

# 1η Εργαστηριακή Άσκηση

Εντοπισμός Σημείων Ενδιαφέροντος και Εξαγωγή  
Χαρακτηριστικών σε Εικόνες

Μάθημα : Όραση Υπολογιστών



Ροή Σ

Συνεργάτες :

- Βαβουλιώτης Γεώργιος ( Α.Μ. : 03112083 )
- Σταυρακάκης Δημήτριος ( Α.Μ. : 03112017 )

## Μέρος 2 : Ανίχνευση Σημείων Ενδιαφέροντος

Στο δεύτερο μέρος της άσκησης θα ασχοληθούμε με την ανίχνευση σημείων ενδιαφέροντος σε εικόνες.

### Ανίχνευση Γωνιών

Μας δίνετε μια εικόνα εισόδου, την οποία παραθέτω παρακάτω, στην οποία καλούμαστε να ανιχνεύσουμε τις γωνίες της.

Original Image



Αρχικά θα υλοποιήσουμε ένα ανιχνευτή, οποίος θα υλοποιεί την κλασσική μέθοδο των Harris-Stephens. Η συνάρτηση **CornerDetect()** που σας επισυνάπτω στο zip αρχείο ουσιαστικά υλοποιεί τον ανιχνευτή αυτό. Στην πράξη για να μπορέσω να ανιχνεύσω τις ακμές μια εικόνας με τη μέθοδο αυτή θα πρέπει αρχικά να υπολογίσω το δομικό τανυστή  $J$  σε κάθε pixel της εικόνας, σύμφωνα με τις σχέσεις :

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

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

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

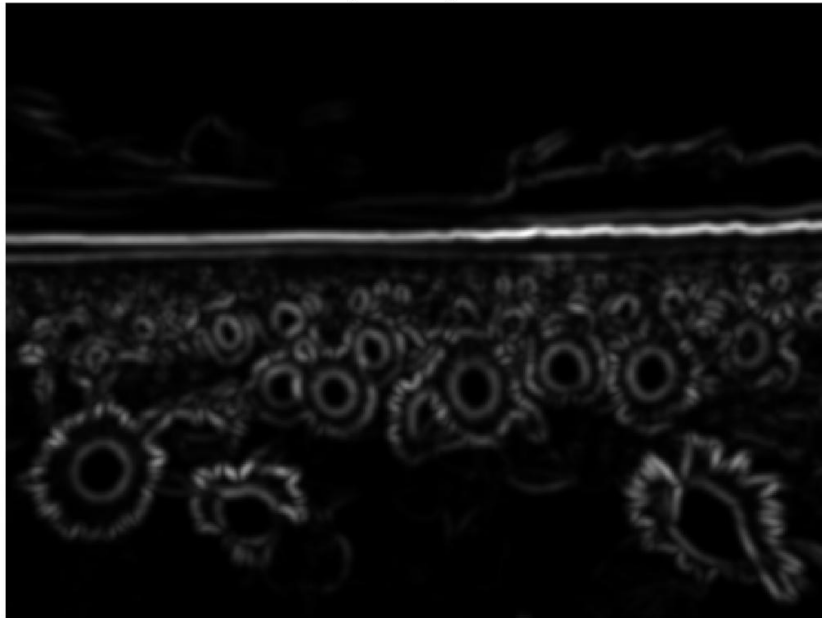
Απο τις παραπάνω μαθηματικές σχέσεις αντιλαμβάνομαι ότι ο δομικός τανυστής  $J$  επηρεάζεται άμεσα από τις τυπικές αποκλίσεις  $\sigma$  και  $\rho$ , οι οποίες αντιστοιχούν στη κλίμακα διαφόρισης και ολοκλήρωσης αντίστοιχα.

Στη συνέχεια θα πρέπει να υπολογιστούν οι ιδιοτιμές του τανυστή  $J$  σε κάθε σημείο του, σύμφωνα με την παρακάτω μαθηματική σχέση :

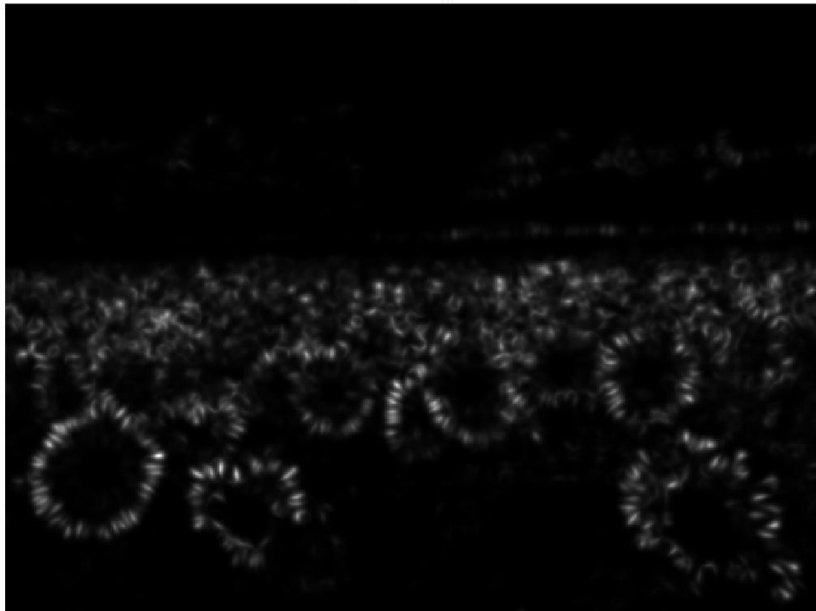
$$\lambda_{\pm}(x, y) = \frac{1}{2} \left( J_1 + J_3 \pm \sqrt{(J_1 - J_3)^2 + 4J_2^2} \right)$$

Στη συνέχεια παραθέτω τα  $\lambda_{\pm}(x, y)$  σαν γκριζες εικόνες :

Grayscale Image for  $I_+$



Grayscale Image for  $I_-$



**Σχολιασμός-Παρατηρήσεις :** Απο τις παραπάνω γραφικές των ιδιοτιμών του δομικού τανυστή J παρατηρούμε ότι οι ακμές της εικόνας αντιστοιχούν σε περιοχές στις οποίες η μια ιδιοτιμή είναι σημαντικά μεγαλύτερη απο την άλλη. Επίσης πως οι γωνίες της εικόνας αντιστοιχούν σε περιοχές στις οποίες οι ιδιοτιμές έχουν παραπλήσια τιμή και η τιμή αυτή είναι σχετικά μεγάλη. Είναι επίσης προφανές ότι οι ομαλές περιοχές της εικόνας αντιστοιχούν σε σχεδόν μηδενικές τιμές και των δυο ιδιοτιμών.

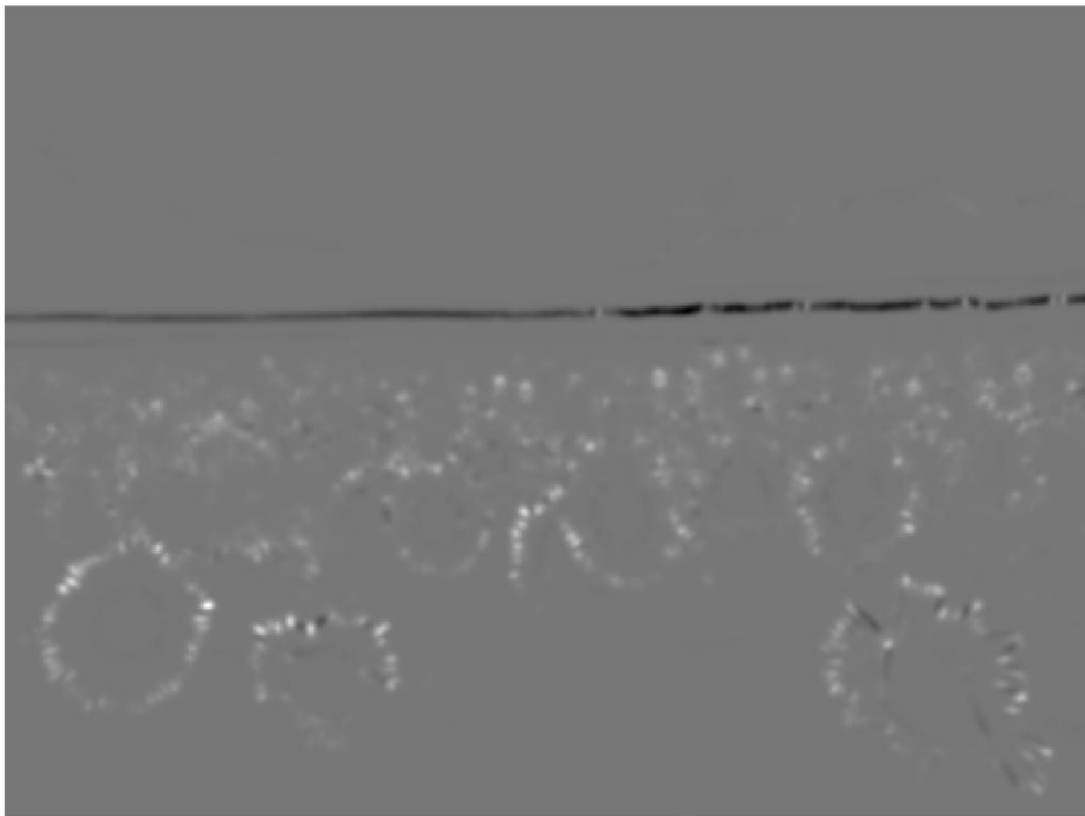
Οι παραπάνω παρατηρήσεις, θα μπορούσαν πολύ εύκολα να προκύψουν και απο το κριτήριο γωνιότητας(*cornerness criterion*) R το οποίο απο μαθηματικής άποψης εκφράζεται ως :

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

όπου k μια θετική παράμετρος.

Αν παρουσιάσουμε το R σε ενα διάγραμμα θα παρατηρήσουμε πως τα σημεία τα οποία αντιστοιχούν σε γωνίες είναι τα σημεία στα οποία το R παίρνει θετική τιμή. Το διάγραμμα του R φαίνεται παρακάτω(επαληθεύει τον παραπάνω ισχυρισμό):

**Cornerness Criterion**





Με βάση το παραπάνω κριτήριο γωνιότητας θα επιλέξω τελικά σαν γωνίες εκείνα τα pixel της εικόνας τα οποία :

- είναι μέγιστα του  $R$  εντός τετραγωνικών παραθύρων
- αντιστοιχούν σε τιμή του  $R$  μεγαλύτερη από ένα ποσοστό του ολικού μεγίστου του  $R$ , δηλαδή :

$$R(x, y) > \theta_{corn} \cdot R_{max}$$

Εφαρμόζοντας τους παραπάνω ελέγχους-κριτήρια εντοπίσαμε τις ακμές της δοσμένης εικόνας για δοσμένες τιμές των παραμέτρων από την εκφώνηση της άσκησης. Το αποτέλεσμα παρουσιάζεται παρακάτω (τα σημεία με πράσινο χρώμα αντιστοιχούν στις γωνίες):



**Παρατήρηση:** Η παραπάνω ανίχνευση ακμών δεν είναι η βέλτιστη αφού είναι προφανές ότι εξαρτάται από τις παραμέτρους που δίνω. Θα μπορούσε να γίνει μια μελέτη σχετικά με την επίδραση κάθε παραμέτρου στο αποτέλεσμα αλλά και η εύρεση της βέλτιστης τετράδας παραμέτρων.

Αν διατηρήσουμε τις τιμές των παραμέτρων στις δοσμένες και αλλάζουμε κάθε φορά μια απο αυτές παίρνουμε τα παρακάτω αποτελέσματα :

Χαμηλότερο  $\sigma$ (αριστερά) - Μεγαλύτερο  $\sigma$ (δεξιά) :



Χαμηλότερο  $\rho$  (αριστερά) - Μεγαλύτερο  $\rho$  (δεξιά) :





Χαμηλότερο  $k$ (αριστερά) - Μεγαλύτερο  $k$ (δεξιά) :



Χαμηλότερο  $\theta_{corn}$ (αριστερά) - Μεγαλύτερο  $\theta_{corn}$ (δεξιά) :



**Σχολιασμός :** Όταν η τυπική απόκλιση  $\sigma$  αυξάνεται χάνω αρκετές ακμές αφού ανιχνεύω μεγαλύτερες ακμές της εικόνας, ενώ όταν μειώνεται παρατηρούνται περισσότερα πράσινα κυκλάκια, δηλαδή γίνεται πιο εστιασμένη ανίχνευση γωνιών. Όταν η τυπική απόκλιση  $\rho$  μεταβάλλεται(αυξάνεται ή μειώνεται) παρατηρείται αμελητέα μεταβολή στην ανίχνευση γωνιών. Όταν η θετική παράμετρος  $k$  ή η παράμετρος κατωφλίου  $\theta_{corn}$  μειωθεί τότε ανιχνεύονται περισσότερες γωνίες ενώ αν αυξηθεί κάποια απο τις δυο παραμέτρους τότε ανιχνεύονται λιγότερες ακμές.

## Πολυκλιμακωτή Ανίχνευση Γωνιών

Το ερώτημα αυτό είναι συνέχεια του παραπάνω ερωτήματος, αφού θα υλοποιήσουμε ένα ανιχνευτή Harris-Laplacian, ο οποίος θα ανιχνεύει γωνίες σε πολλαπλές κλίμακες. Για να γίνει αυτό θα πρέπει πρώτα για κάθε κλίμακα να εκτελέσω τη συνάρτηση `CornerDeductNoVis()` η οποία είναι όμοια με την `CornerDetect()`, μόνο που δεν καλεί την δοσμένη `interest_points_visualization()` αλλά επιστρέφει τις γωνίες κάθε φορά και τις αποθηκεύω σε ένα πίνακα C. Αφού το κάνω αυτό για κάθε κλίμακα και ταυτόχρονα για κάθε κλίμακα διαφορίσης υπολογίζουμε την κανονικοποιημένη Λαπλασιανή της Γκαουσιανής (Laplacian of Gaussian LoG). Στο επόμενο στάδιο απορρίπτουμε τα σημεία για τα οποία η κλίμακα που ανιχνεύθηκαν δεν μεγιστοποιεί την LoG μετρική σε μια γειτονιά 2 διαδοχικών κλιμάκων (έχει υλοποιηθεί στα scripts του Matlab που υπάρχουν στο φάκελο code του zip αρχείου). Στη συνέχεια σας παραθέτω το αποτέλεσμα που πήραμε από την πολυκλιμακωτή ανίχνευση γωνιών για 6 κλίμακες :



**Σχολιασμός :** Παρατηρώ ότι δεν ανιχνεύονται ακριβώς οι ίδιες γωνίες με αυτές που ανίχνευσε η μέθοδος Harris-Stephens, γεγονός το οποίο είναι λογικό αφού η επιλογή των τελικών γωνιών γίνεται με χρήση διαφορετικής μετρικής. Επίσης με την μέθοδο αυτή ανιχνεύοντας τις γωνίες για κάθε κλίμακα και κρατώντας αυτές που πληρούν την μετρική που αναφέρθηκε καταφέρνουμε να ανιχνεύσουμε γωνίες για τις οποίες μπορούμε να είμαστε σε μεγάλο βαθμό σίγουροι πως πρόκειται για πραγματικές γωνίες της εικόνας μας.

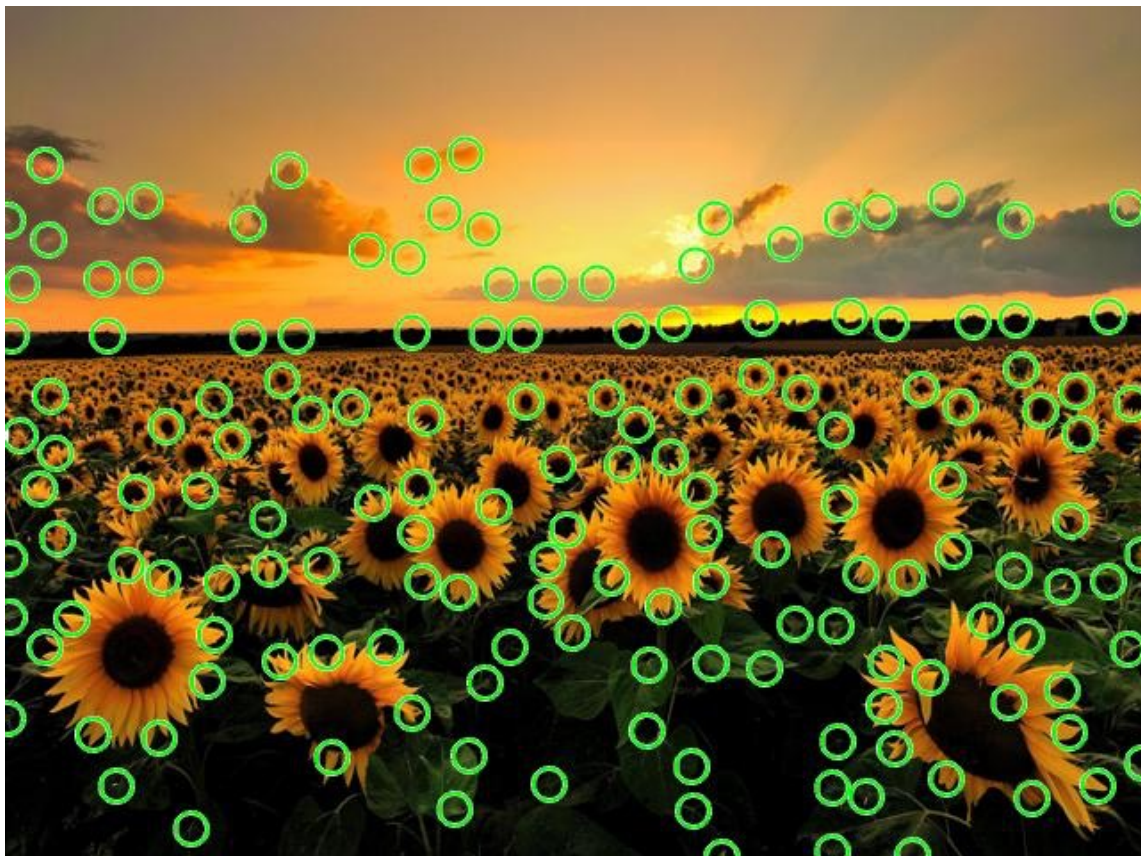


## Ανίχνευση Blobs

Μια απο τις βασικότερες κατηγορίες σημείων ενδιαφέροντος βασίζονται στην ανίχνευση blobs, που ορίζονται ως περιοχές με κάποια ομοιογένεια που διαφέρουν σημαντικά απο την γειτονιά τους. Για να μπορέσω να βρω τέτοιες περιοχές θα κάνω χρήση μερικών παραγώγων και συγκεκριμένα της Hessian, η οποία ορίζεται ως εξής :

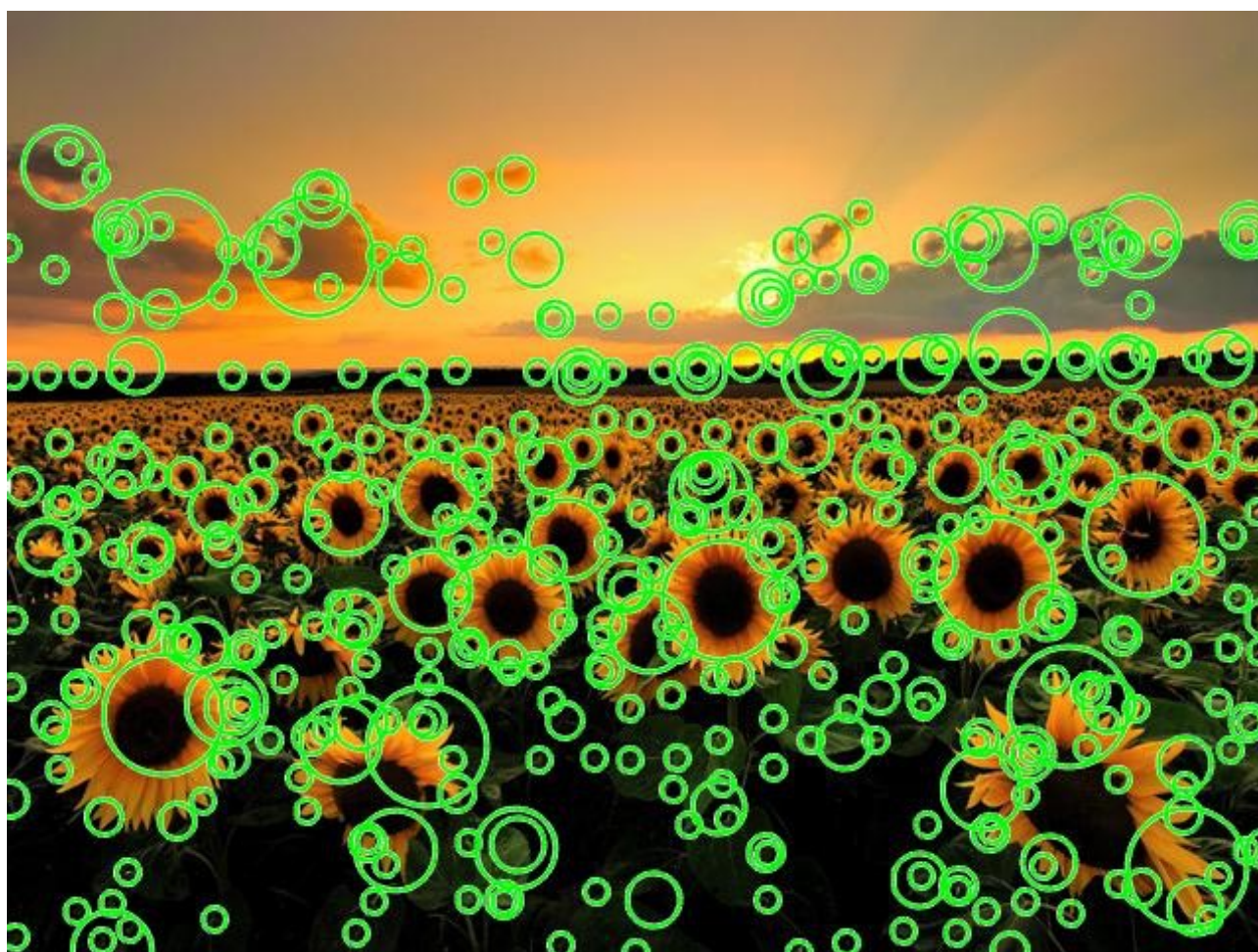
$$H(x, y) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

Στη μέθοδο αυτή, θα υπολογίσω αρχικά τις μερικές παραγώγους, ώστε να κατασκευάσω την Hessian, επιλέγοντας για την παράμετρο  $\sigma$  μια ενδεικτική τιμή. Έπειτα κατασκευάζω το κριτήριο  $R(x, y) = \det(H(x, y))$  για κάθε pixel της εικόνας. Τέλος, επιλέγω σαν σημεία ενδιαφέροντος εκείνα τα οποία είναι τοπικά μέγιστα και έχουν τιμή μεγαλύτερη από ένα κατάλληλα ορισμένο κατώφλι, ακριβώς όπως και στην ανίχνευση γωνιών με την μέθοδο Harris. Η ανίχνευση Blobs υλοποιήθηκε στο Matlab με όνομα **BlobDetect()** και αν τις δώσουμε σαν παραμέτρους την αρχική εικονά και τις ενδεικτικές τιμές για τις άλλες δυο παραμέτρους παίρνουμε σαν αποτέλεσμα, το παρακάτω:



## Πολυκλιμακωτή Ανίχνευση Blobs

Στο ερώτημα αυτό της άσκησης, ουστικά γίνεται κάτι ανάλογο με την πολυκλιμακωτή ανίχνευση γωνιών. Αυτό που αλλάζει είναι ότι τα αρχικά σημεία ενδιαφέροντος επιλέγονται χρησιμοποιώντας τον ανιχνευτή blobs και ταυτόχρονα υπολογίζεται εκ νέου η κανονικοποιημένη Λαπλασιανή της Γκαουσιανής (Laplacian of Gaussian LoG). Το δεύτερο στάδιο περιέχει την επιλογή σημείων ενδιαφέροντος που μεγιστοποιούν την LoG μετρική σε μια γειτονιά 2 διαδοχικών κλιμάκων (έχει υλοποιηθεί στα scripts του Matlab που υπάρχουν στο φάκελο code του zip αρχείου). Στη συνέχεια σας παραθέτω το αποτέλεσμα που πήραμε από την πολυκλιμακωτή ανίχνευση blobs για 5 κλίμακες :



**Σχολιασμός :** Παρατηρώντας τα αποτελέσματα που δίνει η ανίχνευση blobs με αυτά που δίνει η πολυκλιμακωτή ανίχνευση blobs παρατηρώ ότι η δεύτερη δίνει σημαντικά καλύτερα αποτελέσματα, δηλαδή ανιχνεύει και μεγαλύτερου μεγέθους blobs λόγω των διαφορετικών κλιμάκων που χρησιμοποιώ (χαρακτηριστικό παράδειγμα αποτελεί η ανίχνευση του κέντρου των λουλουδιών που καταφέρνει η πολυκλιμακωτή μέθοδος blobs κάτι το οποίο δεν συμβαίνει με την μονής κλίμακας ανίχνευση blobs, όπως φαίνεται και από τα παραπάνω σχήματα).



## Επιτάχυνση με την χρήση Box Filters και Ολοκληρωτικών Εικόνων

Στο ερώτημα αυτό αρχικά θα υπολογίσουμε την ολοκληρωτική εικόνα που μας ζητείται. Θα πρέπει να παρατηρηθεί πως ο υπολογισμός της ολοκληρωτικής εικόνας έγινε απο εμάς και όχι χρησιμοποιώντας την έτοιμη συνάρτηση που παρέχει το Matlab για δημιουργία ολοκληρωτικών εικόνων. Για τον υπολογισμό της ολοκληρωτικής εικόνας κάναμε χρήση της έτοιμης συνάρτησης *cumsum()*, δημιουργώντας αρχικά μια εικόνα που είναι η εφαρμογή της *cumsum()* κατά γραμμές στην αρχική εικόνα και μετά εκ νέου εφαρμογή της *cumsum()* στην νέα εικόνα κατά στήλες. Η εικόνα που προέκυψε είναι η ζητούμενη ολοκληρωτική εικόνα, η οποία υπολογίστηκε διότι είναι πολύ αποτελεσματική για την υλοποίηση αθροισμάτων εντός ορθογώνιων περιοχών για εφαρμογή φίλτρων, δηλαδή έχει πολυπλοκότητα ανεξάρτητη του μεγέθους του παραθύρου του φίλτρου.

Στη συνέχεια, αυτό που πρέπει να κάνουμε είναι να υπολογίσουμε τα  $L_{xx}$ ,  $L_{yy}$  και  $L_{xy}$  τα οποία μας είναι χρήσιμα για να μπορέσουμε να κάνουμε ανίχνευση των σημείων ενδιαφέροντος της εικόνας με χρήση της ολοκληρωτικής εικόνας που υπολογίσαμε παραπάνω με τη βοήθεια των Box Filters  $D_{xx}$ ,  $D_{yy}$  και  $D_{xy}$ .

Χρησιμοποιώντας το παράρτημα που δίνετε στην εκφώνηση της άσκησης έχουμε ότι το αθροισμα των φωτεινοτήτων μιας ορθογώνιας περιοχής που οριοθετείται από τα σημεία A,B,C,D προκύπτει από την ολοκληρωτική εικόνα από τον τύπο:

$$\sum \sum_{ABCD} I(i, j) = S_A + S_C - S_B - S_D$$

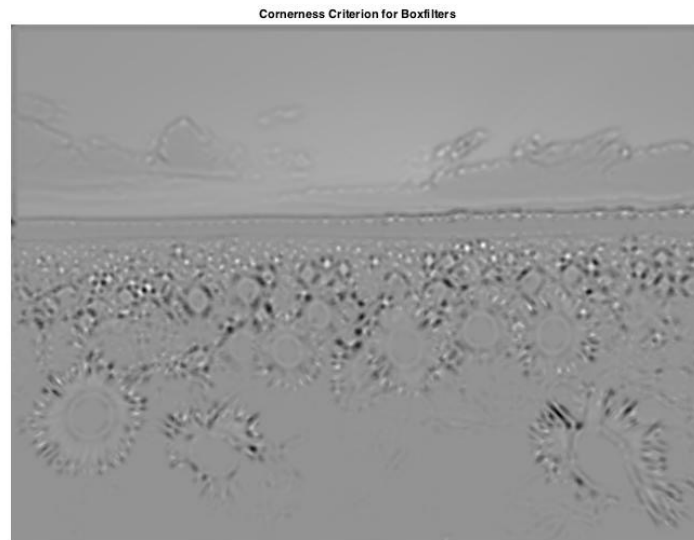
Αυτό μας δίνει την δυνατότητα να υπολογίσουμε την συνέλιξη του φιλτραρίσματος ταχύτατα, αφού πλέον ο υπολογισμός της είναι ανεξάρτητος απο το μήκος του παραθύρου και αυτό μας δίνει προσφέρει ταχύτητα στην ανίχνευση των γωνιών.

Στο σημείο αυτό θα ανιχνεύσουμε τα σημεία ενδιαφέροντος σύμφωνα με όσα κάναμε παραπάνω, δηλαδή δημιουργούμε ενα αλγόριθμο μονής κλίμακας BoxFilter χρησιμοποιώντας σαν κριτήριο γωνιότητας το παρακάτω :

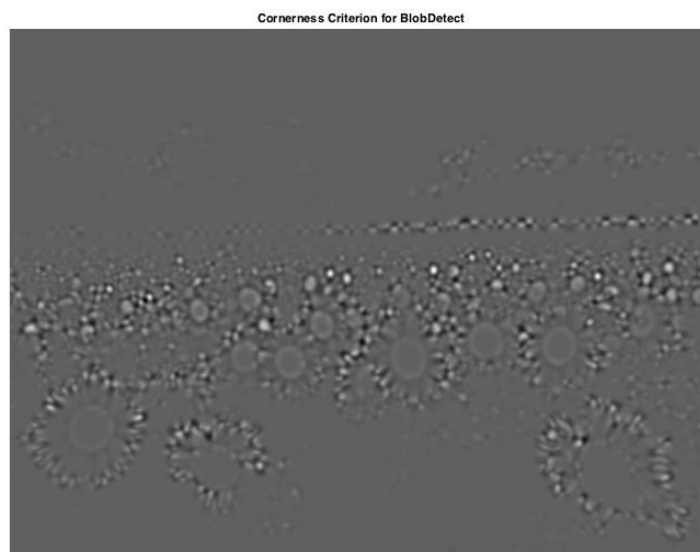
$$R(x, y) = L_{xx}(x, y)L_{yy}(x, y) - (0.9L_{xy}(x, y))^2$$



Το γράφημα του παραπάνω κριτηρίου γωνιότητας R φαίνεται παρακάτω:



Το γράφημα του κριτηρίου γωνιότητας R της απλής Hessian μεθόδου φαίνεται παρακάτω:

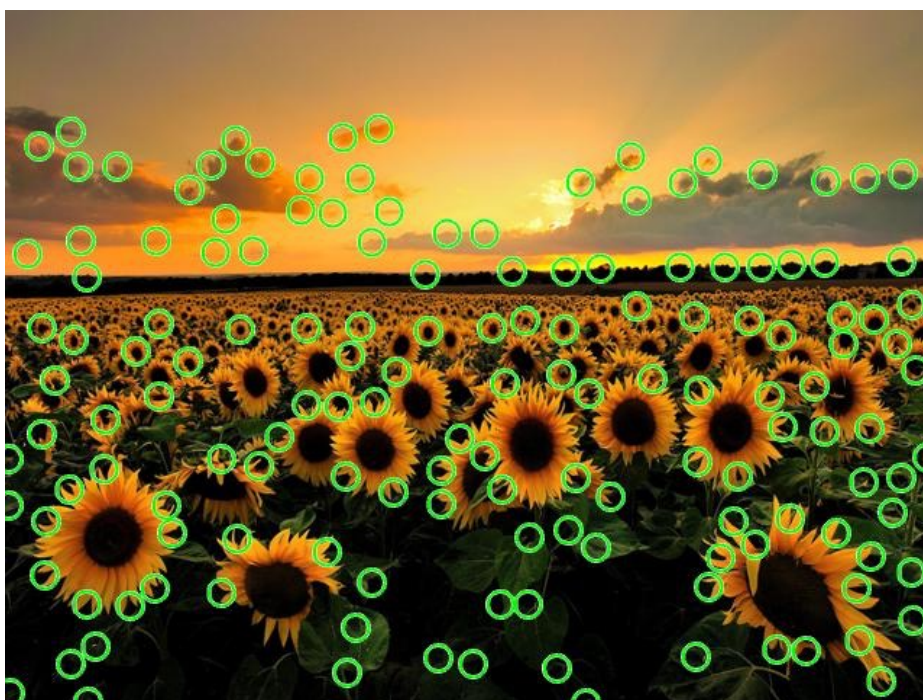


Επιλέγοντας τυχαία κάποιες κλίμακες  $\sigma$  ( $\sigma \in (2,10)$ ) παρουσιάζω τα αποτελέσματα τα οποία προέκυψαν με την παραπάνω μέθοδο ανίχνευσης ακμών :

**Για  $\sigma = 2$  :**



**Για  $\sigma = 3$  :**



Για  $\sigma = 5$  :



Για  $\sigma = 9$  :





**Παρατηρήσεις :** Απο τα παραπάνω γραφήματα καταλαβαίνω ότι όσο αυξάνω την κλίμακα η ανίχνευση με BoxFilters χάνει αρκετές γωνίες, δηλαδή έχει απόκλιση απο τη προηγούμενη μέθοδο αφού στη προσέγγιση με BoxFilters εισάγεται σημαντικό σφάλμα. Συμπερασματικά, η χρήση ολοκληρωτικών εικόνων και BoxFilters για να υπολογίσουμε τα  $L_{xx}$ ,  $L_{yy}$  και  $L_{xy}$  αυξάνει σημαντικά την ταχύτητα υπολογισμού του αποτελέσματος και προσφέρει πολύ καλής ποιότητας αποτελέσματα για σχετικά μικρές τιμές της κλίμακας αλλά όταν η κλίμακα παίρνει μεγαλύτερες τιμές τότε η ποιότητα του αποτελέσματος μειώνεται σημαντικά, ουσιαστικά υπάρχει ένα trade-off ανάμεσα σε ταχύτητα υπολογισμού και ποιότητα αποτελέσματος.

Στο τέλος του μέρους αυτού θα κάνουμε εκ νέου πολυκλιμακωτή ανίχνευση σημείων ενδιαφέροντος, όπως στα προηγούμενα ερωτήματα, με τη μέθοδο αυτή. Τα αποτελέσματα τα οποία πήραμε φαίνονται παρακάτω:



**Παρατηρήσεις :** Η πολυκλιμακωτή ανίχνευση με χρήση BoxFilters και ολοκληρωτικών εικόνων, όπως φαίνεται και απο τα παραπάνω σχήματα, προσφέρει ένα αρκετά καλό αποτέλεσμα για την ανίχνευση των γωνιών της εικόνας. Παρατηρώντας καλά τα παραπάνω σχήματα, βλέπουμε οτι παρότι προσφέρουν ένα πολύ καλό αποτέλεσμα, αντάξιο του αποτελέσματος που πήραμε με πολυκλιμακωτή ανίχνευση γωνιών(ερώτημα 2.2) και με την πολυκλιμακωτή ανίχνευση blobs(ερώτημα 2.4). Επίσης παρατηρούμε οτι τα παραπάνω σχήματα περιέχουν κάποιες αστοχίες οι οποίες είναι αρκετά αισθητές, γεγονός το οποίο είναι λογικό και οφείλεται στο ότι στον υπολογισμό των  $L_{xx}$ ,  $L_{yy}$ ,  $L_{xy}$  χρησιμοποιούμε τα Box Filters και της ολοκληρωτικής εικόνας, κάτι το οποίο μας προσφέρει ταχύτητα αλλά μας χαλάει την ποιότητα όσο αυξάνεται η κλίμακα.

