

# Όραση Υπολογιστών

## 1η Εργαστηριακή Άσκηση

Θέμα: Εντοπισμός σημείων ενδιαφέροντος και  
εξαγωγή χαρακτηριστικών σε εικόνες

ΣΧΟΛΗ: ΣΗΜΜΥ

Ονοματεπώνυμο	Αριθμός Μητρώου
Γιάννης Πιτόσκας	03115077
Γιώργος Χιονάς	03115132



## Μέρος 1: Ανίχνευση Ακμών σε Γκρίζες Εικόνες

Στην άσκηση αυτή θα ασχοληθούμε με την ανίχνευση ακμών σε πραγματικές εικόνες. Για το λόγο αυτό υλοποιήσαμε τη συνάρτηση EdgeDetect ακολουθώντας τα βήματα που αναγράφονται στην εκφώνηση της άσκησης. Η συνάρτηση αυτή παίρνει ως ορίσματα την θορυβώδη εικόνα  $I$ , τις παραμέτρους  $\sigma$  (χωρική κλίμακα της Gaussian), την  $\theta_{\text{edge}}$  (παράμετρος κατωφλίου) και μια λογική μεταβλητή, η οποία αντιστοιχεί την γραμμική ή μη προσέγγιση της Laplacian. Στη συνέχεια θα υπολογίσουμε τις αληθινές ακμές χρησιμοποιώντας την καθαρή εικόνα και θα συγκρίνουμε τα αποτελέσματα για να καταλάβουμε το ρόλο που παίζουν οι παράμετροι  $\sigma$ ,  $\theta_{\text{edge}}$  καθώς και το PSNR

Καταρχάς παραθέτουμε την εικόνα που θα επεξεργαστούμε

Real image



Στη συνέχεια παρατίθενται οι εικόνες που προκύπτουν με την προσθήκη λευκού θορύβου  $I(x, y) = I_0(x, y) + n(x, y)$ , οι οποίες θα χρησιμοποιηθούν ως είσοδοι στην συνάρτηση EdgeDetect.

Noisy image with psnr = 10



Noisy image with psnr = 20



Όπως είναι ήδη γνωστό αλλά παρατηρώντας το και από τα παραπάνω σχήματα η μείωση του PSNR επιφέρει μεγαλύτερο θόρυβο.

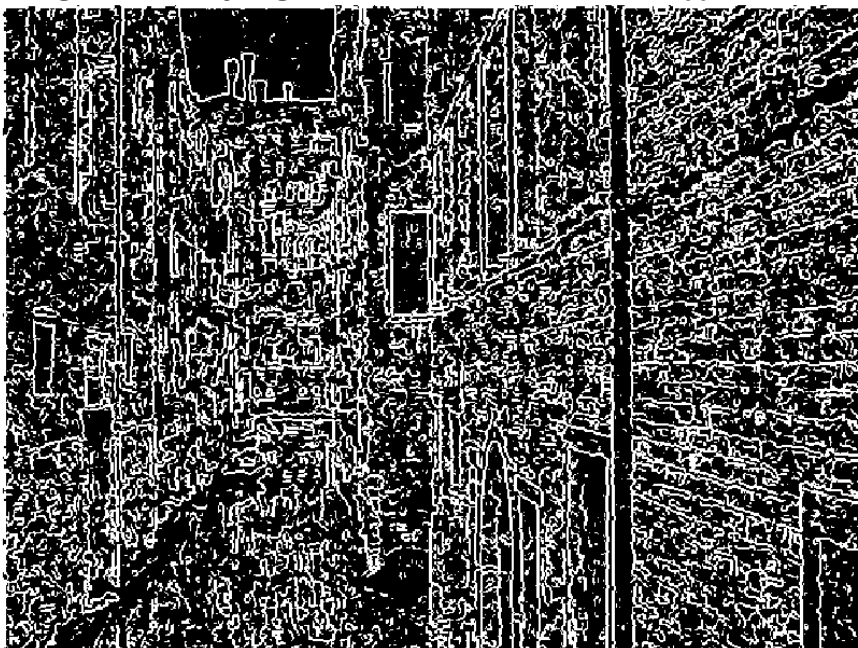
Ο υπολογισμός των πραγματικών ακμών με κατώφλι  $\theta_{\text{edge}} = 0.2$  μας δίνει το παρακάτω αποτέλεσμα:

**Real Edge detection**



Για είσοδο με  $PSNR = 20$  dB,  $\sigma = 1.5$  και  $\theta_{edge} = 0.2$  (ενδεικτικές τιμές από εκφώνηση) η συνάρτηση και με τις δυο διαφορετικές προσεγγίσεις Laplace μας δίνει :

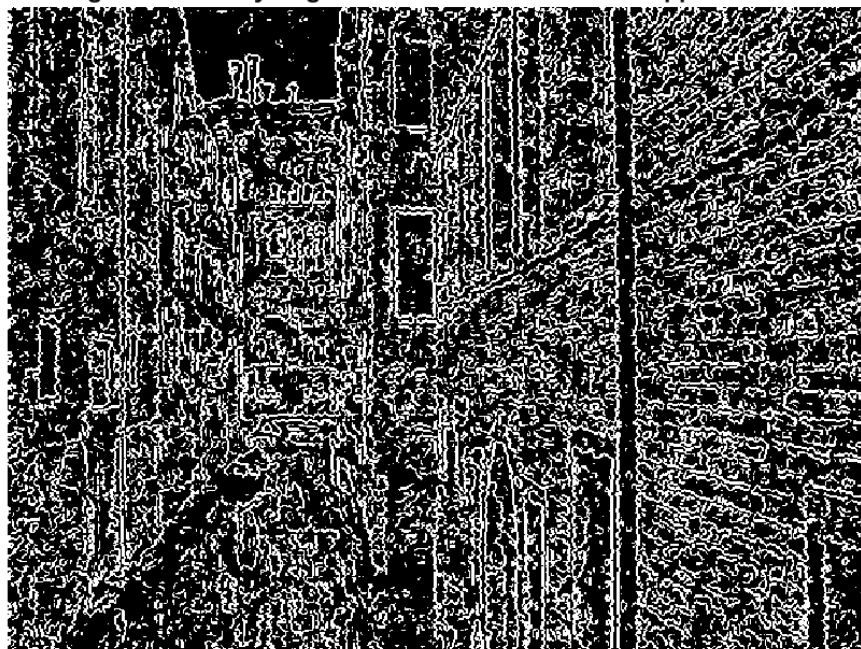
**Edge detection by EdgeDetect function with non linear approximation**



Σε αυτή την περίπτωση τα αποτελέσματα που παίρνουμε είναι :

$$Precision = 0.5939, Recall = 0.4413 \text{ και } C = \frac{Precision + Recall}{2} = 0.5176$$

**Edge detection by EdgeDetect function with linear approximation**



Σε αυτή την περίπτωση παίρνουμε :

$Precision = 0.5994$  ,  $Recall = 0.394$  και  $C = 0.4967$

Δοκιμάζοντας διάφορες τιμές για τις παραμέτρους στην συνάρτηση EdgeDetect μπορούμε να πούμε τα εξής. Η μη γραμμική προσέγγιση της Laplace μας δίνει καλύτερα αποτελέσματα σε σύγκριση με τη γραμμική. Όσο αυξάνεται το  $\sigma$  της Γκαουσιανής, το οποίο αποτελεί του μέγεθος του βαθυπερατού φίλτρου, οι ακμές υποχωρούν καθώς η εικόνα θολαίνει. Όσο αυξάνονται τα επίπεδα θορύβου της εικόνας, δηλαδή όσο μειώνεται το PSNR είναι λογικό ότι χρειαζόμαστε μεγαλύτερο  $\sigma$  για να εξομαλυνθεί η εικόνα με αποτέλεσμα να χάνουμε πραγματικές ακμές. Το  $\theta_{edge}$  αποτελεί ένα κατώφλι, το οποίο όσο αυξάνεται περιορίζει τα zerocrossings.

## Μέρος 2ο –Ανίχνευση Σημείων Ενδιαφέροντος (Interest Point Detection)

Σε αυτό το μέρος θα χρησιμοποιήσουμε ένα σύνολο από ανιχνευτές προκειμένου να ανιχνεύσουμε σημεία ενδιαφέροντος στις παρακάτω εικόνες, καθώς και να δούμε κατά πόσο λειτουργεί διαφορετικά ο καθέ ένας εκ των χρησιμοποιηθέντων ανιχνευτών.

### 2.1) Ανίχνευση Γωνιών

2.1.1) Υπολογίζουμε τα στοιχεία  $J_1, J_2, J_3$  του δομικού τανυστή  $J$  σε κάθε pixel  $(x, y)$  της εικόνας ως εξής:

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

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

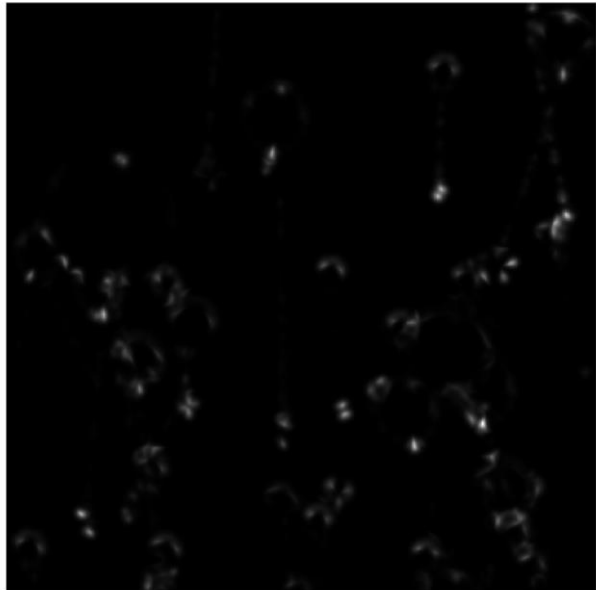
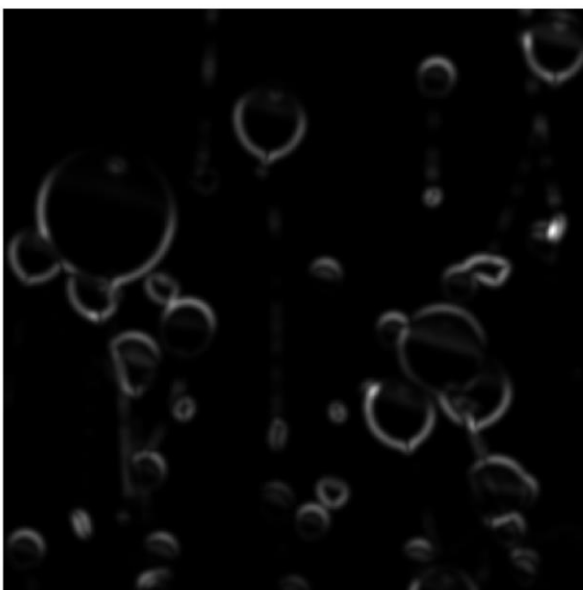
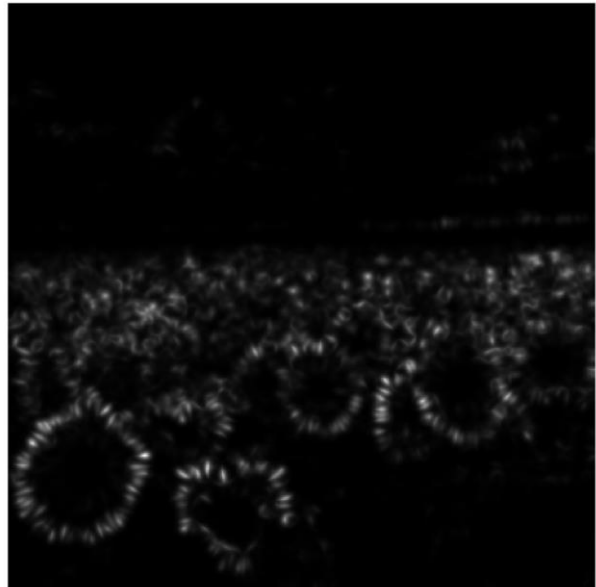
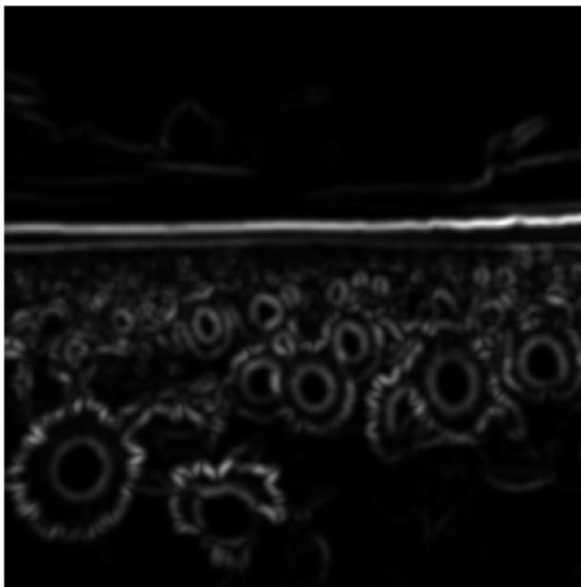
$$J_3(x, y) = G_\rho * \left( \frac{\partial I_\sigma}{\partial y} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

Όπου  $I_\sigma = G_\sigma * I$  και  $G_\rho, G_\sigma$  δισδιάστατοι Gaussian πυρήνες ομαλοποίησης με τυπικές αποκλίσεις  $\sigma$  (κλίμακα διαφόρισης) και  $\rho$  (κλίμακα ολοκλήρωσης) αντίστοιχα.

2.1.2) Υπολογίζουμε τις ιδιοτιμές  $\lambda_+, \lambda_-$  του  $J$  ως εξής:

$$\lambda_{\pm}(x, y) = \frac{1}{2} \left( J_1 + J_3 \pm \sqrt{(J_1 - J_3)^2 + 4J_2^2} \right)$$

Παρακάτω φαίνονται οι ιδιοτιμές  $\lambda_+$ ,  $\lambda_-$  για καθεμία από τις δύο εικόνες “sunflowers19.png” και “balloons19.png”:





2.1.3) Στη συνέχεια, δεδομένων των ιδιοτιμών εξάγουμε το παρακάτω κριτήριο γωνιότητας (cornerness criterion):

$$R(x, y) = \lambda_- \lambda_+ - k \cdot (\lambda_- + \lambda_+)^2$$

Επιλέγουμε ενδεικτικά παραμέτρους:

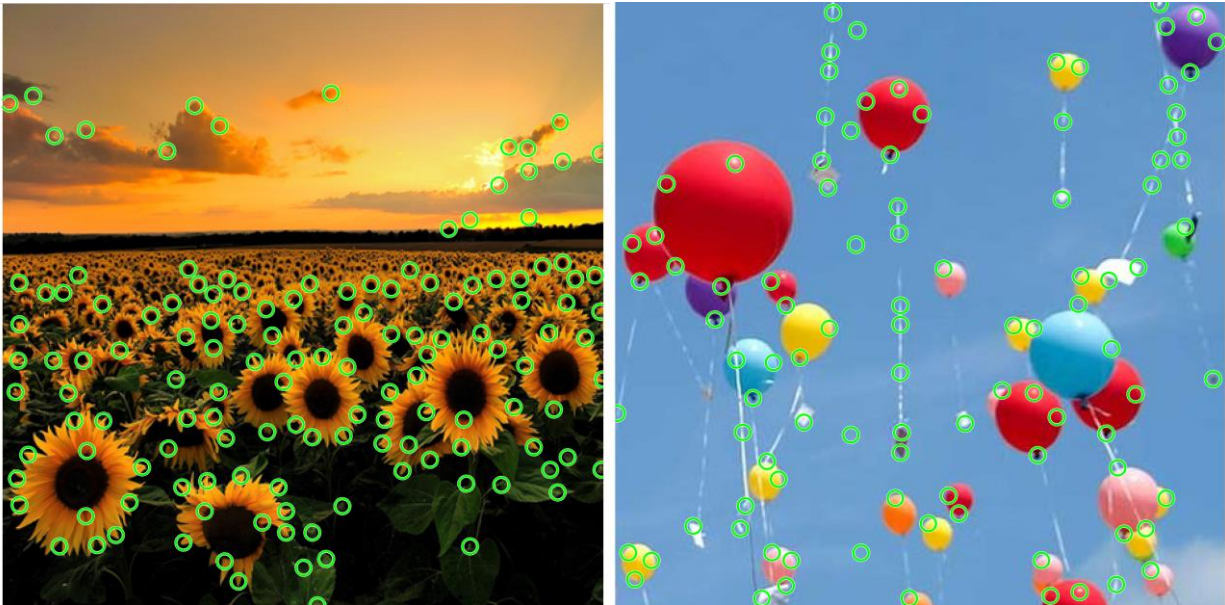
$$\sigma = 2;$$

$$\rho = 2.5;$$

$$k = 0.05;$$

$$\text{theta\_corn} = 0.008;$$

Τελικά επιλέξαμε τα σημεία που είναι τοπικά μέγιστα του  $R$  εντός τετραγωνικών παραθύρων και ξεπερνούν ένα κατώφλι  $\theta_{\text{corn}} \cdot R_{\text{max}}$ , και αναπαρηστούμε την εικόνα με τις γωνίες με την συνάρτηση που μας δόθηκε:



## 2.2) Πολυκλιμακωτή ανίχνευση γωνιών

2.2.1) Υπολογίζουμε τις κλίμακες διαφόρισης και ολοκλήρωσης ως εξής:

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

2.2.2) Ύστερα υπολογίζουμε την κανονικοποιημένη LoG για κάθε κλίμακα  $\sigma_i$  ως εξής:

$$|LoG(\mathbf{x}, \sigma_i)| = \sigma_i^2 |L_{xx}(\mathbf{x}, \sigma_i) + L_{yy}(\mathbf{x}, \sigma_i)|, \quad i = 0, \dots, N - 1$$

Και ύστερα κάνουμε discard τα σημεία για τα οποία η κλίμακα που ανιχνεύθηκαν δεν μεγιστοποιεί την LoG μετρική σε μια γειτονιά δύο διαδοχικών κλιμακών.

Επιλέγουμε ενδεικτικά παραμέτρους:

$$\sigma_0 = 2$$

$$\rho_0 = 2.5$$

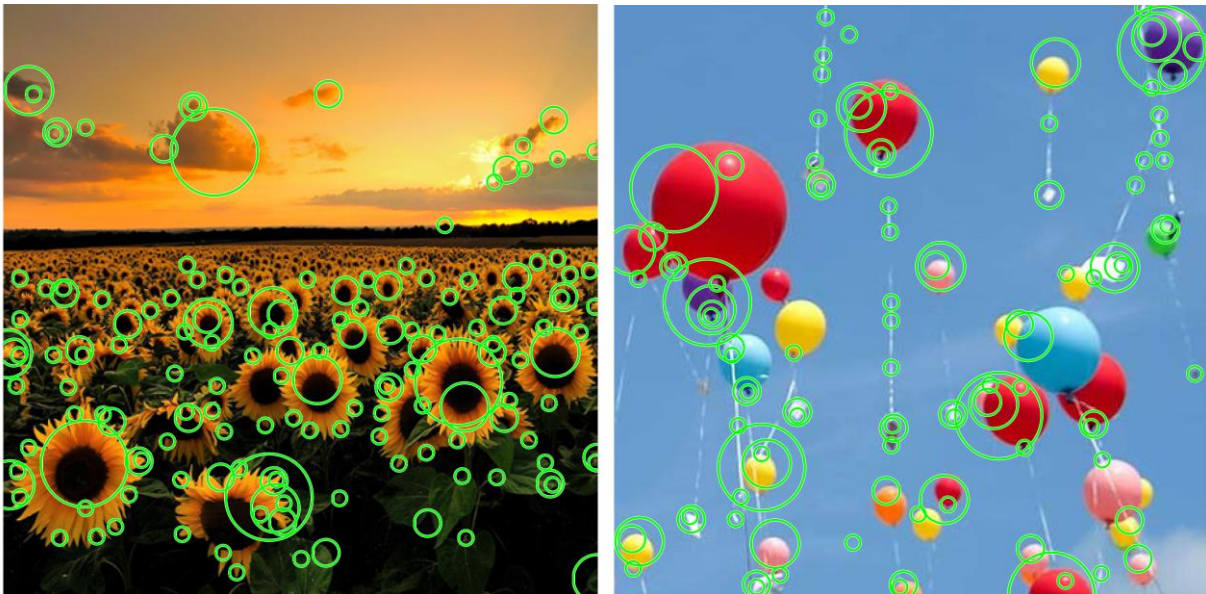
$$k = 0.05$$

$$\text{theta\_corn} = 0.008$$

$$s = 1.8$$

$$N = 4$$

Έτσι παίρνουμε τα παρακάτω αποτελέσματα:



## 2.3) Ανίχνευση Blobs

2.3.1) Για την ανίχνευση των blobs αρχικά υπολογίζουμε τον πίνακα Hessian για κάθε pixel της εικόνας ως εξής:

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

όπου  $L_{xx}(x, y, \sigma) = \frac{\partial^2}{\partial x^2} \{I_\sigma(x, y)\}$ ,  $L_{yy}(x, y, \sigma) = \frac{\partial^2}{\partial y^2} \{I_\sigma(x, y)\}$  και  $L_{xy}(x, y, \sigma) = \frac{\partial^2}{\partial x \partial y} \{I_\sigma(x, y)\}$

Επομένως απαιτείται ο υπολογισμός των παραπάνω δεύτερων μερικών παραγώγων.

Το κριτήριο που χρησιμοποιούμε για την ανίχνευση blobs είναι το παρακάτω:

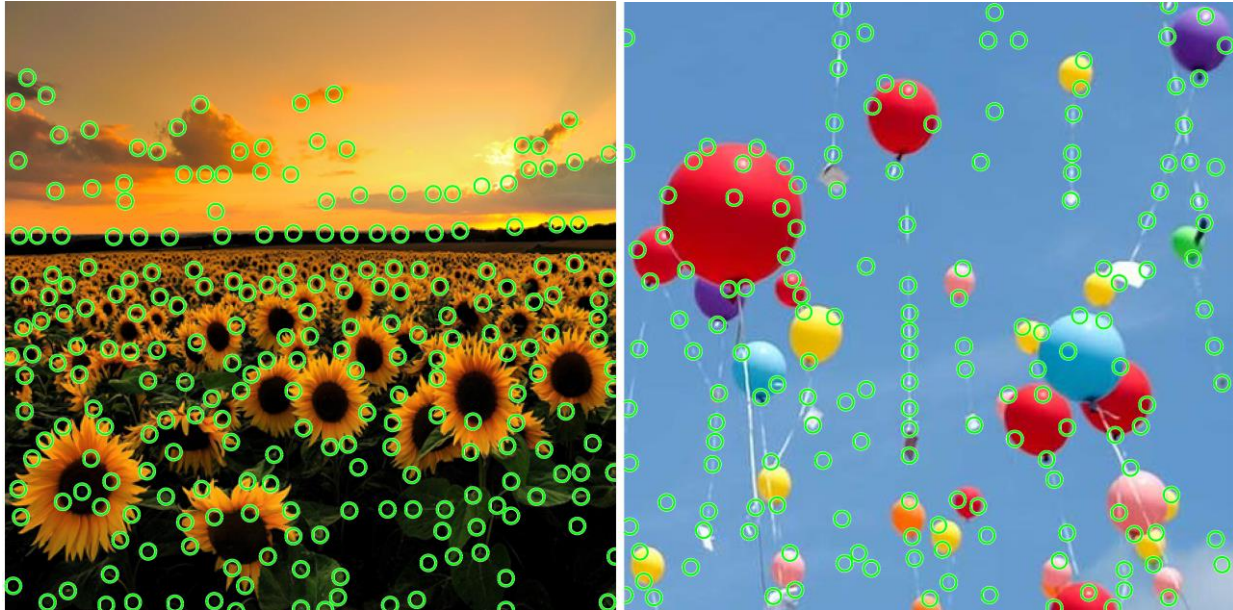
$$R(x, y) = \det(H(x, y))$$



Θεωρούμε ως blobs τα σημεία που είναι τοπικά μέγιστα και έχουν μεγαλύτερη τιμή από ένα ορισμένο threshold. Παρακάτω τα αναπαριστούμε και οι παράμετροι που χρησιμοποιήσαμε για να πετύχουμε αυτό το αποτέλεσμα είναι οι εξής:

$$\sigma = 2$$

$$\text{theta\_corn} = 0.008$$



## 2.4) Πολυκλιμακωτή ανίχνευση blobs

Η μέθοδος που θα ακολουθήσουμε είναι αντίστοιχη με αυτήν που περιγράφηκε παραπάνω για την ανίχνευση γωνιών σε πολλές κλίμακες.

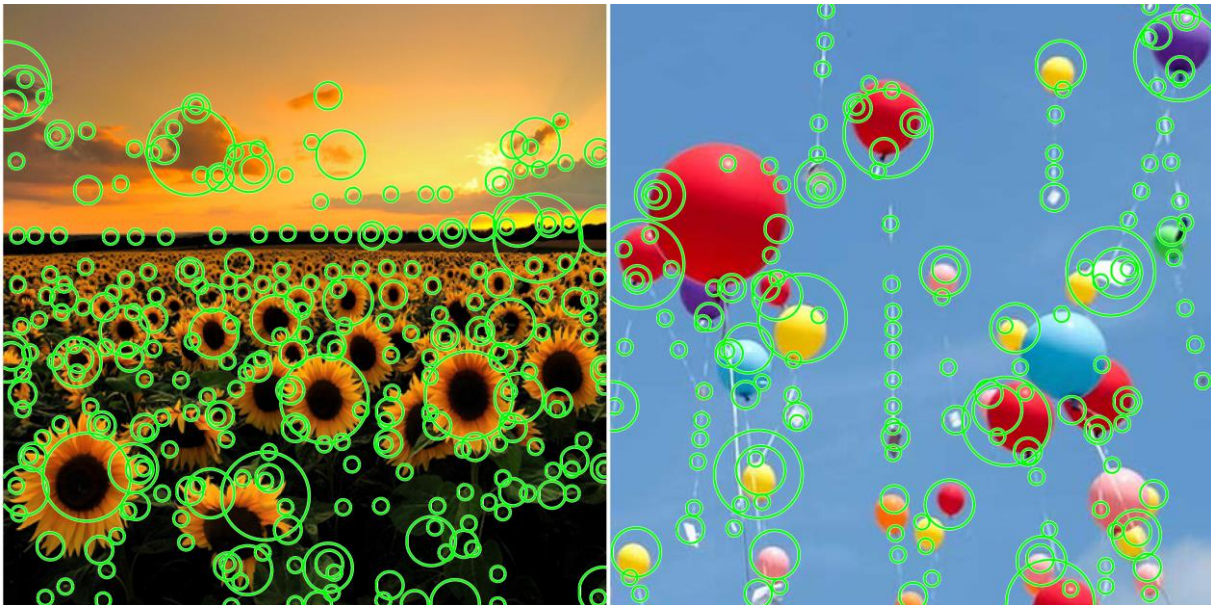
Οι παράμετροι που χρησιμοποιούμε εδώ είναι:

$$\sigma_0 = 2$$

$$\text{theta\_corn} = 0.008$$

$$s = 1.8$$

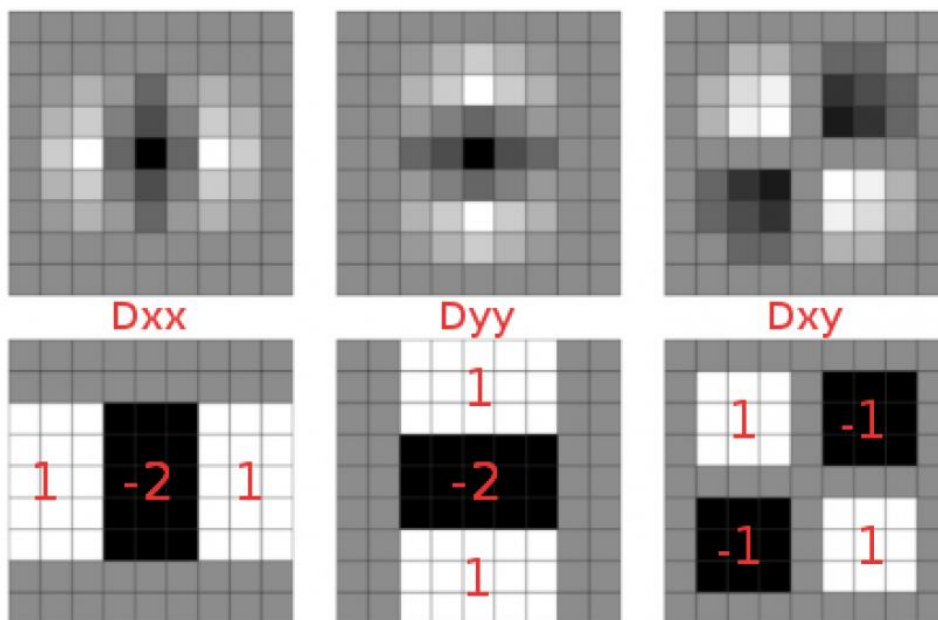
$$N = 4$$



## 2.5) Επιτάχυνση με χρήση Box Filters και Ολοκληρωτικών εικόνων

Σε αυτό το μέρος της άσκησης θα προσεγγίσουμε της Hessian με τη χρήση φίλτρων δεύτερης παραγώγου με Box Filters. Σκοπός είναι να μειώσουμε την υπολογιστική πολυπλοκότητα. Αυτό θα επιτευχθεί με την κατασκευή της ολοκληρωτικής εικόνας, μέσω της οποίας τα αθροίσματα πάνω σε ορθογώνια παράθυρα (που αντιστοιχούν σε συνέλιξη) υπολογίζονται σε σταθερό χρόνο καθώς ο χρόνος δεν εξαρτάται από το μέγεθος του παραθύρου.

Για υπολογισμό των  $L_{xx}$ ,  $L_{yy}$ ,  $L_{xy}$  φιλτράρουμε την εικόνα με τα παρακάτω τρία φίλτρα αντίστοιχα.

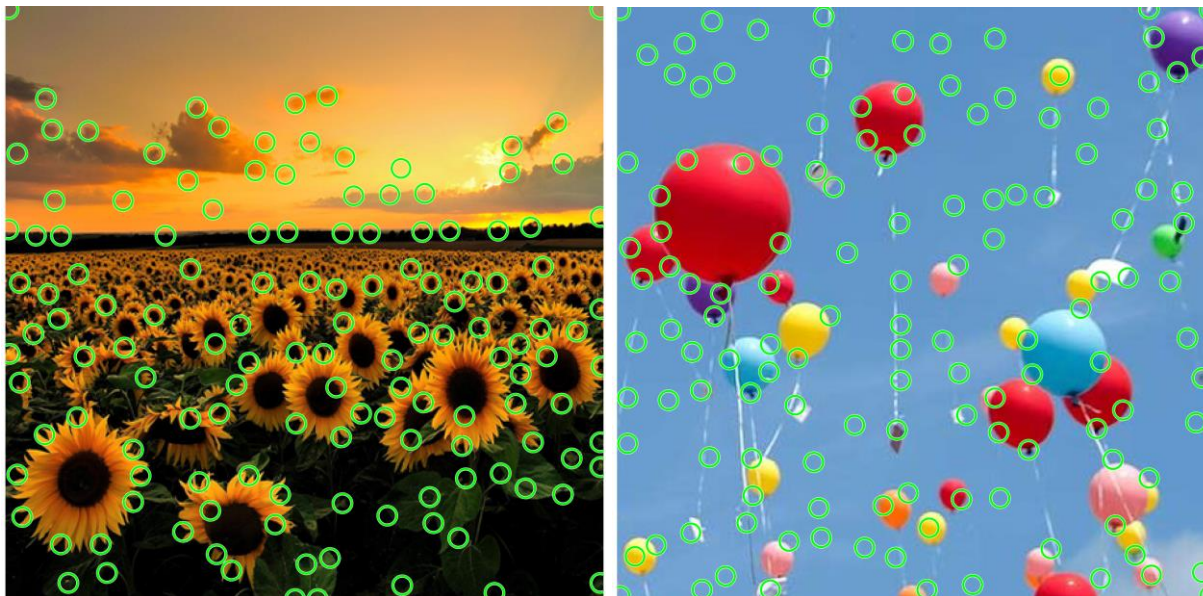


Τελικά η προσέγγιση για την Hessian ορίζουσα προκύπτει από τον τύπο:

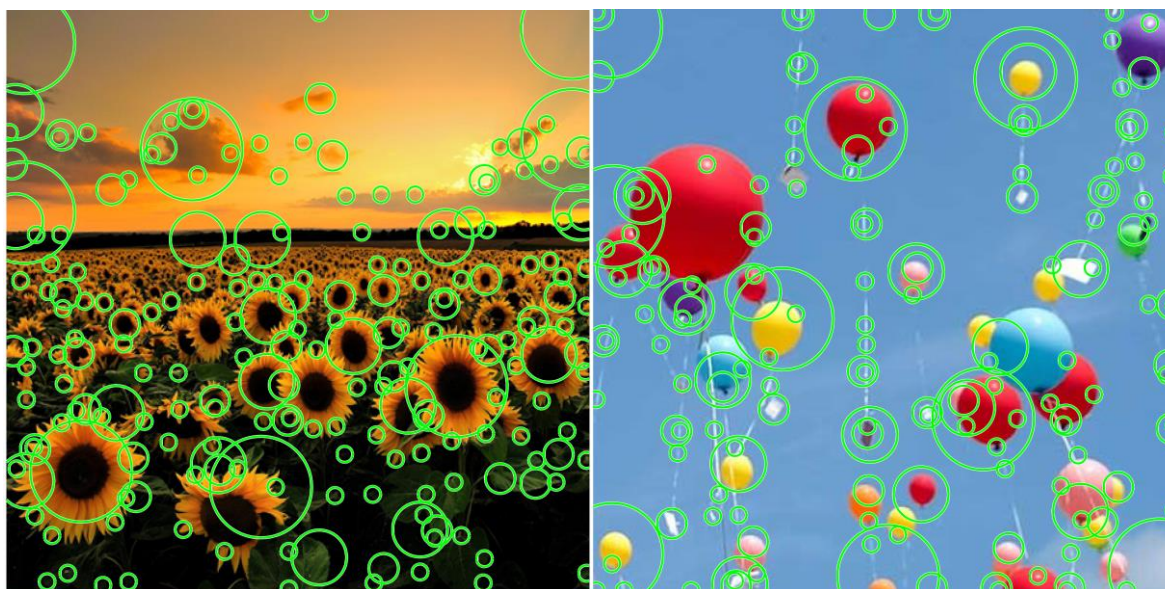
$$\det(H_{approx}) = L_{xx} * L_{yy} - (0.9L_{xy})^2$$



Παρακάτω παραθέτουμε τα αποτελέσματα από την υλοποίηση Blob detection με box filters σε μια κλίμακα, για  $\sigma = 2.5$ :



Για την πολυκλιμακωτή ανίχνευση σημείων ενδιαφέροντος εργαζόμαστε όπως στο ερώτημα με τη Hessian. Τα αποτελέσματα που παίρνουμε με αρχική κλίμακα  $\sigma=2.2$  και παράγοντα 1.85 είναι :



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

Σε αυτό το μέρος του εργαστηρίου καλούμαστε να κάνουμε δύο πράγματα. Αρχικά, προσπαθούμε να εκτιμήσουμε την κλίση και την κλίμακα μιας εικόνας ως προς κάποια αρχική μορφή που είχε. Η αρχική μορφή αναφέρεται και σαν εικόνα αναφοράς. Για να επιτύχουμε το στόχο μας θα χρησιμοποιήσουμε τοπικούς περιγραφητές για την κάθε εικόνα, δηλαδή θα αναπαραστήσουμε τα σημεία ενδιαφέροντος της με δύο τρόπους. Και οι δύο επιτελούν παρόμοια λειτουργία η οποία είναι η περιγραφή μιας γειτονιάς κάθε σημείου ενδιαφέροντος μέσω της κατευθυντήριας πρώτης παραγώγου.

Επαναλαμβάνοντας αυτή τη μέθοδο για όλα τα σημεία ενδιαφέροντος της εικόνας θα αποκτήσουμε ένα σύνολο περιγραφητών που θα μας δώσουν μια εκτίμηση της κλίσης και της κλίμακάς της. Οι τοπικοί περιγραφητές που θα χρησιμοποιηθούν είναι η SURF και η HOG. Ένας ακόμη παράγοντας που θα επηρεάσει την εκτίμηση του σφάλματος μας είναι η συνάρτηση που θα χρησιμοποιηθεί για τον εντοπισμό των σημείων ενδιαφέροντος της εικόνας. Για να έχουμε ένα καθαρό αποτέλεσμα θα χρησιμοποιήσουμε και τις πέντε μεθόδους που αναπτύξαμε στο προηγούμενο μέρος της εργασίας μας, δηλαδή την Harris Stephens, Harris Laplacian, Hessian, Hessian Laplacian και μια πολυκλιμακωτή υλοποίηση των Box Filters. Για κάθε μια από τις παραπάνω μεθόδους θα περιγράψουμε τα σημεία ενδιαφέροντος που προκύπτουν και με την SURF και με την HOG.

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

Descriptor	Detector	Scale Error	Theta Error
SURF	One Scale Corner Detect	0,055 0,033 0,039	1,12 1,45 4,93
SURF	Multi Scale Corner Detect	0,004 0,007 0,011	0,19 0,44 0,56
SURF	One Scale Blob Detect	0,109 0,009 0,089	6,75 0,34 4,30
SURF	Multi Scale Blob Detect	0,002 0,004 0,002	0,16 0,35 0,20
SURF	Box Filter Multi Scale Blob Detect	0,005 0,048 0,008	0,38 3,65 0,36
HOG	One Scale Corner Detect	0,232 0,316 0,474	21,49 21,17 32,68
HOG	Multi Scale Corner Detect	0,159 0,290 0,206	20,52 28,74 17,27
HOG	One Scale Blob Detect	0,268 0,232 0,412	33,55 26,39 33,34



HOG	Multi Scale Blob Detect	0,167 0,291 1,204	14,40 22,94 10,12
HOG	Box Filter Multi Scale Blob Detect	0,229 0,304 0,237	25,72 29,74 16,57

Βλέπουμε ότι ο συνδυασμός Hessian Laplacian (Multiscale Blob Detect) με SURF δίνει συνολικά τα μικρότερα σφάλματα. Επομένως, αυτός ο συνδυασμός μπορεί να μας δώσει σε αυτήν την περίπτωση την καλύτερη εκτίμηση και της κλίμακας αλλά και της περιστροφής μιας εικόνας με αναφορά την αρχική της.

### 3.2) Κατηγοριοποίηση Εικόνων

Στο δεύτερο τμήμα αυτού του μέρους, είχαμε μια νέα βάση δεδομένων με άλλες εικόνες οι οποίες ταξινομούνταν σε τρεις κατηγορίες: αμάξια, ποδήλατα και άνθρωποι. Σκοπός μας είναι να μπορέσουμε να φτιάξουμε ένα μοντέλο και να το εκπαιδεύσουμε έτσι ώστε να κατηγοριοποιεί τις εικόνες με τέτοιο τρόπο έτσι ώστε να έχουμε όσο καλύτερη δυνατότητα διάκρισης μιας νέας εικόνας και ένταξής της σε κάποια κατηγορία από τις προηγούμενες. Για να κατορθώσουμε το στόχο μας, αρχικά θα πρέπει να εξάγουμε περιγραφητές για κάθε εικόνα του dataset μας. Αυτό θα γίνει χρησιμοποιώντας τις SURF και HOG πάνω στα σημεία ενδιαφέροντος που προκύπτουν από τις πολυκλιμακωτές εκδόσεις των συναρτήσεων που χρησιμοποιήσαμε προηγουμένως. Στη συνέχεια, θα χωρίσουμε τους περιγραφητές σε train και test data και θα εφαρμόσουμε πάνω στο train data τον αλγόριθμο συσταδοποίησης kmeans με σκοπό να μπορέσουμε να ομαδοποιήσουμε τους περιγραφητές μας. Στη συνέχεια, αφού έχουμε υπολογίσει τα κέντρα, θα βρούμε αυτό που απέχει λιγότερο από κάθε διάνυσμα περιγραφητών, για κάθε διάνυσμα περιγραφητών και με βάση αυτή την παραπάνω απόσταση θα κατασκευάσουμε το ιστόγραμμα που προκύπτει από την συχνότητα εμφάνισης της κάθε λέξης του λεξικού μας, δηλαδή του καθενός περιγραφητή. Τέλος, θα εφαρμόσουμε τον αλγόριθμο Support Vector Machine για να ορίσουμε τα όρια που θα διαχωρίζουν τα διαφορετικά αντικείμενα. Αφού εκτελέσαμε τα παραπάνω βήματα για όλους τους δυνατούς συνδυασμούς, καταλήξαμε στα εξής αποτελέσματα:

Ποσοστό Επιτυχίας	Harris Laplacian (Multiscale Angle Detect)	Hessian Laplacian (Multiscale Blob Detect)	Box Filters Multiscale Blob Detect
SURF	66.6%	66.2%	64.7%
HOG	67.6%	67.5%	67.1%

Συμπεραίνουμε λοιπόν ότι Harris Laplacian (Multiscale angle detect) σε συνδυασμό με την HOG και πάλι μας δίνει βέλτιστα αποτελέσματα, αφού σε ποσοστό περίπου 67.6% (απέχοντας αμελητέα από την Hessian Laplacian με HOG) κάνει σωστό classification της εκάστοτε εικόνας.

