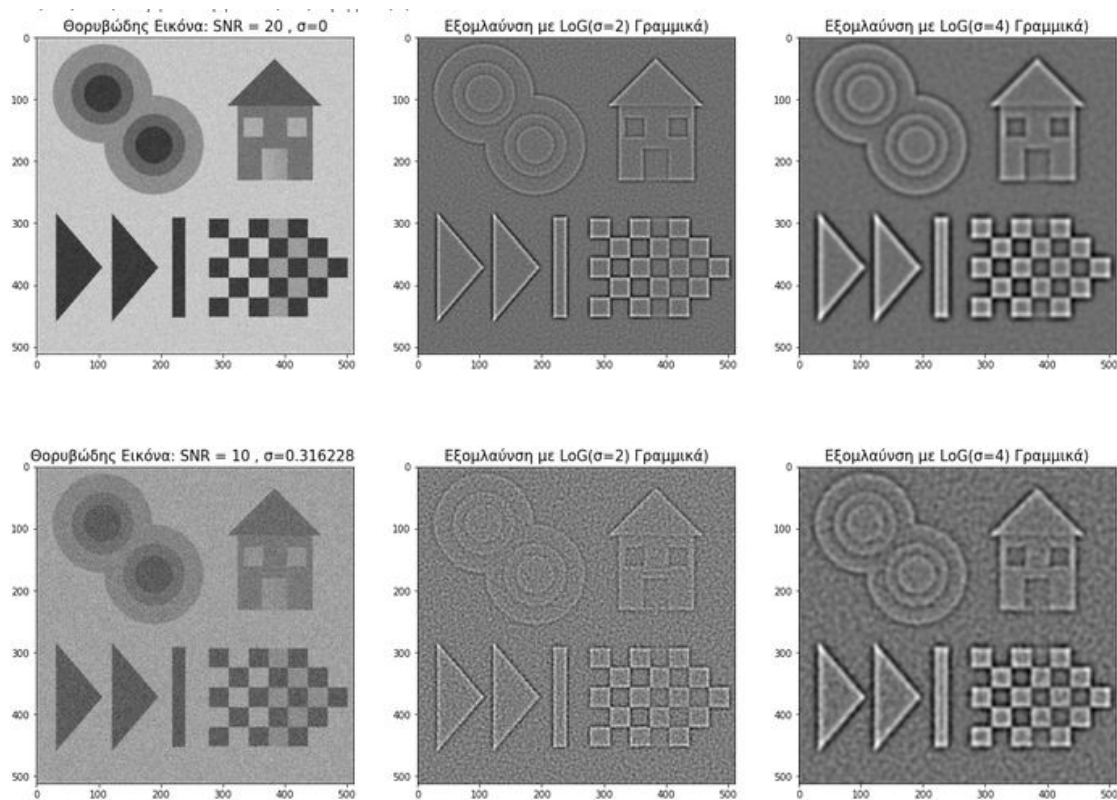


Μη Γραμμικά

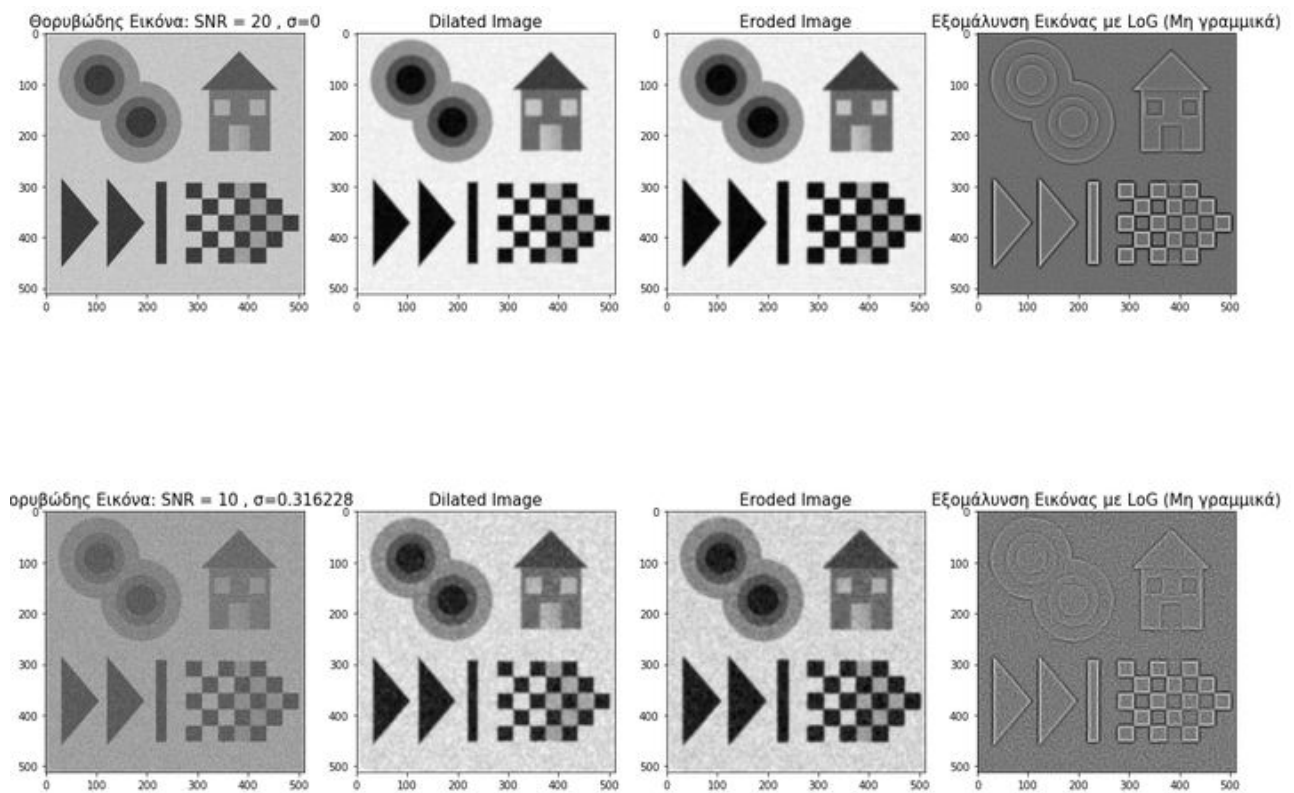
$$B = \begin{bmatrix} & \bullet & \\ \bullet & \bullet & \bullet \\ & \bullet & \end{bmatrix}$$

$$L_2 = I_\sigma \oplus B + I_\sigma \ominus B - 2I_\sigma,$$

## 1. Γραμμικά



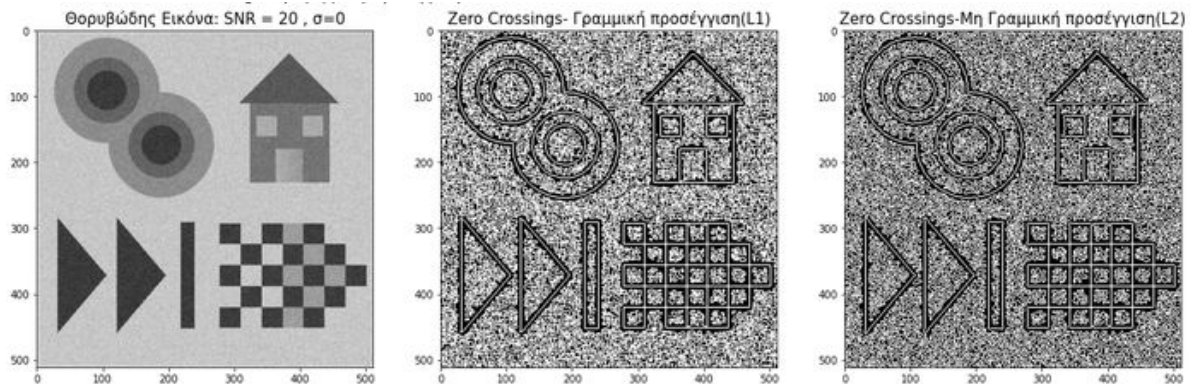
## 2. Μη Γραμμικά



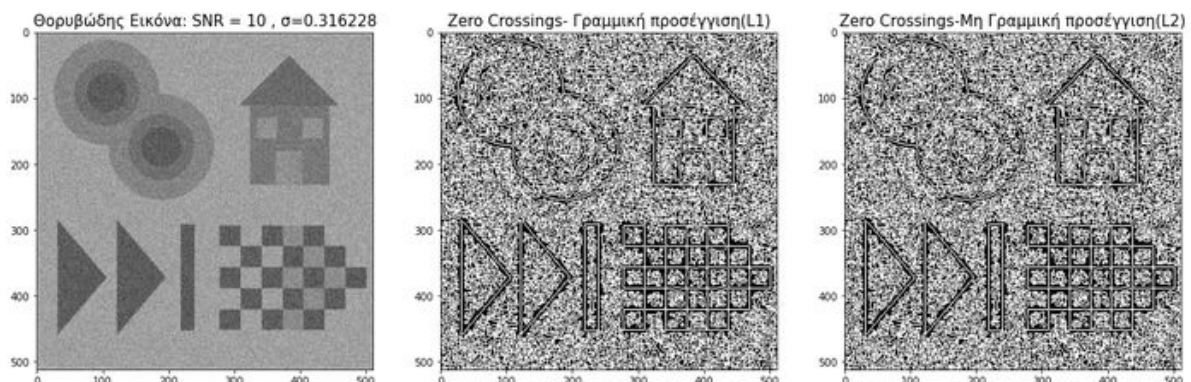
Παρατηρούμε στο **μη γραμμικό τρόπο** μία όξυνση των ακμών και προδιάθεση για καλύτερα αποτελέσματα ανίχνευσης ακμών.

**1.2.3** Στην συνέχεια εντοπίσαμε τα Zero Crossings και για τις δύο θορυβώδεις εικόνες (Γραμμικά και Μη Γραμμικά)

### 1. Πρώτη Θορυβώδης (Iσ1)

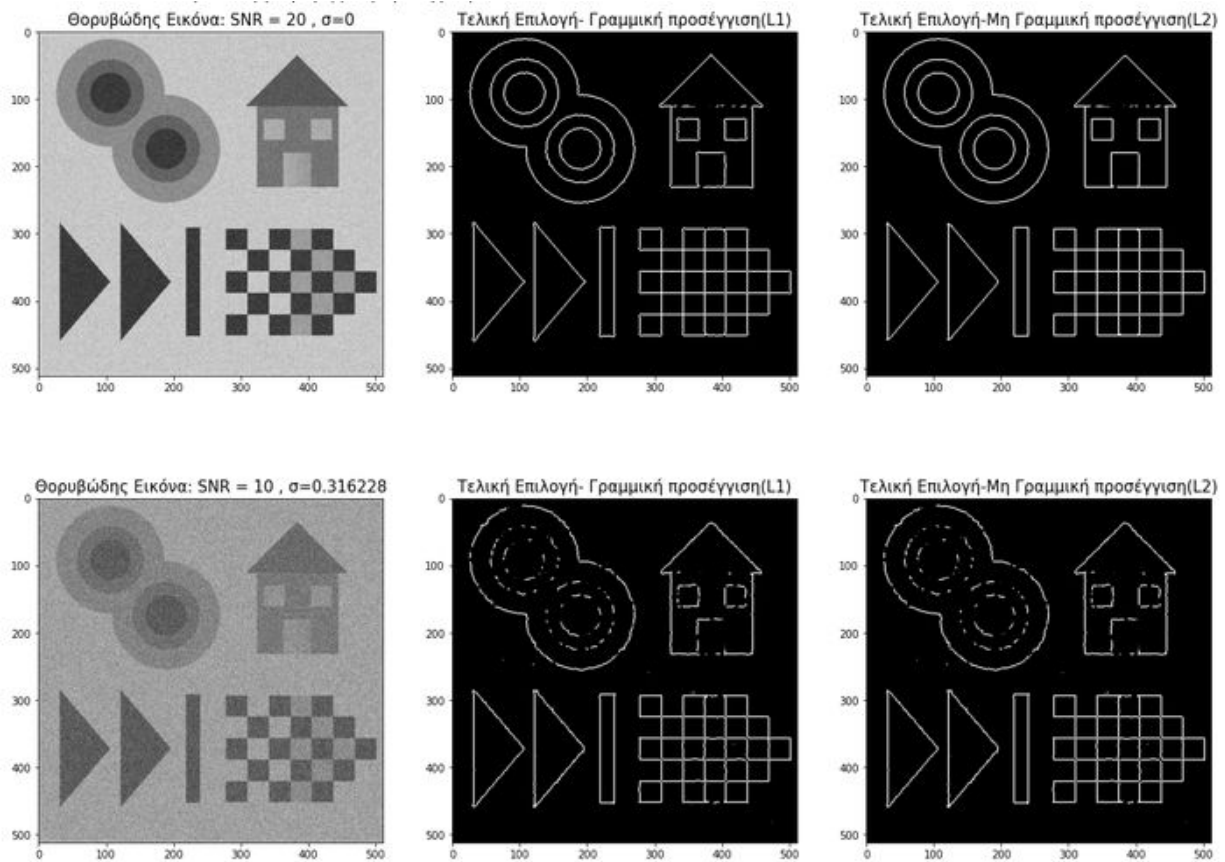


### Δεύτερη Θορυβώδης (Iσ2)



Στην μέθοδο με Γραμμική Προσέγγιση L, μετά την Δημιουργία Δυαδικής Εικόνας Προσέμου X, και την εύρεση του περιγράμματος Y, παρατηρεί κανείς ότι θα απορροφηθούν αρκετά σημεία (λευκά pixels), τα οποία όμως συνιστούν μέρος ακμών

**1.2.4** Τέλος απορρίψαμε τα παραπάνω zero crossings σημεία σε σχετικά ομαλές περιοχές. Τα αποτελέσματα για τις δύο θορυβώδεις εικόνες:



Όλα τα παραπάνω ομαδοποιήθηκαν καθ' υπόδειξη σε μια συνάρτηση Edge Detect ()

### **1.3. Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών**

**1.3.1** Για να αξιολογήσουμε τον αλγόριθμό μας καλέσαμε την συνάρτηση για την πραγματική εικόνα(χωρίς θόρυβο) και υπολογίσαμε τις αληθινές ακμές:



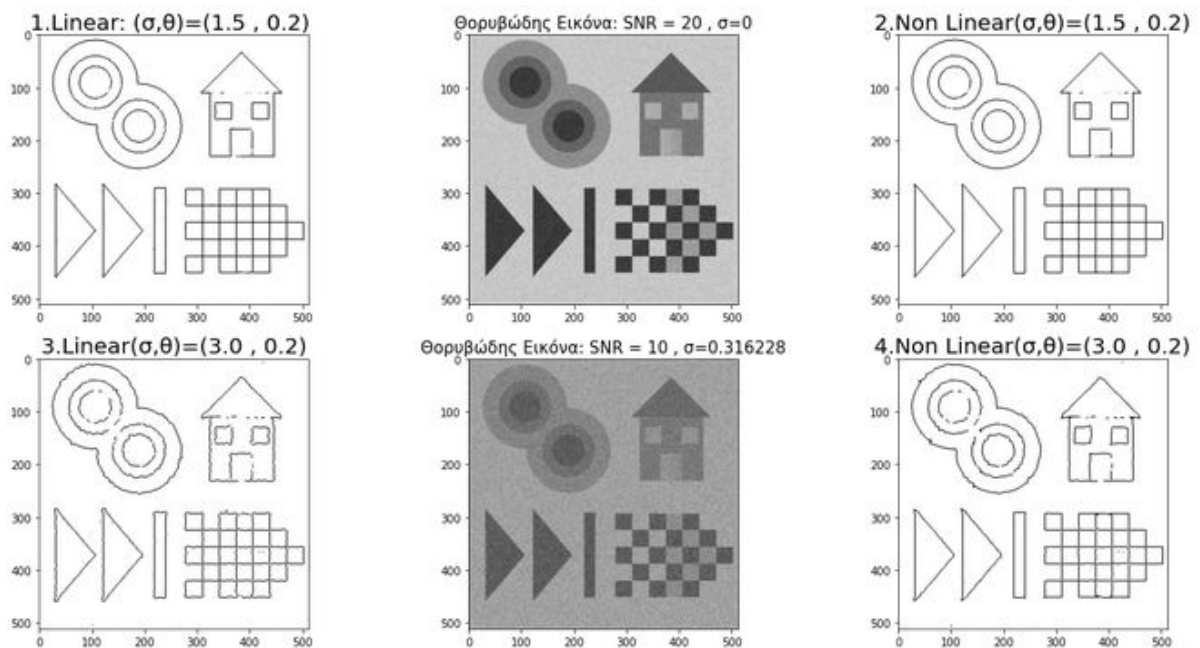
**1.3.2** Στη συνέχεια υλοποιήσαμε μια διαδικασία evaluation() για εξαγωγή κριτηρίου ποιότητας σύμφωνα με τη σχέση:

$$C = [Pr(D|T) + Pr(T|D)] / 2$$

Το ημίθροισμα του ποσοστού των ανιχνευθεισών ακμών που είναι αληθινές (Precision), και του ποσοστού των αληθινών ακμών που ανιχνεύθηκαν (Recall))

**1.3.3** Με τη χρήση αυτής της διαδικασίας αλλά και της Edge Detect πήραμε τα ακόλουθα αποτελέσματα καλώντας την συνάρτηση για τις τιμές που φαίνεται στους επιμέρους τίτλους

Outputs:



Evaluation:

PSNR =20 sigma = 1.5 theta = 0.2

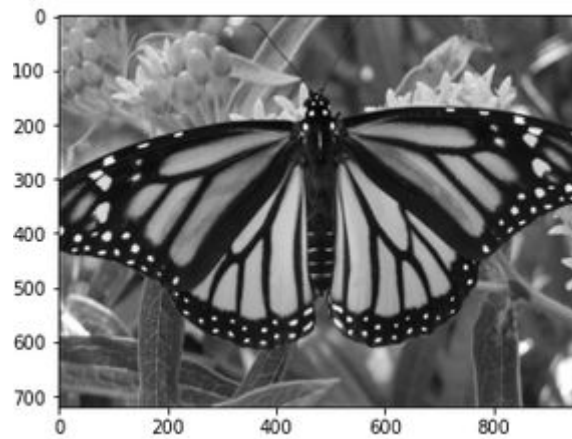
1.Linear:  
Precision = 0.9674028268551237 Recall = 0.880659429030961 Κριτήριο ποιότητας C = 0.9240311279430423  
2.Non Linear:  
Precision = 0.9828424869549837 Recall = 0.8936871733011661 Κριτήριο ποιότητας C = 0.9382648301280749

PSNR =10 sigma = 3 theta = 0.2

3.Linear:  
Precision = 0.7835597107073557 Recall = 0.7144350623240853 Κριτήριο ποιότητας C = 0.7489973865157205  
4.Non Linear:  
Precision = 0.8493103448275862 Recall = 0.7922798552472858 Κριτήριο ποιότητας C = 0.820795100037436

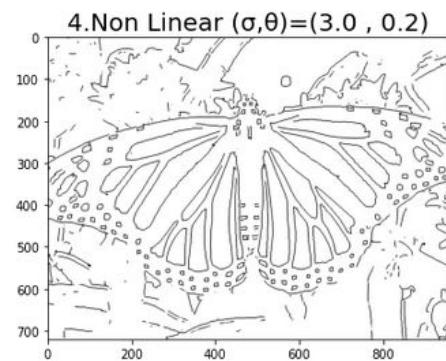
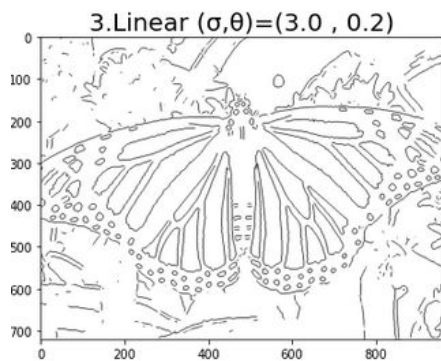
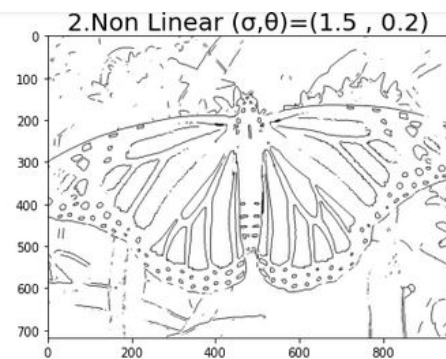
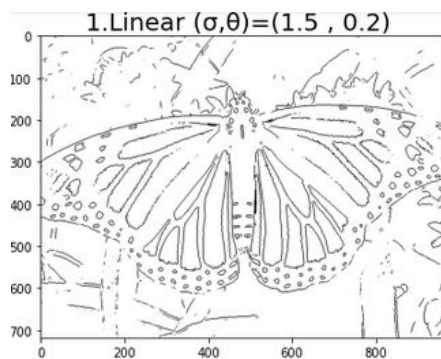
Εδώ επιβεβαιώνονται και οι αρχικές εκτιμήσεις για υπεροχή της μη γραμμικής προσέγγισης. Και στις δύο περιπτώσεις βέβαια ο αλγόριθμος μας πέτυχε C = 0.92 (περίπου) στην Ισ1, και C = 0.80 στην Ισ2 (Που ήταν και πιο θορυβώδης)

## 1.4. Εφαρμογή των Αλγορίθμων Ανίχνευσης Ακμών σε Πραγματικές εικόνες



**1.4.1** Στην συνέχεια πειραματιστήκαμε με μια πραγματική εικόνα (butterfly). Τα αποτελέσματά μας για διάφορες τιμές  $\sigma, \theta$  φαίνονται και για τις δύο περιπτώσεις (Γραμμικά μη Γραμμικά)

Outputs:





**1.4.2** Να σημειωθεί αφού υπολογίσαμε (όπως και πριν) τις αληθινές ακμές κάναμε αξιολόγηση :



Αξιολόγηση:

butterFly sigma = 1.5 theta = 0.2

1.Linear:

Precision = 0.9643063659780265 Recall = 0.40508788976238846 Κριτήριο ποιότητας C = 0.6846971278702075

2.Non Linear:

Precision = 0.9628344370860927 Recall = 0.40132276300680153 Κριτήριο ποιότητας C = 0.6820786000464472

PutterFly sigma = 3 theta = 0.2

3.Linear:

Precision = 0.7636846151938292 Recall = 0.33997659217383625 Κριτήριο ποιότητας C = 0.5518306036838327

4.Non Linear:

Precision = 0.7871070242164556 Recall = 0.3610325942937903 Κριτήριο ποιότητας C = 0.574069809255123

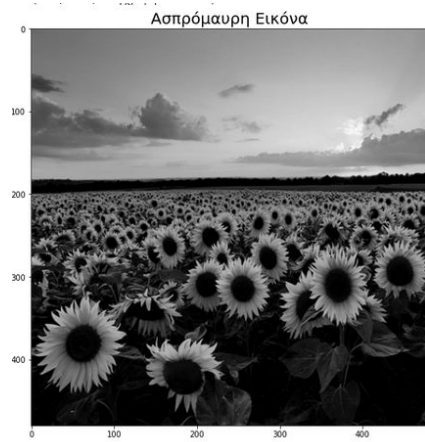
Γίνεται αντιληπτό ότι στην πραγματική εικόνα της πεταλουδας το C κυμαίνεται στο 0.6, γεγονός που εξηγείται αν παρατηρήσει κανείς την πολυπλοκότητα σε ακμές του background. Μετά από πειραματισμό με τις τιμές εξάγαμε τα εξής συμπεράσματα :

- ❖ Όσο αυξάνεται το  $\sigma$ , το φίλτρο της αρχικής εικόνας γίνεται πίο ομαλό και η εικόνα θολώνεται περισσότερο καθιστώντας δύσκολη την εύρεση των ακμών
- ❖ Όσο αυξάνεται το  $\theta$ , τίθεται ένα μεγαλύτερο φράγμα στα zero-crossings με αποτέλεσμα να μην επιλέγονται σημεία ακμών



## Μέρος 2: Ανίχνευση Σημείων Ενδιαφέροντος (Interest Point Detection)

Αρχικά λοιπόν διαβάσαμε τις εικόνες **sunflowers.png** και **caravaggio.jpg**, τις κανονικοποιήσαμε στο διάστημα  $[0,1]$  κρατώντας και τις έγχρωμες και τις grayscale εικόνες



### 2.1. Ανίχνευση Γωνιών

#### Μεθοδος Harris-Stephens

**2.1.1** Προτού υπολογιστούν τα στοιχεία **J1**, **J2** και **J3** του δομικού τανυστή **J**, εξομαλύνουμε τις εικόνες όπως και στο 1ο Μέρος

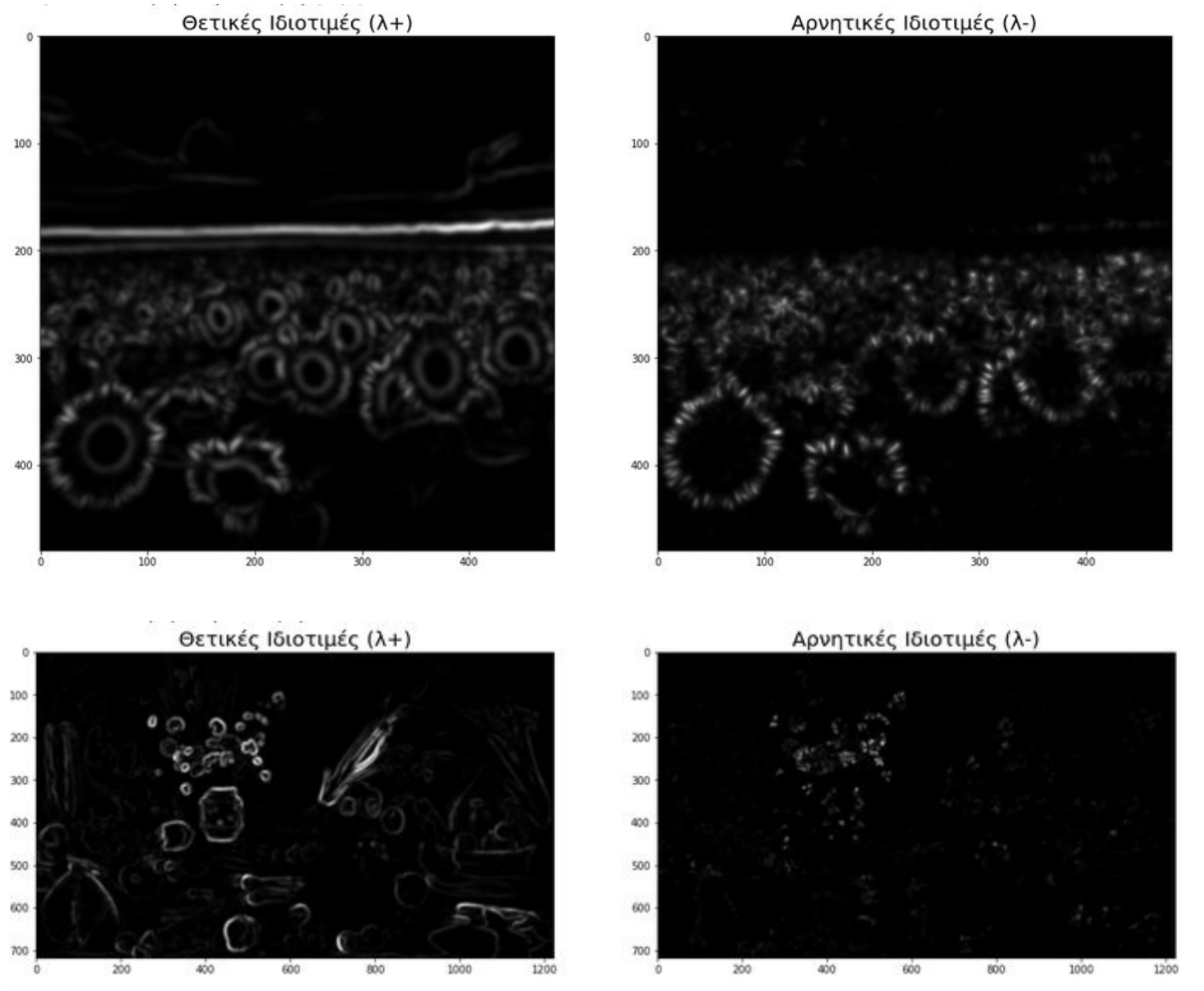
Εξομάλυνση Εικόνας



Εξομάλυνση Εικόνας



**2.1.2** Αφού υπολογίσαμε τις ιδιοτιμές  $\lambda^-$ ,  $\lambda^+$  του τανυστή  $J$ , τις σχεδιάσαμε σαν γκρι εικόνες:



Οι ιδιοτιμές αντιπροσωπεύουν τις μεταβολές της φωτεινότητας της εικόνας τόσο κάθετα όσο και οριζόντια. Πιο ειδικά η  $\lambda^+$  αντιστοιχεί στην μέγιστη μεταβολής της συχνότητας στο σημείο αυτό, ενώ η  $\lambda^-$  στην ελάχιστη. Στις εικόνες δεξιά (αρνητικές ιδιοτιμές) παρατηρείται μεγάλη ένταση στις γωνίες ενώ στις εικόνες αριστερά (θετικές ιδιοτιμές) παρατηρείται μεγάλη ένταση στις ακμές.

**2.1.3** Με βάση τις ιδιοτιμές αυτές εξήχθη το “κριτήριο γωνιότητας”  $R$ . Έτσι επιλέξαμε σαν γωνίες τα pixels  $(x, y)$  τα οποία:

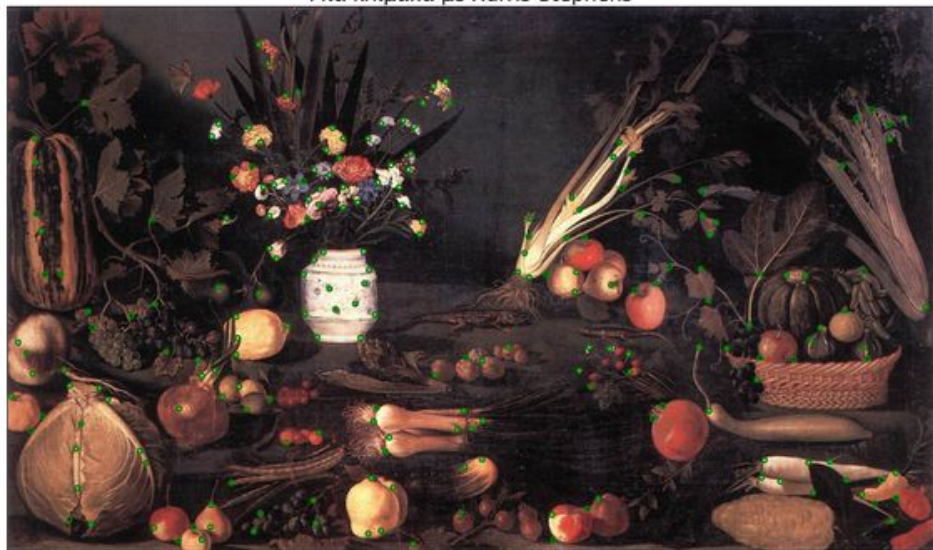
(Σ1) Είναι **μέγιστα** του  $R$  εντός τετραγωνικών παραθύρων που τα περιβάλλουν το μέγεθος των οποίων εξαρτάται από την κλίμακα  $\sigma$ ,

(Σ2) Αντιστοιχούν σε τιμή του  $R$  μεγαλύτερη από ένα ποσοστό του ολικού μεγίστου του  $R$ , δηλαδή  $R(x, y) > \theta_{corn} \cdot R_{max}$ , όπου  $\theta_{corn}$  ένα κατάλληλα επιλεγμένο κατώφλι.

Μια κλίμακα με Harris-Stephens



Μια κλίμακα με Harris-Stephens



**Σχόλιο:** Παρατηρήσαμε ότι με την αύξηση  $\sigma$  ή του  $\rho$ , η εικόνα θαμπώνει περισσότερο με αποτέλεσμα να **μην** εντοπίζονται μικρές γωνίες. Από την άλλη η παράμετρος  $k$  επηρέαζε τον αριθμό των σημείων που εντοπίζονται. Τέλος το κατώφλι  $\theta_{corn}$  επηρεάζει τον αριθμό των γωνιών που εντοπίζονται.



## 2.2. Πολυκλιμακωτή Ανίχνευση Γωνιών

### Μεθοδος Harris-Laplacian

**2.2.1** Στο πρώτο στάδιο εφαρμόσαμε τον αλγόριθμο εύρεσης γωνιών μονής κλίμακας για διαφορετικές κλίμακες ολοκλήρωσης και διαφόρισης.

**2.2.2** Στο δεύτερο στάδιο, αφού υπολογίσαμε την κανονικοποιημένη LoG για τις διάφορες κλίμακες του, απορρίψαμε τα σημεία για τα οποία η κλίμακα που ανιχνεύθηκαν δεν μεγιστοποιεί την LoG μετρική σε μια γειτονιά 2 διαδοχικών κλιμάκων.

Παρακάτω φαίνεται το αποτέλεσμα εκτέλεσης του αλγορίθμου για 4 κλίμακες:

Πολυκλιμακωτή με Harris-Laplacian



Πολυκλιμακωτή με Harris-Laplacian



**Σχόλιο:** Με την πολυκλιμακωτή ανάλυση μπορούμε να εντοπίσουμε μεγαλύτερες γωνίες(βλ Ουρανός), αφού δουλεύουμε για παραπάνω από μια κλίμακες και έτσι βρίσκουμε περισσότερες γωνίες.

### 2.3. Ανίχνευση Blobs

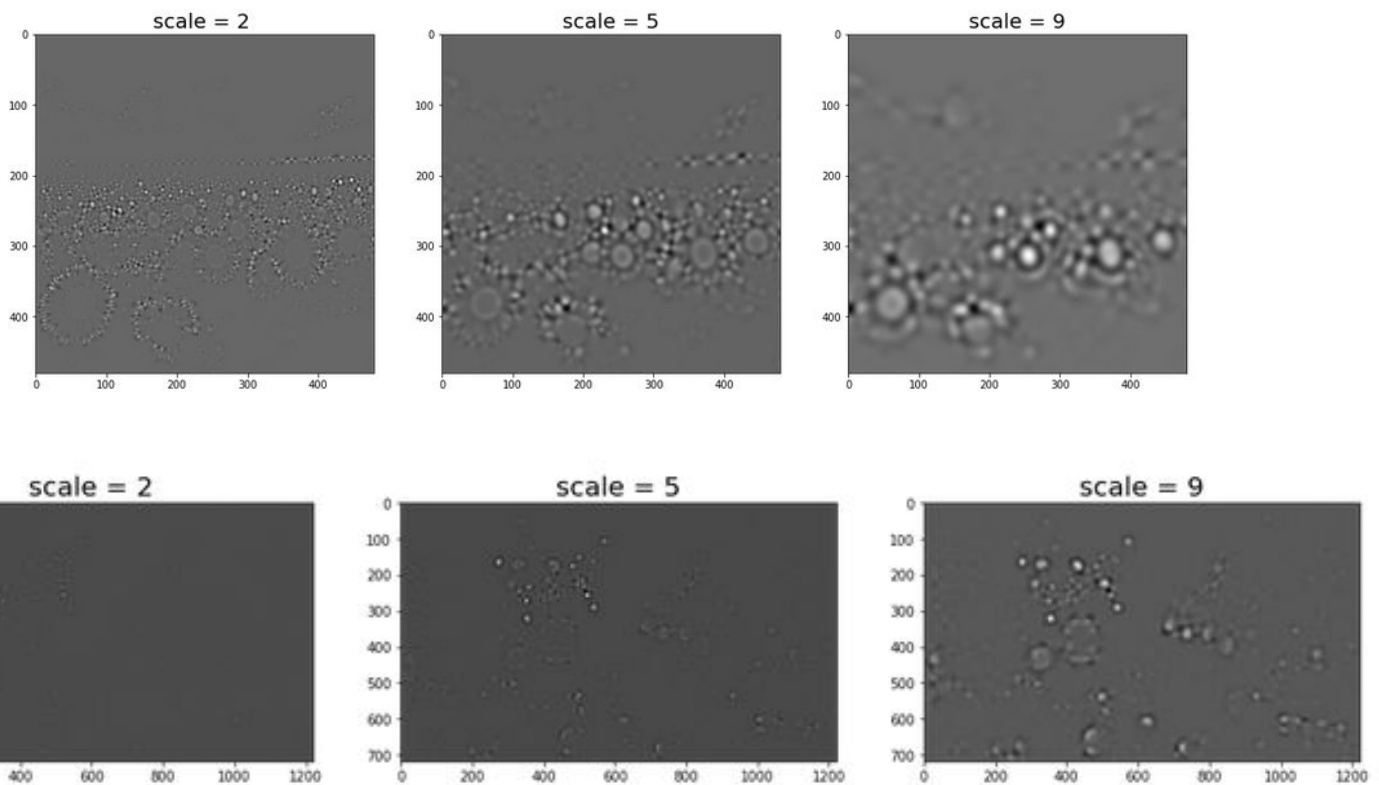
#### Hessian Matrix(Πίνακας Hessian)

Για την εύρεση σημείων ενδιαφέροντος με την μέθοδο ανίχνευσης blobs, χρησιμοποιούμε την ορίζουσα του πίνακα **Hessian**:

$$H(x, y) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

όπου τα **Lxx**, **Lxy**, **Lyy** είναι οι μερικές παράγωγοι δεύτερης τάξης της εικόνας.

**2.3.1** Υπολογίζουμε το κριτήριο **R** από την ορίζουσα του παραπάνω πίνακα και οπτικοποιούμε το αποτέλεσμα για διάφορα  $\sigma$ .



Τα αποτελέσματα αυτά θα σχολιαστούν στο ερώτημα 2.5.3 σε σχέση με αυτά των Box Filters.

**2.3.2** Όπως και στο ερώτημα **2.1.3** χρησιμοποιούμε κάποιες επιπλέον συνθήκες για την εύρεση των σημείων ενδιαφέροντος, με μόνη διαφορά ότι αντί για  $\theta_{\text{corn}}$  χρησιμοποιήσαμε ένα κατώφλι(threshold).

Blobs με μια κλίμακα



Blobs με μια κλίμακα



**Σχόλιο:** Με τη μέθοδο αυτή παρατηρεί κανείς ότι εντοπίσαμε αρκετές περιοχές με κάποια ομοιογένεια που διαφέρουν σημαντικά από την γειτονιά τους. Βέβαια κρίνεται αναγκαία μια πολυκλιμακωτή ανάλυση ώστε να εντοπίσουμε blobs και σε άλλες κλίμακες



## **2.4. Πολυκλιμακωτή Ανίχνευση Blobs**

**2.4.1** Επαναλαμβάνουμε την διαδικασία πολυκλιμακωτής ανίχνευσης γωνιών(ερώτημα **2.2**) και παρακάτω παρουσιάζεται το αποτέλεσμα για 4 κλίμακες:

Blobs με πολλές κλίμακες



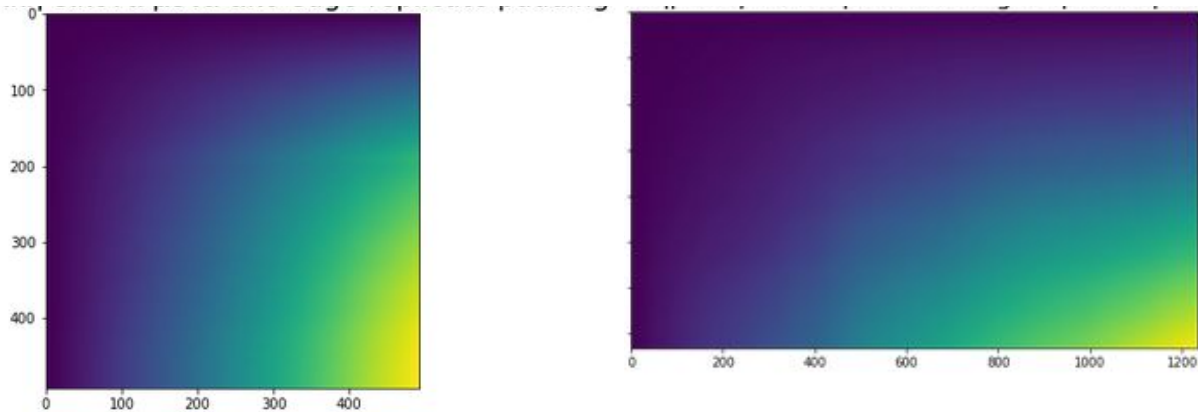
Blobs με πολλές κλίμακες



**Σχόλιο:** Με την πολυκλιμακωτή ανάλυση μπορούμε να εντοπίσουμε μεγαλύτερες περιοχές με ομοιογένεια (βλ Ουρανός), αφού δουλεύουμε για παραπάνω από μια κλίμακες

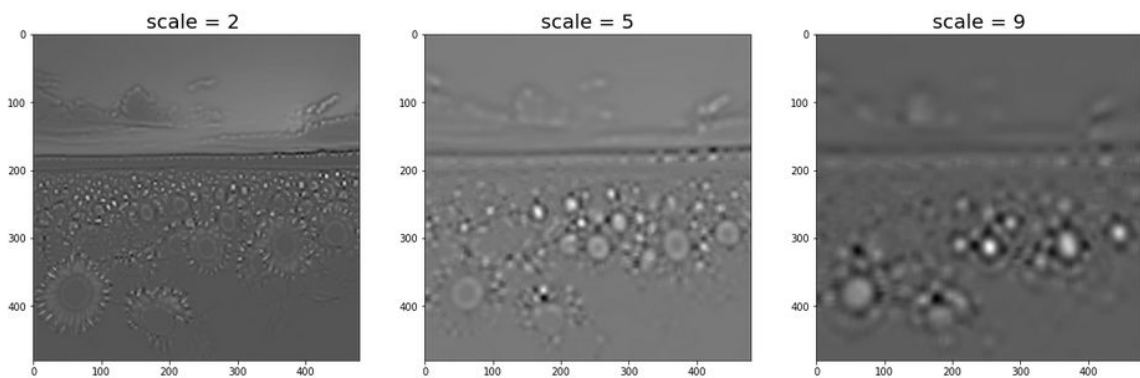
## 2.5. Επιτάχυνση με την χρήση Box Filters και Ολοκληρωτικών Εικόνων (Integral Images)

**2.5.1** Αρχικά υπολογίσαμε την Ολοκληρωτική Εικόνα  
(Αριστερά για sunflowers, δεξιά για caravaggio)

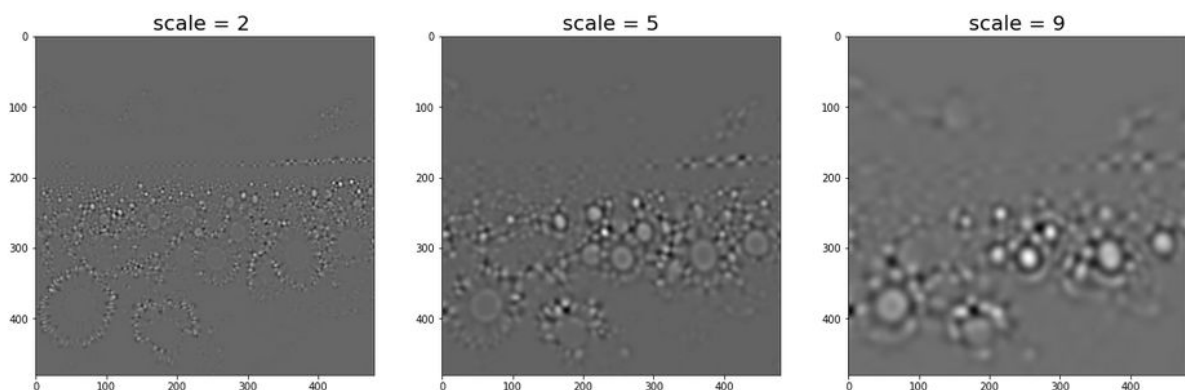


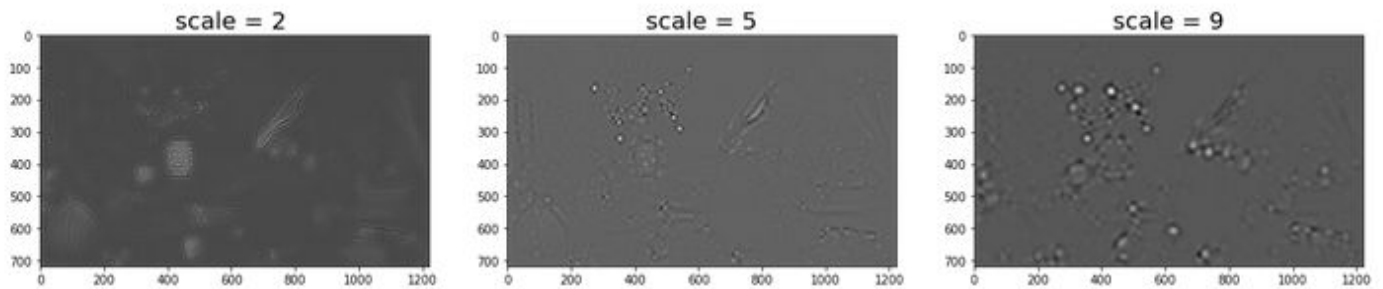
**2.5.2** Έπειτα υπολογίσαμε των  $L_{xx}$ ,  $L_{xy}$ ,  $L_{yy}$  με χρήση της Ολοκληρωτικής Εικόνας σύμφωνα με τα **Box Filters** ( $D_{xx}$ ,  $D_{xy}$ ,  $D_{yy}$ ).

**2.5.3** Στη συνέχεια βρήκαμε τα σημεία ενδιαφέροντος ως τα τοπικά μέγιστα του κριτηρίου  $R(x, y)$  με τον ίδιο τρόπο με τα προηγούμενα ερωτήματα. Το κριτήριο αυτό οπτικοποιήθηκε παρακάτω σε αντιπαραβολή με το αντίστοιχο της απλής Hessian

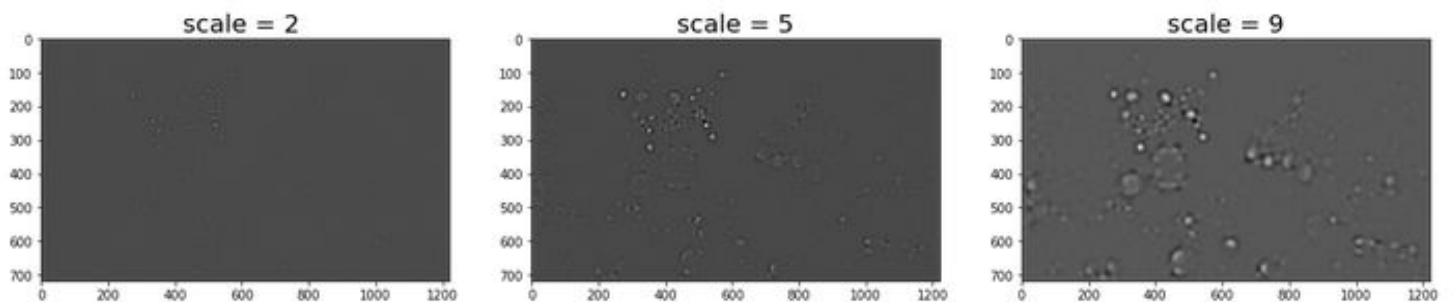


### Hessian





### Hessian

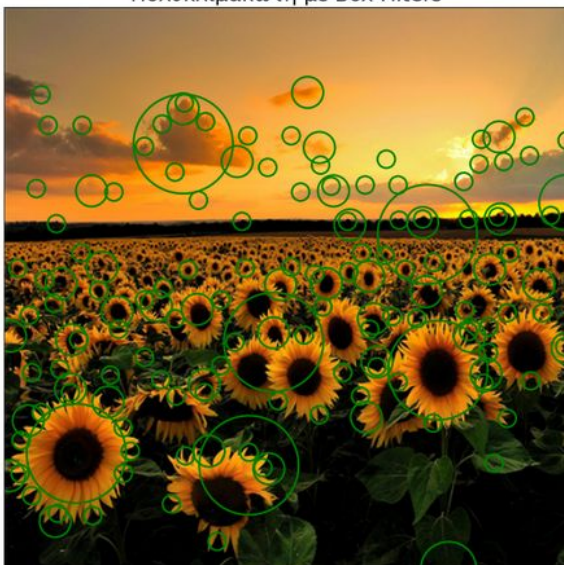


Παρατηρούμε ότι η εφαρμογή των αλγορίθμων παράγει παρόμοιες εικόνες. Για μεγαλύτερες κλίμακες γίνεται πιο εμφανής η διαφορά των 2 μεθόδων με τα box filters να παράγουν πιο "θολό" κριτήριο R (Box Filter = Προσεγγιστική μέθοδος).

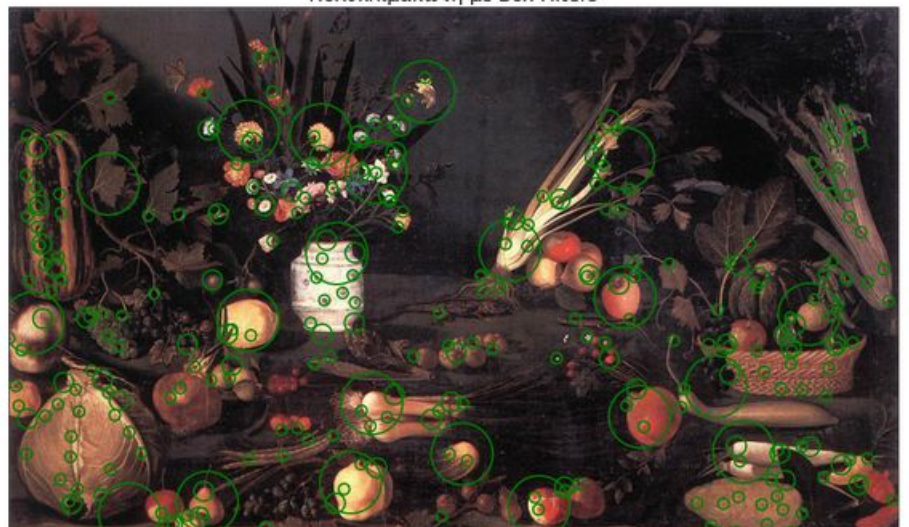
Επιπρόσθετα διαπιστώνουμε από τη σύγκριση των κριτηρίων που προκύπτουν, ότι εξαιτίας της ύπαρξης μεγαλύτερων ακραίων τιμών στο αποτέλεσμα των box filters οι εικόνες εμφανίζουν διαφορά σταθερής ποσότητας (με την εικόνα των box filters να εμφανίζεται πιο λευκή λόγω του μεγαλύτερου μέσου όρου).

**2.5.4** Επαναλαμβάνοντας την διαδικασία της πολυκλιμακωτής, πήραμε τα ακόλουθα αποτελέσματα:

Πολυκλιμακωτή με Box Filters



Πολυκλιμακωτή με Box Filters



**Σχόλιο:** Είναι προφανές ότι με την μέθοδο των Box Filters και Integral Images για τον υπολογισμό των μερικών παραγώγων δεύτερης τάξης της εικόνας, επιταχύνουμε την διαδικασία σε σχέση με την μέθοδο ανίχνευσης Blobs χωρίς χρήση Box Filters, αλλά σαν συνέπεια αυτής της επιτάχυνσης είχαμε μια πτώση στην ποιότητα των αποτελεσμάτων. Παρ' όλα αυτά, τα αποτελέσματα είναι ικανοποιητικά με την επιλογή κατάλληλων παραμέτρων.

Συνεπώς, αν θέλουμε τα καλύτερα αποτελέσματα, η μέθοδος ανίχνευσης Blobs χωρίς χρήση Box Filters είναι ακριβέστερη, διαφορετικά αν μας ενδιαφέρει και η ταχύτητα η χρήση Box Filters είναι μονόδρομος.

## **Τέλος Εργαστηριακής Αναφοράς**

### **Σ Η Μ Ε Ι Ω Σ Ε Ι Σ:**

→ Το Μέρος 1 και το Μέρος 2, σε ξεχωριστό .ipynb