

Πρωτότυπο Αναφοράς 1ης Εργαστηριακής Άσκησης  
Όραση Υπολογιστών Ακ. Έτος 2020-2021

Τσακανίκα Χριστίνα: 03117012  
Παυλάκη Παρασκευή-Ευγενία: 03117190

Μέρος 1: Ανίχνευση Ακμών σε Γκρίζες Εικόνες

Στο ερώτημα αυτό θα χρησιμοποιήσουμε αλγορίθμους εύρεσης ακμών, για να παράξουμε και να εκμεταλλευτούμε τις φυσικές ακμές της εικόνας για να δημιουργήσουμε "περιγράμματα".

Στόχος της ανίχνευσης ακμών είναι η άντληση των περιγραμμάτων των αντικειμένων της εικόνας, μειώνοντας έτσι την πληροφορία που χρειάζεται για την περιγραφή και τη μετέπειτα επεξεργασία αυτής.

Οπτικά, ακμές θεωρούμε τα σημεία της εικόνας στα οποία έχουμε απότομη μεταβολή της φωτεινότητας της εικόνας.

Οι απότομες μεταβολές αυτές οφείλονται σε:

- σκίαση
- αντανακλαστικότητα της επιφάνειας
- μεταβολές στις φυσικής ιδιότητες(περίγραμμα, υφή)

Ποιά είναι τα εμπόδια που συναντάμε στην εύρεση ακμών σε μία εικόνα;

- Μπορεί οι μεταβολές έντασης να μην είναι τόσο ισχυρές
- Οι εικόνες πολλές φορές περιέχουν θόρυβο και μας οδηγεί να εντοπίζουμε εσφαλμένες ακμές
- Οι εικόνες μπορεί να περιέχουν αντικείμενα σε πολλά διαφορετικά επίπεδα

Θα ανιχνεύσουμε τις ακμές με δύο τρόπους:

1η Παράγωγος:

Θα χρησιμοποιήσουμε διακριτούς τελεστές ώστε να υπολογίσουμε την πρώτη παράγωγο σε κάθε στοιχείο. Pixels με μεγάλη τιμή στο μέτρο της παραγώγου θα αντιστοιχούν σε ακμές. (Αυτό μας δημιουργεί πρόβλημα διότι ενισχύει το θόρυβο)

2η Παράγωγος:

Ομοίως, θα χρησιμοποιήσουμε διακριτούς τελεστές για τον υπολογισμό της Laplacian σε κάθε εικονοστοιχείο, αυτή τη φορά τα pixels με μηδενική τιμή θα αντιστοιχούν σε ακμές.

Βήμα 1ο:

Εξομάλυνση της εικόνας για μείωση θορύβου με ένα Gaussian φίλτρο.

Βήμα 2ο:

Laplacian τελεστής παραγώγου. με γραμμική η μορφολογική προσέγγιση.

Βήμα 3ο:

Σημεία εντοπισμού για μέγιστα ή ελάχιστα.

Βήμα 4ο:

Threshold

Γραμμικές μέθοδοι:

Συνέλιξη σε δύο διαστάσεις θα μας δώσει το φιλτράρισμα που θέλουμε.

Τελεστής Laplace:

Δύο διαδοχικά γραμμικά φιλτραρίσματα

1. Smoothing με Gaussian
2. Laplace operator

Επιπλέον, αναφέρεται πως είναι εφικτή η αντικατάσταση των δύο φιλτραρισμάτων σε ένα κάνοντας συνέλιξη τα δύο φιλτραρίσματα.

Αντί να κάνουμε smoothing και differation ξεχωριστά, δημιουργήσαμε ένα derivative of smoothing kernel το οποίο εφαρμόσαμε πάνω στην εικόνα μας.

Προσπαθούμε να εντοπίσουμε τα πιθανά σημεία για ακμές, ψάχνοντας τα zero crossing rates, δηλαδή τα σημεία μηδενισμού του τελεστή αυτού.

Μη γραμμικές μέθοδοι μέσω της θεωρίας συνόλων, μορφολογικά φίλτρα.

Ορισμός dilation και erosion

Dilation: επέκταση του χώρου ενός σχήματος δίσκου B

Erosion: Η τομή των στοιχείων ενός δίσκου B

Εφαρμόζοντας το Dilation και το Erosion διαδοχικά καταλήγουμε στα φίλτρα Opening και Closing

## 1.1 Δημιουργία Εικόνων Εισόδου

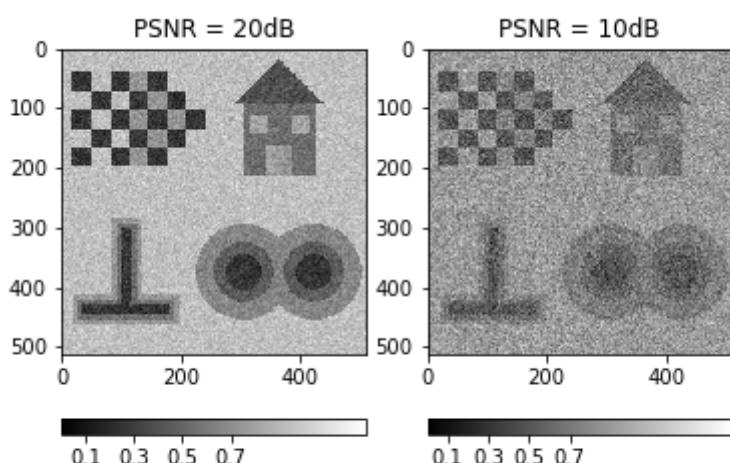
### **1.1.1**

Αρχικά διαβάσαμε την εικόνα “edgetest 21.png” η οποία μας δόθηκε στο πρόσθετο υλικό για το εργαστήριο, με χρήση της εντολής cv2.imread σε GRAYSCALE μορφή, και την κανονικοποιήσαμε στο διάστημα [0,1]

### **1.1.2**

Έπειτα προσθέσαμε λευκό gaussian θόρυβο στην εικόνα, με μηδενική μέση τιμή. Σκοπός μας είναι να χρησιμοποιήσουμε τις παρακάτω εικόνες σαν είσοδο στον αλγόριθμο ανίχνευσης ακμών. Εκτελέσαμε την διαδικασία δύο φορές, για PSNR(Peak-to-peak Signal to Noise Ratio) ίσο με

- i) PSNR = 20dB
- ii) PSNR = 10dB



Παραπάνω φαίνεται το επιθυμητό αποτέλεσμα και, όπως περιμέναμε, παρατηρούμε ότι για μεγαλύτερο PSNR έχουμε πιο καθαρή εικόνα, δηλαδή λιγότερο θόρυβο.

## 1.2 Γλοποίηση Αλγορίθμων Ανίχνευσης Ακμών

### **1.2.1**

Πρώτο μας βήμα είναι ο υπολογισμός της συνέλιξης των εικόνων με τη δισδιάστατη Gaussian.

Έναντι ξεχωριστής εφαρμογής smoothing και differation, δημιουργήσαμε ένα derivative of smoothing kernel το οποίο εφαρμόσαμε πάνω στην εικόνα μας.

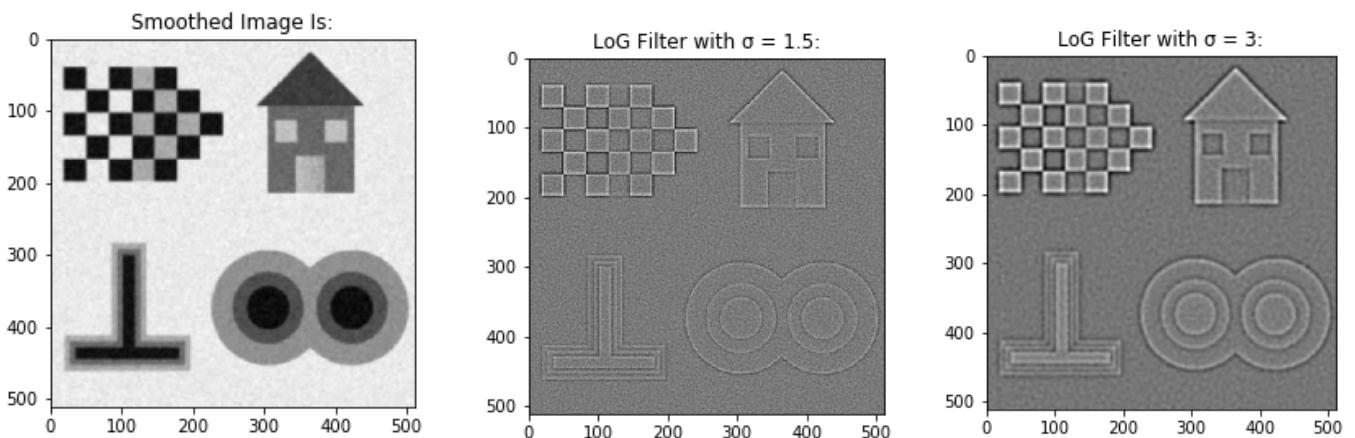
### 1.2.2

Στη συνέχεια προσεγγίσαμε την Laplacian L της εξομαλυμένης εικόνας, με δύο διαφορετικές εναλλακτικές

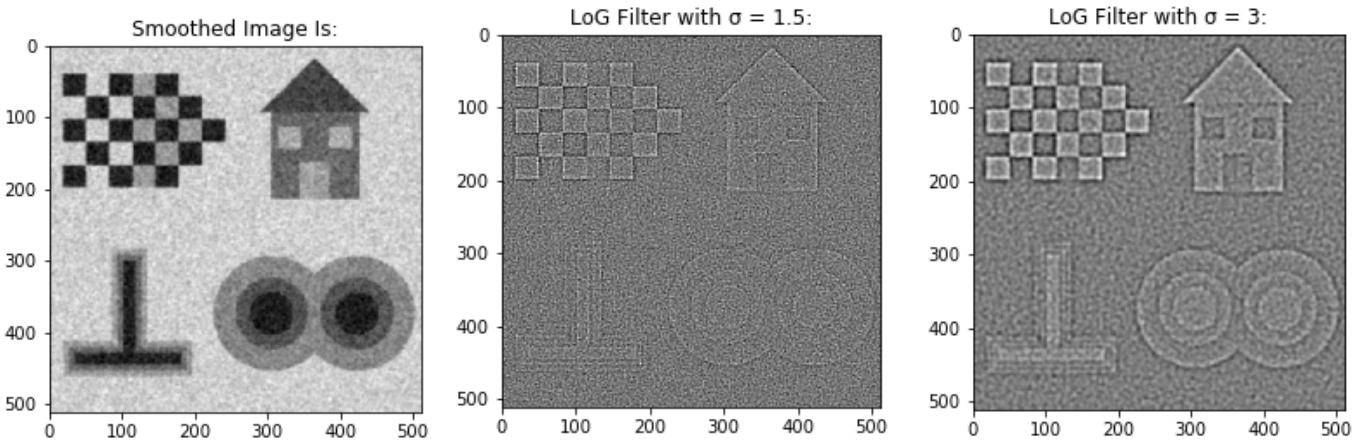
1. Γραμμική: Σε αυτή την εναλλακτική εφαρμόσαμε συνέλιξη της I με το LoG filter που φτιάχαμε παραπάνω, χρησιμοποιώντας την filter2D
2. Μη γραμμικά: όπου εκτιμήσαμε μη-γραμμικά την Laplacian παράγωγο με μορφολογικά φίλτρα, χρησιμοποιώντας τις συναρτήσεις dilate, erode, filter2D της cv2

Για τα γραμμικά φίλτρα έχουμε τις εξής εικόνες:

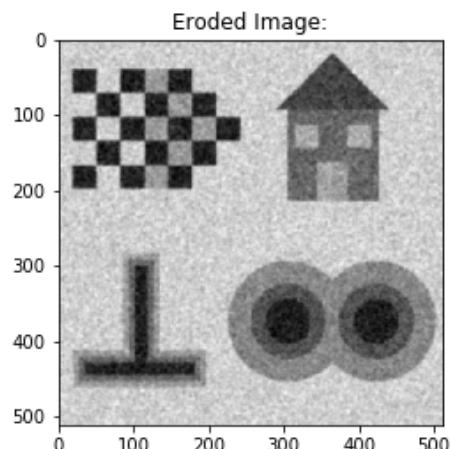
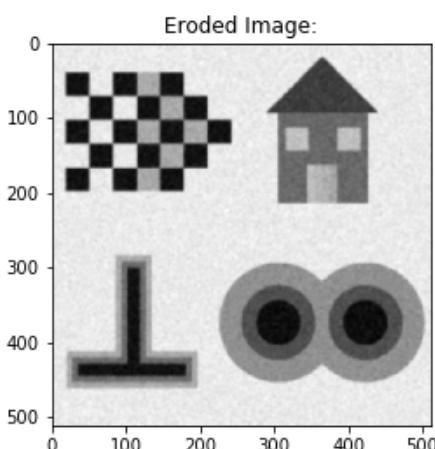
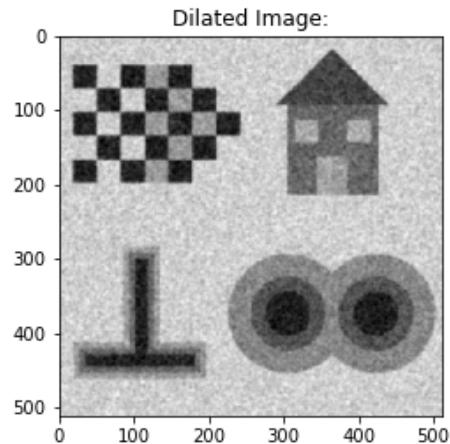
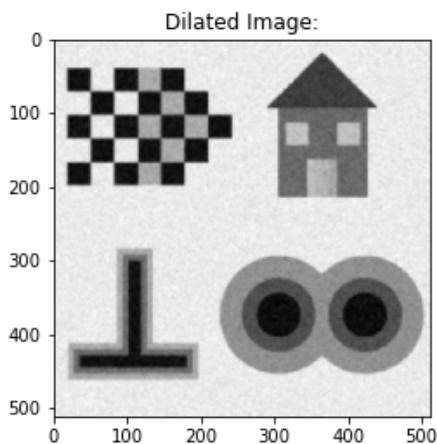
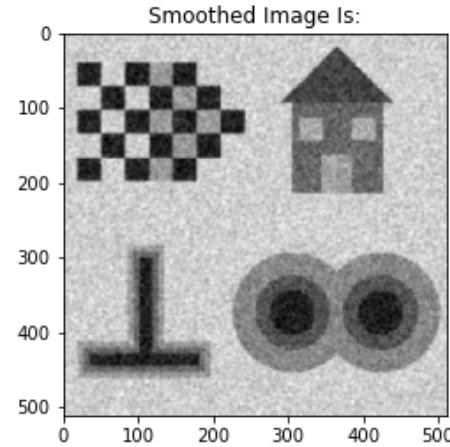
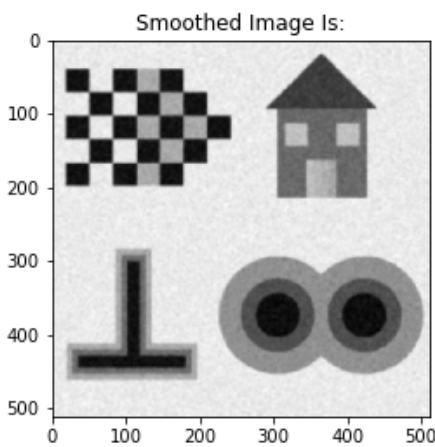
1. Για την πρώτη θορυβώδη εικόνα με PSNR = 20

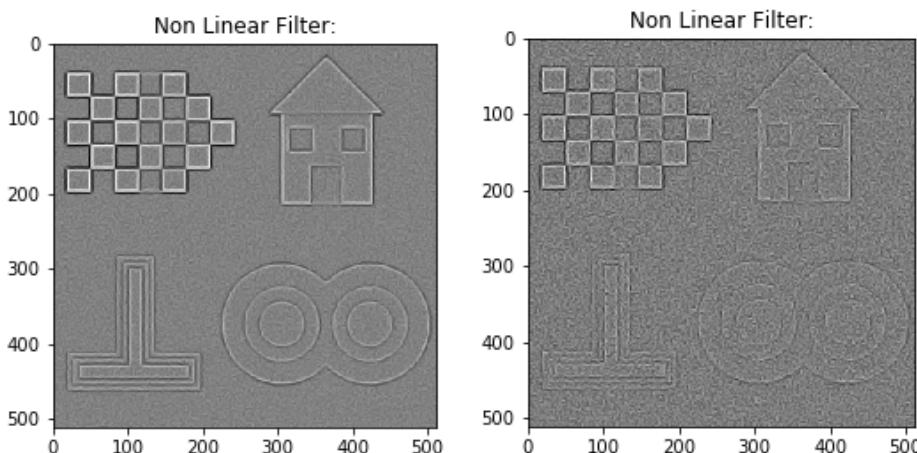


2. Για την δεύτερη θορυβώδη εικόνα με PSNR = 10



Για τα μη γραμμικά φίλτρα έχουμε τις εξής εικόνες, όπου αριστερά βλέπουμε την επεξεργασία εικόνας με PSNR = 20dB ενώ στα δεξιά PSNR = 10dB.



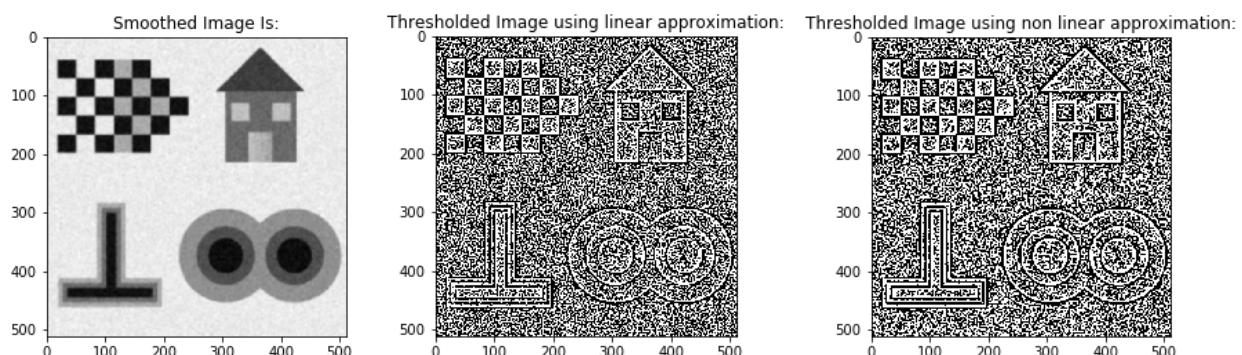


Ανάμεσα στα δύο διαφορετικά φίλτρα παρατηρούμε οτι στα μη-γραμμικά έχουμε όξυνση των ακμών και προδιάθεση για καλύτερα αποτελέσματα, πράγμα το οποίο για να πετύχουμε στο γραμμικό τρόπο έπρεπε να αυξήσουμε το  $\sigma$ .

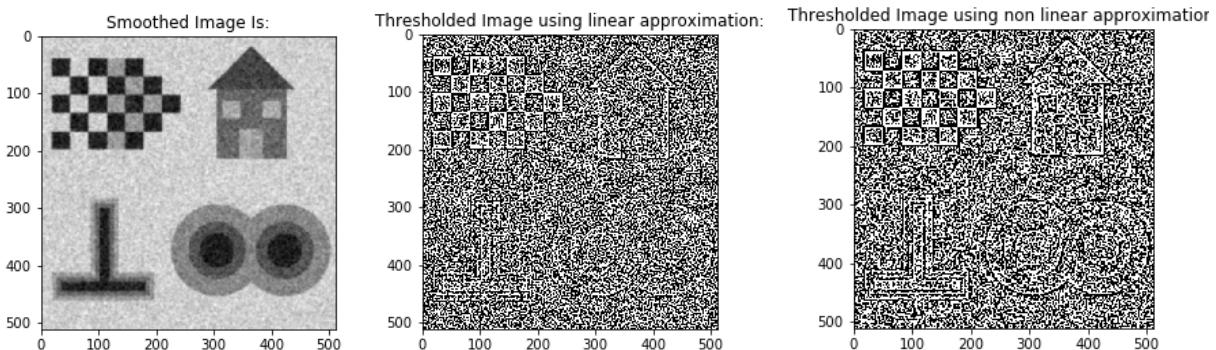
### 1.2.3

Εντοπίζουμε τα zero crossings και για τις δυο θορυβώδεις εικόνες, περνώντας το αποτέλεσμα από το προηγούμενο ερώτημα απο κατώφλι ίσο με 1:

#### 1. PSNR = 20dB



## 2. PSNR = 10dB

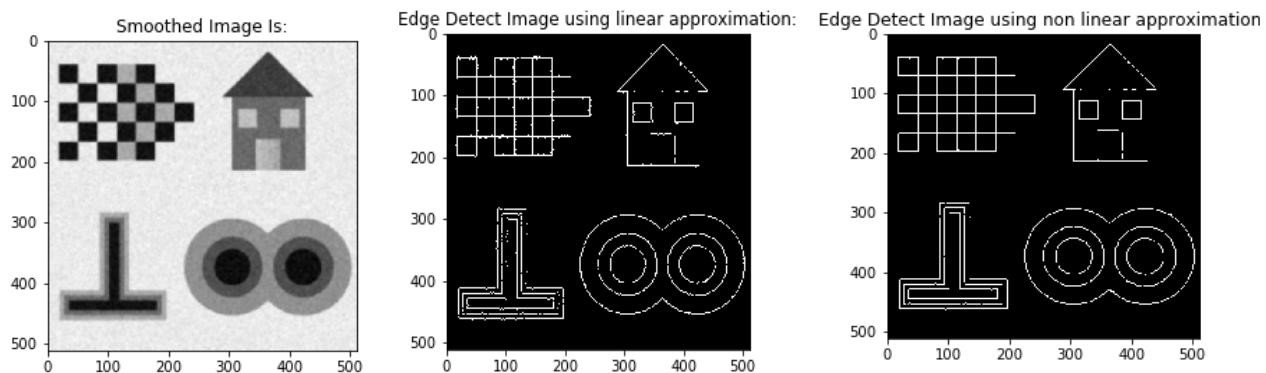


### 1.2.4

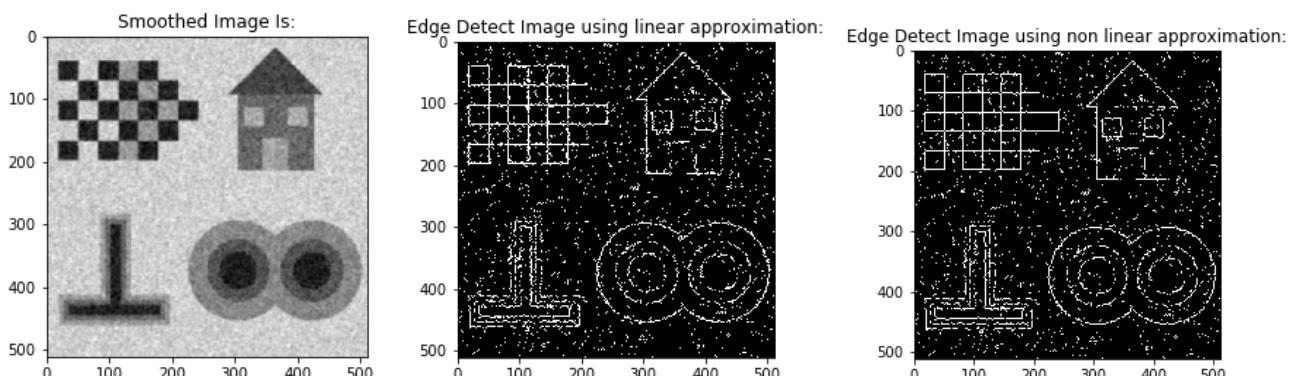
Τέλος απορρίπτουμε τα παραπάνω zero crossings σημεία σε σχετικά ομαλές περιοχές.

Τα αποτελέσματα για τις δύο θορυβώδεις εικόνες:

## 1. PSNR = 20dB



## 2. PSNR = 10dB



Όλα τα παραπάνω συμπεριλήφθηκαν και υλοποιήθηκαν βηματικά στη συνάρτηση EdgeDetect().

### 1.3 Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών

#### 1.3.1

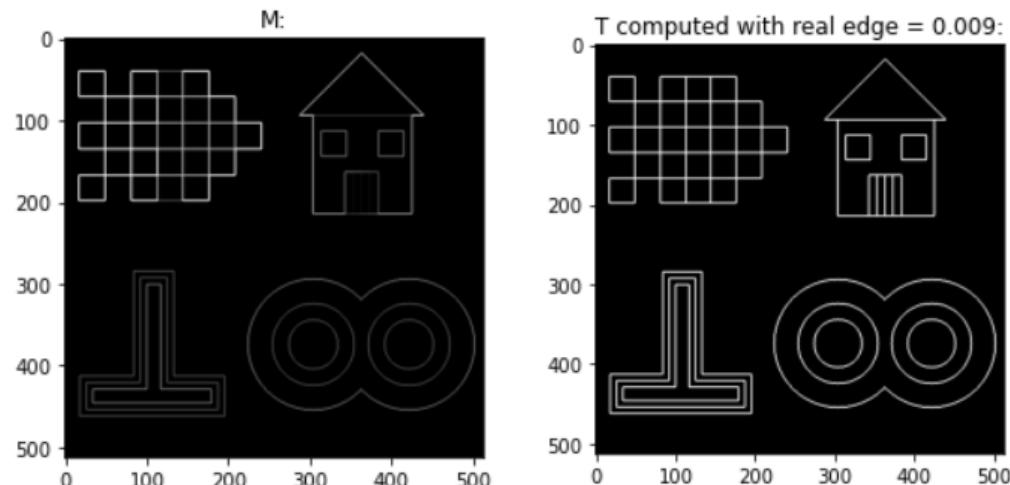
Υπολογίσαμε τις πραγματικές ακμές, χρησιμοποιώντας την αρχική καθαρή εικόνα. Η δυαδική εικόνα αληθινών ακμών T μπορεί να βρεθεί υπολογιστικά εφαρμόζοντας τον απλό τελεστή ακμών (edge operator) :

$$M = (Io \oplus B) - (Io \ominus B)$$

το οποίο προγραμματιστικά μεταφράζεται σε dilation - erosion.

Χρησιμοποιώντας για άλλη μία φορά την κατωφλίωση, επιλέγοντας αυτή τη φορά real\_edge = 0.009 με την εντολή \_T = cv2.threshold(M, 0.009, 1, cv2.THRESH\_BINARY) εξάγουμε τις δυαδικές εικόνες των πραγματικών ακμών από την αρχική εικόνα.

Συγκεκριμένα,



#### 1.3.2

Στη συνέχεια υλοποιήσαμε τη συνάρτηση ComputerPercision() για την εξαγωγή κριτηρίου ποιότητας σύμφωνα με τη σχέση:

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

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

Για τις παραμέτρους PSNR = 20db, s = 1.5, theta\_edge=0.2 και linear προσέγγιση προκύπτει ο πίνακας D από την Edge Detect, ενώ το κριτήριο ποιότητας δίνει το πολύ ικανοποιητικό αποτέλεσμα 0.9228982996832767.

### 1.3.3

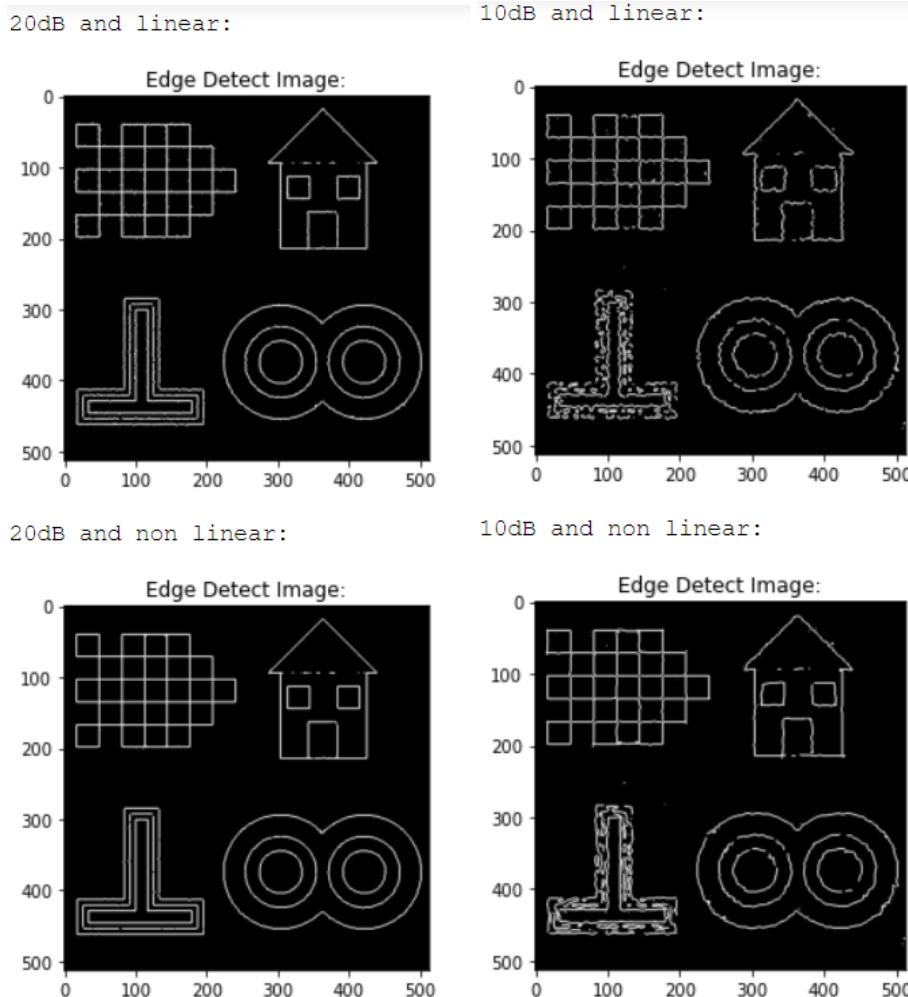
Στα παραπάνω, για κάθε διαφορετική είσοδο (διαφορετικό PSNR), πειραματιστήκαμε με τις τιμές των παραμέτρων σ και θ edge ώστε να πετύχουμε όσο το δυνατόν καλύτερα αποτελέσματα.

Εδώ επιβεβαιώνονται και οι αρχικές εκτιμήσεις για την αποτελεσματικότητα της μη γραμμικής προσέγγισης.

Στην εικόνα με PSNR = 20dB ο γραμμικός αλγόριθμος πέτυχε evaluation 0.92, ενώ ο μη γραμμικός 0.95. Πολύ πιο ευδιάκριτη όμως είναι η διαφορά ανάμεσα στους δύο τρόπους προσέγγισης όταν πάμε στην αξιολόγηση των δύο αλγορίθμων στην εικόνα με PSNR = 10dB, όπου ο μη-γραμμικός τρόπος προσέγγισης έφτασε στο 0.72 έναντι του γραμμικού ο οποίος έφτασε στο 0.60.

Αναλυτικά τα αποτελέσματα που προέκυψαν είναι τα εξής:

20dB and linear: 0.9228982996832767  
20dB and non linear: 0.9554066319591279  
10dB and linear: 0.6002867509008485  
10dB and non linear: 0.7291481113291536



### 1.33

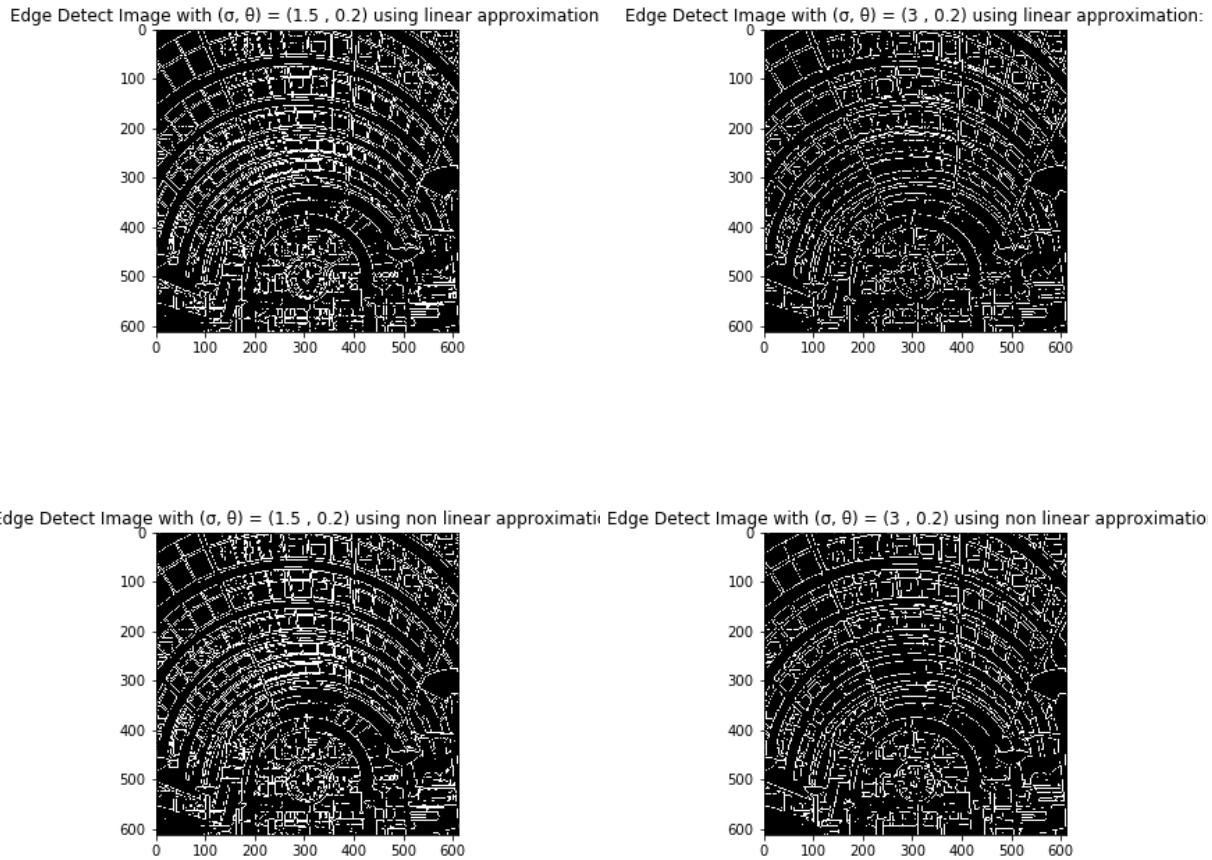
Πειραματίζομενοι με τις παραμέτρους  $\sigma$ , θedge, καταλήξαμε στις βέλτιστες τιμές παραμέτρων. Για κάθε περίπτωση έχουμε αναλυτικά:

- 20dB και γραμμική προσέγγιση:  $\sigma=1.7$ ,  $\text{theta\_edge}=0.2$ , ποιότητα: 0.9299 όπου παρατηρήθηκε ότι όσο μειώναμε το theta edge η ποιότητα μειωνόταν.
- 20dB και μη γραμμική προσέγγιση:  $\sigma=1.7$ ,  $\text{theta\_edge}=0.16$ , ποιότητα: 0.9586
- 10dB και γραμμική προσέγγιση:  $\sigma=2$ ,  $\text{theta\_edge}=0.3$ , ποιότητα: 0.6571
- 10dB και μη γραμμική προσέγγιση:  $\sigma=2$ ,  $\text{theta\_edge}=0.3$ , ποιότητα: 0.7186

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

#### 1.4.1

Στην συνέχεια πειραματιστήκαμε με μια πραγματική εικόνα. Τα αποτελέσματα και για τις δύο περιπτώσεις για διάφορες τιμές  $\sigma, \theta$  παρατίθενται παρακάτω:



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

Αρχικά για τις αναγκες αυτού του μέρους θα διαβάσαμε τις εικόνες `blood smear.jpg` και `mars.png`, πάνω στις οποίες θα πρέπει να σχεδιαστούν τα αποτελέσματα των ανιχνευτών που θα υλοποιήσουμε στη συνέχεια.

### 2.1. Ανίχνευση Γωνιών

Μια διαισθητικά απλή κατηγορία σημείων ενδιαφέροντος είναι η επιλογή σημείων που αντιστοιχούν σε γωνίες της εικόνας.

Για την ανίχνευση των γωνιών θα υλοποιήσουμε την κλασική μέθοδο των Harris-Stephens.

Συγκεκριμένα, υπολογίζουμε τις  $G_x$ ,  $G_y$ , οι οποίες αποτελούν τους δισδιάστατους Gaussian πυρήνες ομαλοποίησης που αφορούν τις κλίμακες διαφόρισης ( $\sigma$ ) και ολοκλήρωσης ( $\rho$ ) αντίστοιχα.

#### **2.1.1**

Έπειτα, φιλτράρουμε την εικόνα μας ( $G_x * I = I_x$ ), υπολογίζουμε τις μερικές της παραγώγους αντής, φιλτράρουμε με την  $G_y$  υπολογίζουμε τα  $J_1, J_2, J_3$  του δομικού τανυστή  $J$ , όπου

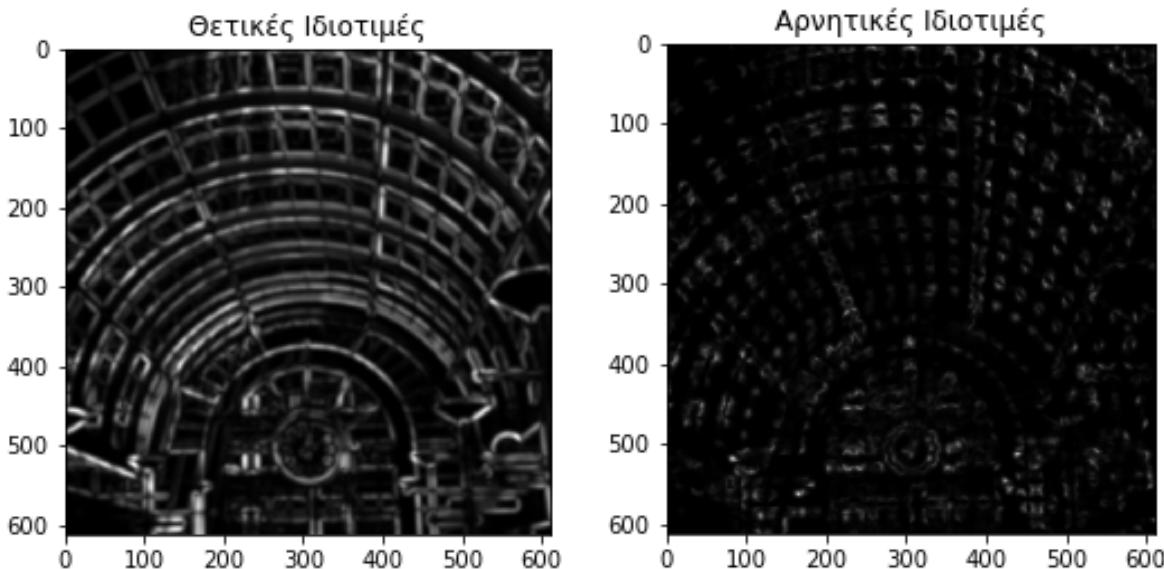
$$J_1(x, y) = G_\rho * \left( \frac{\partial I_\sigma}{\partial x} \cdot \frac{\partial I_\sigma}{\partial x} \right) (x, y)$$

$$J_2(x, y) = G_\rho * \left( \frac{\partial I_\sigma}{\partial x} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

$$J_3(x, y) = G_\rho * \left( \frac{\partial I_\sigma}{\partial y} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

### 2.1.2

Υπολογίζουμε τις ιδιοτιμές του παραπάνω τανυστή για κάθε πίξελ, από το δεδομένο τύπο. Οπτικοποιούμε τις θετικές και τις αρνητικές ιδιοτιμές και τα αποτελέσματα που λαμβάνουμε είναι τα ακόλουθα:



Οι ιδιοτιμές αντιπροσωπεύουν τις μεταβολές της φωτεινότητας της εικόνας τόσο κάθετα όσο και οριζόντια.

$\lambda+$  : αντιστοιχεί στην μέγιστη μεταβολής της συχνότητας στο σημείο αυτό, ενώ

$\lambda-$  : αντιστοιχεί στην ελάχιστη μεταβολής της συχνότητας στο σημείο αυτό.

Στην δεξιά εικόνα (αρνητικές ιδιοτιμές) διακρίνουμε μεγάλη ένταση στις γωνίες. Γι' αυτό παρατηρούμε μικρά άσπρα σημεία ενδιαφέροντος, ενώ στις εικόνες αριστερά (θετικές ιδιοτιμές) παρατηρούμε μεγάλη ένταση στις ακμές, καθώς οι περιοχές ενδιαφέροντος έχουν μεγαλύτερο μήκος και βρίσκονται πάνω στη "διαδρομή" των αναμενόμενων.

### 2.1.3

Με βάση τις ιδιοτιμές αυτές εξήχθη το "κριτήριο γωνιότητας" R. Έτσι, επιλέξαμε

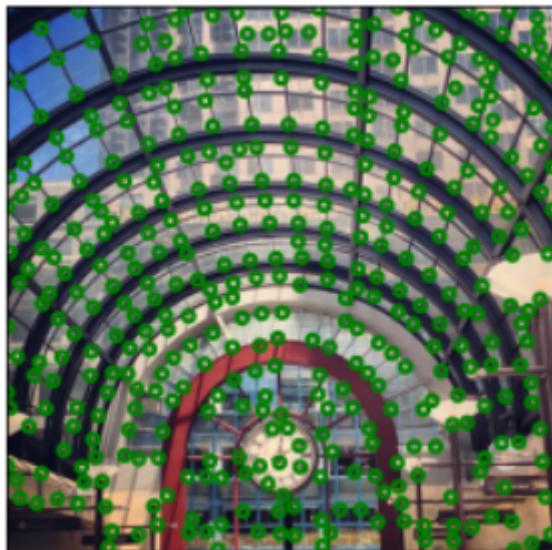
σαν γωνίες τα pixels (x, y) τα οποία:

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

(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. Ένα προτεινόμενο κατώφλι είναι το  $\theta_{corn} = 0.005$ .

Από την ανίχνευση γωνιών, τα αποτελέσματα που εξάγονται φαίνονται παρακάτω:

Εφαρμογή της μεθόδου Harris-Stephens με  $\sigma=2$ ,  $\rho=2.5$ ,  $\theta_{corn} = 0.005$ ,  $k = 0.05$



Με την αύξηση σ παρατηρούμε ότι η εικόνα θα μπώνει περισσότερο με αποτέλεσμα να μην εντοπίζονται μικρές γωνίες. Η παράμετρος k επηρεάζει τον αριθμό των σημείων που εντοπίζονται. Τέλος, το κατώφλι  $\theta_{corn}$  επηρεάζει τον αριθμό των γωνιών που εντοπίζονται.

## 2.2. Πολυκλιμακωτή Ανίχνευση Γωνιών

### **2.2.1**

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

$$\begin{aligned}\sigma_0, \sigma_1, \dots, \sigma_{N-1} &= s^0 \sigma_0, s^1 \sigma_0, \dots, s^{N-1} \sigma_0 \\ \rho_0, \rho_1, \dots, \rho_{N-1} &= s^0 \rho_0, s^1 \rho_0, \dots, s^{N-1} \rho_0\end{aligned}$$

όπου  $\sigma_0, \rho_0$  οι αρχικές κλίμακες διαφόρισης και ολοκλήρωσης, s ο παράγοντας κλιμάκωσης και N ο αριθμός των κλιμάκων.

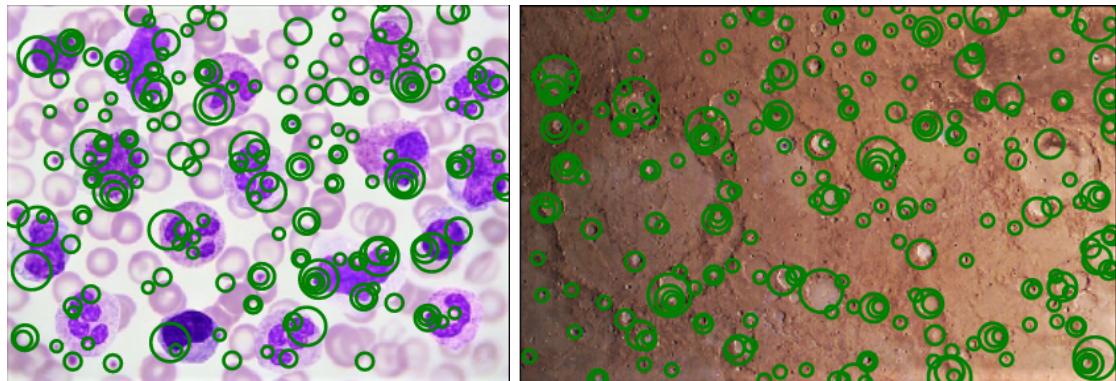
### 2.2.2

Στο δεύτερο στάδιο, αφού υπολογίσαμε την κανονικοποιημένη LoG

$$|\text{LoG}(x, \sigma_i)| = \sigma_i^2 |L_{xx}(x, \sigma_i) + L_{yy}(x, \sigma_i)|$$

απορρίψαμε τα σημεία για τα οποία η κλίμακα που ανιχνεύθηκαν δεν μεγιστοποιεί την LoG μετρική σε μια γειτονιά 2 διαδοχικών κλιμάκων.

Με τη βοήθεια της interest\_points\_visualization παρουσιάσαμε για τις δύο εικόνες το αποτέλεσμα που είχαν με τη χρήση του HarrisLaplacian.



### 2.3. Ανίχνευση Blobs

Μια από τις σημαντικότερες κατηγορίες σημείων ενδιαφέροντος βασίζονται στην ανίχνευση “blobs”, που ορίζονται ως περιοχές με κάποια ομοιογένεια που διαφέρουν σημαντικά από την γειτονιά τους.

Για την εύρεση τέτοιων περιοχών, σε αντιστοιχία με το κριτήριο γωνιότητας της μεθόδου Harris, γίνεται χρήση των μερικών παραγώγων δεύτερης τάξης της εικόνας και συγκεκριμένα η ορίζουσα του πίνακα Hessian:

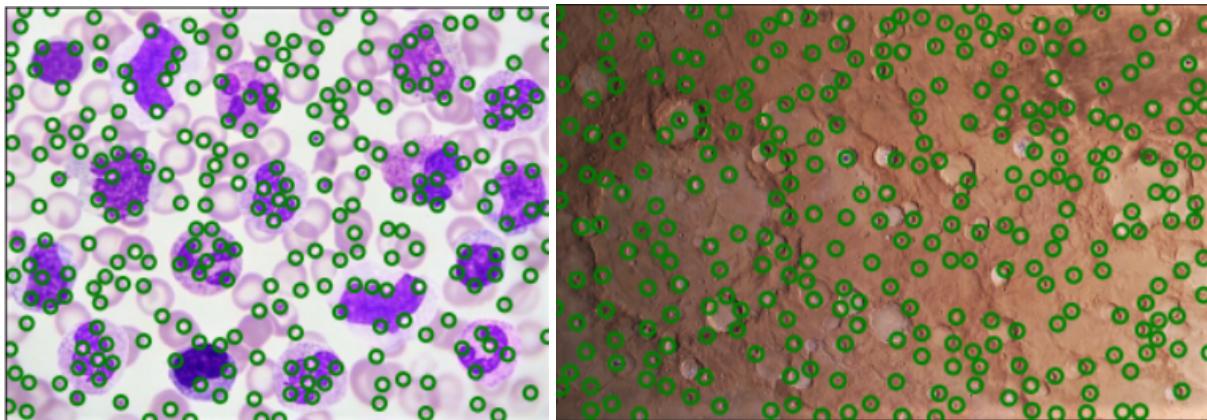
$$H(x, y) = \begin{bmatrix} J_{xx}(x, y, \sigma) & J_{xy}(x, y, \sigma) \\ J_{xy}(x, y, \sigma) & J_{yy}(x, y, \sigma) \end{bmatrix}$$

### 2.3.1

Υπολογίσαμε τις μερικές παραγώγους δεύτερης τάξης της εικόνας  $L_{xx}$ ,  $L_{xy}$ ,  $L_{yy}$  επιλέγοντας μια τιμή για την κλίμακα  $\sigma$  και κατασκευάσαμε το κριτήριο  $R(x, y)$  το οποίο προκύπτει από την ορίζουσα του παραπάνω πίνακα για κάθε pixel.

### 2.3.2

Όπως και στο ερώτημα 2.1.3, έτσι και εδώ χρησιμοποιήσαμε κάποιες επιπλέον συνθήκες για την εύρεση των σημείων ενδιαφέροντος.

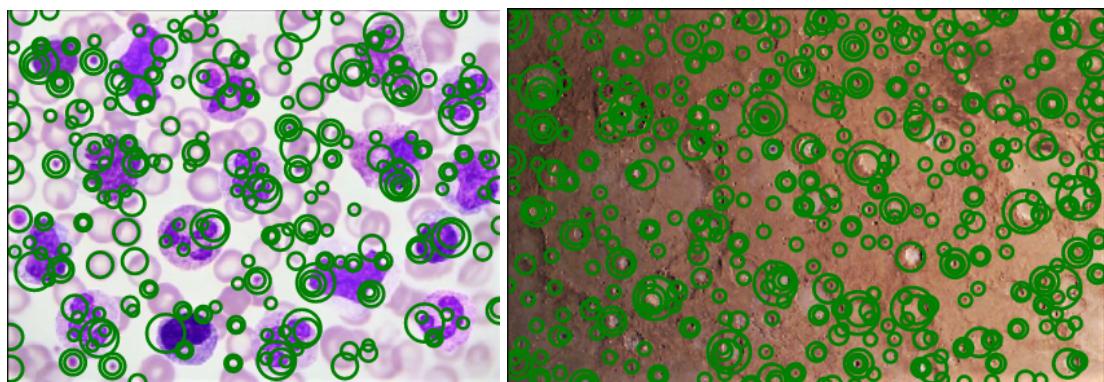


Πράγματι, ο αλγόριθμος έχει εστιάσει περισσότερο σε σημεία της εικόνας τα οποία παρουσιάζουν μεγάλη διαφορά από το περιβάλλον τους.

### 2.4. Πολυκλιμακωτή Ανίχνευση Blobs

#### 2.4.1

Επαναλαμβάνουμε την διαδικασία της πολυκλιμακωτής ανίχνευσης γωνιών, προσθέτοντας ένα δεύτερο στάδιο για την επιλογή σημείων ενδιαφέροντος που παρουσιάζουν μέγιστο σε γειτονικές κλίμακες (Hessian-Laplace).



Αυτό που παρατηρούμε ανάμεσα στο 2.4 και 2.3 είναι ότι με την πολυκλιμακωτή ανίχνευση εντοπίζουμε πολύ περισσότερες περιοχές ενδιαφέροντος όπου παρουσιάζεται ανομοιογένεια, συνέπεια του ότι δουλέψαμε σε περισσότερες από μια κλίμακες.

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

### 2.5.1

Υπολογίζουμε την ολοκληρωτική εικόνα, η οποία προκύπτει από την αρχική, όταν στη τιμή του κάθε πίξελ προσθέσουμε τιμές των πίξελ που βρίσκονται από αριστερά, από πάνω του και διαγώνιά του.

### 2.5.2

Έχοντας πλέον σχηματίσει την ολοκληρωτική εικόνα, επιθυμούμε να εφαρμόσουμε σε αυτην κάθε ένα από τα φίλτρα D<sub>xx</sub>, D<sub>yy</sub>, D<sub>xy</sub>. Τα φίλτρα αυτά πρέπει να “κεντραριστούν” σε κάθε πίξελ της εικόνας. Για το λόγο αυτό εφαρμόζουμε σε κάθε περίπτωση φίλτρου zero padding στην ολοκληρωτική εικόνα, προσθέτοντάς της (ύψος φίλτρου -1)/2 γραμμές πάνω και κάτω, και (πλάτος φίλτρου -1)/2 στήλες δεξιά και αριστερά της εικόνας. Η μέθοδος που επιλέχθηκε και για να εφαρμόσουμε τα φίλτρα στην εικόνα είναι οι 2 for loops. Επομένως, για κάθε στοιχείο της αρχικής εικόνας, προσδιορίζουμε τις συντεταγμένες των γωνιών των φίλτρων (συμβολίζονται με A<sub>1,B<sub>1,C<sub>1,D<sub>1 για το πρώτο παραλληλόγραμμο, A<sub>2,B<sub>2,C<sub>2,D<sub>2 για το δεύτερο και A<sub>3,B<sub>3,C<sub>3,D<sub>3 για το τρίτο, ενώ χρησιμοποιούμε τα A<sub>4,B<sub>4,C<sub>4,D<sub>4 για το τρίτο φίλτρο) στην zero padded εικόνα.</sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub></sub>

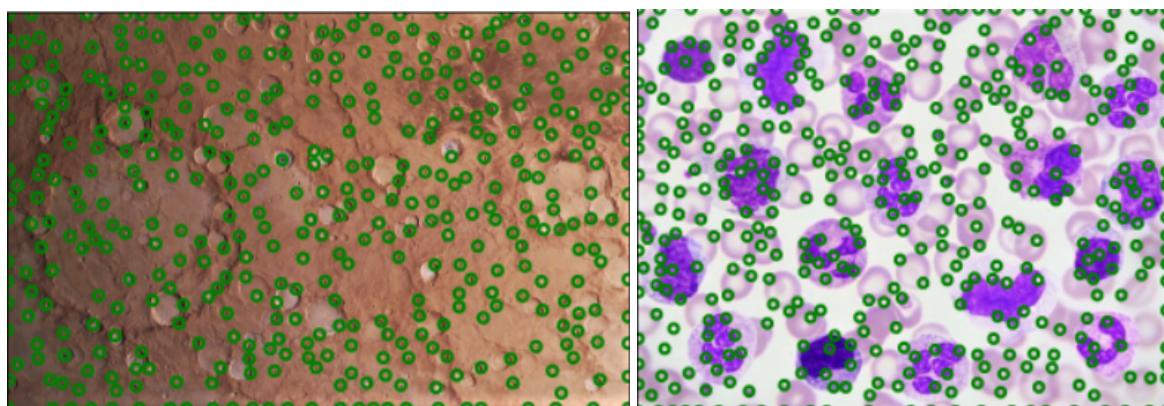
Έπειτα, πολλαπλασιάζοντας την παράσταση S<sub>A</sub> + S<sub>C</sub> - S<sub>B</sub> - S<sub>D</sub> με τον συντελεστή του παραλληλογράμμου (είτε 1, είτε -2, είτε -1) προκύπτει η φιλτραρισμένη τιμή για το συγκεκριμένο πίξελ.

Σκοπός της παραπάνω διαδικασίας είναι η αποφυγή της συνέλιξης της εικόνας με κάθε φίλτρο, για τις τέσσερις κλίμακες και άρα η εξοικονόμηση χρόνου.

### 2.5.3

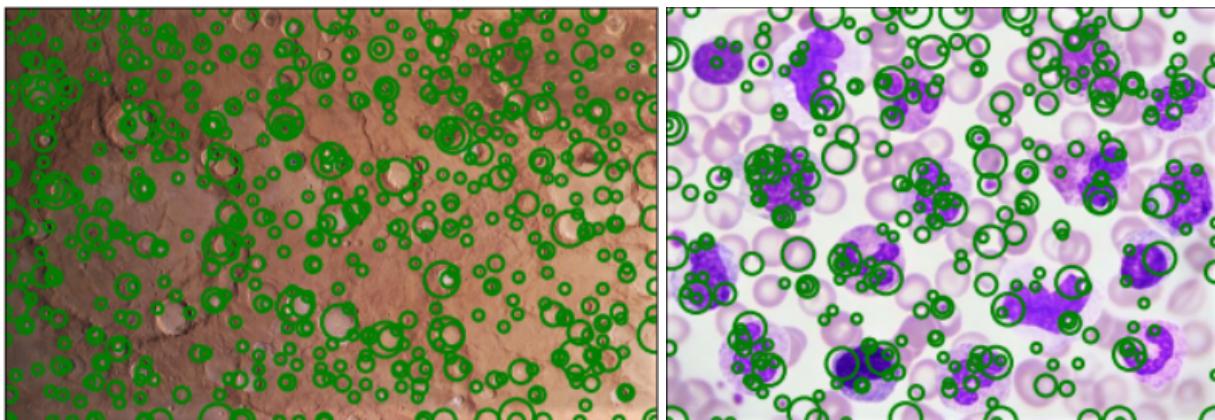
Το κριτήριο R σε αυτό το ερώτημα προσδιορίζεται ως:

$$R(x, y) = L_{xx}(x, y)L_{yy}(x, y) - 0.81L_{xy}^2(x, y)$$



## 2.5.4

Το αποτέλεσμα της πολυκλιμακωτής ανίχνευσης σημείων ενδιαφέροντος με τη μέθοδο των ολοκληρωτικών εικόνων και τών box filters παρατίθεται παρακάτω:



### Μέρος 3: Εφαρμογές σε Ταίριασμα και Κατηγοριοποίηση Εικόνων με Χρήση Τοπικών Περιγραφητών στα Σημεία Ενδιαφέροντος

Τα σημεία ενδιαφέροντος δίνουν μια εκτίμηση περιοχών, οι οποίες περιέχουν σημαντικά χαρακτηριστικά της εικόνας. Από τις περιοχές αυτές εξάγουμε τοπικούς περιγραφητές (local descriptors), που κωδικοποιούν μια γειτονιά γύρω από τα σημεία ενδιαφέροντος (η ακτίνα της οποίας εξαρτάται από την κλίμακα).

Ως τοπικούς περιγραφητές θα χρησιμοποιήσουμε τους εξής:

- SURF (Speed Up Robust Features)

Για τον εντοπισμό σημείων ενδιαφέροντος, ο SURF χρησιμοποιεί μια ακέραια προσέγγιση της παραγώγου του ανιχνευτή Hessian blob, ο οποίος έχει υλοποιηθεί στο μέρος 2ο της εργαστηριακής άσκησης. Ο περιγραφέας χαρακτηριστικών του βασίζεται στο άθροισμα της απόκρισης κύματος Haar γύρω από το σημείο ενδιαφέροντος, τα οποία μπορούν επίσης να υπολογιστούν με τη βοήθεια της ολοκληρωμένης εικόνας. Η μέθοδος των SURF υπολογίζει πρώτα την γενική κατεύθυνση της γειτονιάς, έτσι ώστε να εξαχθούν περιστροφικά ανεξάρτητοι περιγραφητές.

- HOG (Histogram of Oriented Gradients)

Η τεχνική μετρά τις εμφανίσεις προσανατολισμού της παραγώγου στις εκτιμήσεις περιοχών της εικόνας.

Η εικόνα χωρίζεται σε μικρές συνδεδεμένες περιοχές (cells) και για τα pixels σε κάθε cell, καταρτίζεται ένα ιστόγραμμα κατευθύνσεων παραγώγου/κλίσης. Ο περιγραφέας είναι η συνένωση αυτών των ιστογραμμάτων.

Και οι δυο περιγραφητές έχουν ως είσοδο μια γειτονιά ενός σημείου ενδιαφέροντος και βασίζονται στην κωδικοποίηση της πληροφορίας υποτυμημάτων της γειτονιάς αυτής με χρήση της πρώτης κατευθυντικής παραγώγου.

Οι υλοποιήσεις των περιγραφητών δεν μας απασχόλησαν στην παρούσα άσκηση και θα γίνει χρήση των έτοιμων υλοποιήσεων που μας δόθηκαν στα εκτελέσιμα αρχεία μαζί με την εκφώνηση, και συγκεκριμένα με τις συναρτήσεις `featuresSURF(I,points)` και `featuresHOG(I,points)` για SURF και HOG αντίστοιχα.

### 3.1. Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας

Στο πείραμα αυτό έχουν χρησιμοποιηθεί τρεις εικόνες, οι οποίες έχουν παραμορφωθεί κατα 5 περιστροφές:  $-20^\circ$ ,  $-10^\circ$ ,  $0^\circ$ ,  $10^\circ$ ,  $20^\circ$ ) και των οποίο το μέγεθος έχει μεταβληθεί σε 4 κλίμακες (0.6, 0.8 1.0 1.2). Συνολικά έχουμε 20 μετασχηματισμούς ομοιότητας. Θα χρησιμοποιήσουμε μία εκ των εικόνων σαν εικόνα αναφοράς και θα προσπαθήσουμε να αντιστοιχίσουμε τους τοπικούς περιγραφητές αυτής της εικόνας με τους αντίστοιχους των υπολοίπων(matching).

#### 3.1.1

Χρησιμοποιούμε την *matching\_evaluation* σαν συνάρτηση εκτίμησης της περιστροφής των εικόνων και αποτίμησης τους.

Εκτελέσαμε τη διαδικασία δύο φορές, μία για ταίριασμα χρησιμοποιώντας την *featureSURF* και άλλη μία για τη *featureHOG*.

Παρατίθενται screenshots από το terminal της εκτέλεσης:

```
Avg. Scale Error for Image 1 with SURF features: 0.007
Avg. Theta Error for Image 1 with SURF features: 1.793

Avg. Scale Error for Image 2 with SURF features: 0.002
Avg. Theta Error for Image 2 with SURF features: 0.214

Avg. Scale Error for Image 3 with SURF features: 0.005
Avg. Theta Error for Image 3 with SURF features: 0.453
```

```

Avg. Scale Error for Image 1 with HOG features: 0.235
Avg. Theta Error for Image 1 with HOG features: 18.725

```

```

Avg. Scale Error for Image 2 with HOG features: 0.498
Avg. Theta Error for Image 2 with HOG features: 25.890

```

```

Avg. Scale Error for Image 3 with HOG features: 0.284
Avg. Theta Error for Image 3 with HOG features: 16.883

```

Παρατηρούμε κατα μία με δύο τάξεις μεγέθους μεγαλύτερα σφάλματα κλίμακας και γωνίας περιστροφής όταν εκτελέσαμε το evaluation με την featureHOG σε αντίθεση με τη χρήση της featureSURF.

### 3.1.2

Για κάθε έναν από τους 5 ανιχνευτές σημείων ενδιαφέροντος που υλοποιήσαμε και για τους δυο τοπικούς περιγραφητές να υπολογίσαμε την ικανότητα εκτίμησης της περιστροφής και κλίμακας των εικόνων για κάθε συνδυασμό.

Παρακάτω φαίνονται τα αποτελέσματα:

	Περιγραφητής SURF		Περιγραφητής HOG	
Μέθοδος ανίχνευσης Σημείων Ενδιαφέροντος	Avg. Scale Error	Avg. Theta Error	Avg. Scale Error	Avg. Theta Error
<i>Image Set 1</i>				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.007	1.793	0.235	18.725
Πολυκλιμακωτή Ανίχνευση Γωνιών	0.001	0.109	0.192	17.109
Μονοκλιμακωτή Ανίχνευση Blobs	0.049	2.099	0.121	15.471
Πολυκλιμακωτή Ανίχνευση Blobs	0.001	0.054	0.170	13.297
Πολυκλιμακωτή Ανίχνευση Blobs με BoxFilter	0.001	0.096	0.138	14.919
<i>Image Set 2</i>				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.002	0.214	0.498	25.890

Πολυκλιμακωτή Ανίχνευση Γωνιών	0.003	0.255	0.211	18.910
Μονοκλιμακωτή Ανίχνευση Blobs	0.001	0.066	0.272	14.185
Πολυκλιμακωτή Ανίχνευση Blobs	0.002	0.078	0.256	17.133
Πολυκλιμακωτή Ανίχνευση Blobs με BoxFilter	0.002	0.078	0.199	20.013
<i>Image Set 3</i>				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.005	0.453	0.284	16.883
Πολυκλιμακωτή Ανίχνευση Γωνιών	0.002	0.143	0.450	22.295
Μονοκλιμακωτή Ανίχνευση Blobs	0.041	1.610	0.128	24.956
Πολυκλιμακωτή Ανίχνευση Blobs	0.001	0.071	0.126	13.278
Πολυκλιμακωτή Ανίχνευση Blobs με BoxFilter	0.001	0.045	0.175	15.399

Αυτό που αναμέναμε ήταν οι πολυκλιμακωτοί ανιχνευτές ενδιαφέροντος να σημειώνουν μικρότερο σφάλμα, καθώς αναζητούν σημεία ενδιαφέροντος σε πολλές κλίμακες, αυξάνοντας έτσι την ακρίβεια της αναζήτησης. Συγκεκριμένα, συγκρίνοντας τους ανιχνευτές σημείων ενδιαφέροντος παρατηρούμε, ότι ο ανιχνευτής MultiscaleBox εμφανίζει ελάχιστα μικρότερο λάθος στο σφάλμα της γωνίας, το οποίο όμως βρίσκεται στην ίδια τάξη μεγέθους με τους προηγούμενούς του.

Αντίθετα για τον HOG παρατηρούμε ότι οι διαφορές ανάμεσα στα σφάλματα είναι ελάχιστες, με τους πολυκλιμακωτούς HarrisLaplacian MultiscaleBlobs και MultiscaleBox ανιχνευτές να έχουν παλι το προβάδισμα.

Μεταξύ των δύο περιγραφητών, βλέπουμε ότι ο SURF είναι αρκετά πιο ακριβής με προβάδισμα δύο τάξεων μεγέθους, σε συνδυασμό με όλους τους ανιχνευτές σημείων ενδιαφέροντος! Ο λόγος αναλύθηκε στο δεύτερο εργαστήριο του μαθήματος. Ο αλγόριθμος του περιγραφητή SURF εντάσσει την πληροφορία των κατευθυντικών παραγώγων σε ένα ευρύτερο σύνολο πληροφοριών. Κατά συνέπεια, δεν παρατηρείται σημαντική αλλοίωση και παραμόρφωση των εικόνων κατά τις περιστροφές και τις μεταβολές κλίμακας.

### 3.2 Κατηγοριοποίηση Εικόνων

Στο μέρος αυτό αξιολογήσαμε την επίδοση και την καταλληλότητα των διαφόρων ανιχνευτών και περιγραφητών σε ένα τυπικό πρόβλημα κατηγοριοποίησης εικόνων. Μας δόθηκε ένα σύνολο εικόνων από τη βάση Pascal VOC2005. Κάθε εικόνα ανήκει σε μια από τις τρεις κλάσεις (αυτοκίνητο, άνθρωπος και ποδήλατο). Σκοπός είναι η κατηγοριοποίηση της κάθε εικόνας στη σωστή κλάση χρησιμοποιώντας σαν χαρακτηριστικά αναγνώρισης τους περιγραφητές που θα κατασκευάσαμε.

Η διαδικασία που κατα την εκφώνηση ακολουθήσαμε ήταν η παρακάτω:

- Χρησιμοποιήσαμε την FeatureExtraction για εξαγωγή χαρακτηριστικών σε όλη τη βάση. Για τη χρήση ανιχνευτών χρησιμοποιήσαμε αποκλειστικά τους πολυκλιμακωτού, αφού όπως είδαμε και στο προηγούμενο ερώτημα αποτελούν τη βέλτιστοι επιλογή, καθώς παρουσιάζουν λιγότερο σφάλμα.
- Έπειτα χωρίσαμε τις εικόνες σε training και test sets χρησιμοποιώντας την createTrainTest
- Χρησιμοποιήσαμε την BagOfWords εκτελώντας kmeans για το clustering
- Εκπαιδεύσαμε εναν classifier στο training set και τον εφαρμόσαμε στο test set για την πρόβλεψη της κλάσης της κάθε εικόνας.

Πειραματιστήκαμε με τους διαφορετικούς ανιχνευτές και περιγραφητές (τους βέλτιστους) και τα αποτελέσματα που λάβαμε είναι τα παρακάτω:

Μέθοδος ανίχνευσης Σημείων Ενδιαφέροντος	SURF	HOG
Μονοκλιμακωτή Ανίχνευση Γωνιών	54.759%	61.665%
Πολυκλιμακωτή Ανίχνευση Γωνιών	58.621%	65.379%
Μονοκλιμακωτή Ανίχνευση Blobs	55.310%	65.241%
Πολυκλιμακωτή Ανίχνευση Blobs	58.207%	66.483%
Πολυκλιμακωτή Ανίχνευση Blobs με BoxFilter	56.414%	66.345%

Συμπερασματικά, για τη συγκεκριμένη βάση δεδομένων, η μέθοδος HOG σημειώνει υψηλότερες αποδόσεις. Ο λόγος εντοπίζεται στο ότι η μέθοδος αυτή, δίνοντας έμφαση στις κατευθυντήριες παραγώγους, παρέχει μια πυκνή επικαλυπτόμενη περιγραφή των περιοχών μια εικόνας και υπολογίζεται σε ένα πυκνό πλέγμα ομοιόμορφα κατανεμημένων κελιών. Επεξηγηματικά, ανιχνεύει και απομονώνει το αντικείμενο ενδιαφέροντος καθιστώντας έτσι απλούστερη την κατηγοριοποίηση ομοίων αντικειμένων.

Τέλος, τα παραπάνω στατιστικά μας φανερώνουν, ότι βέλτιστο συνδυασμό αποτελεί η πολυκλιμακωτή ανίχνευση Blobs με HOG περιγραφητή.