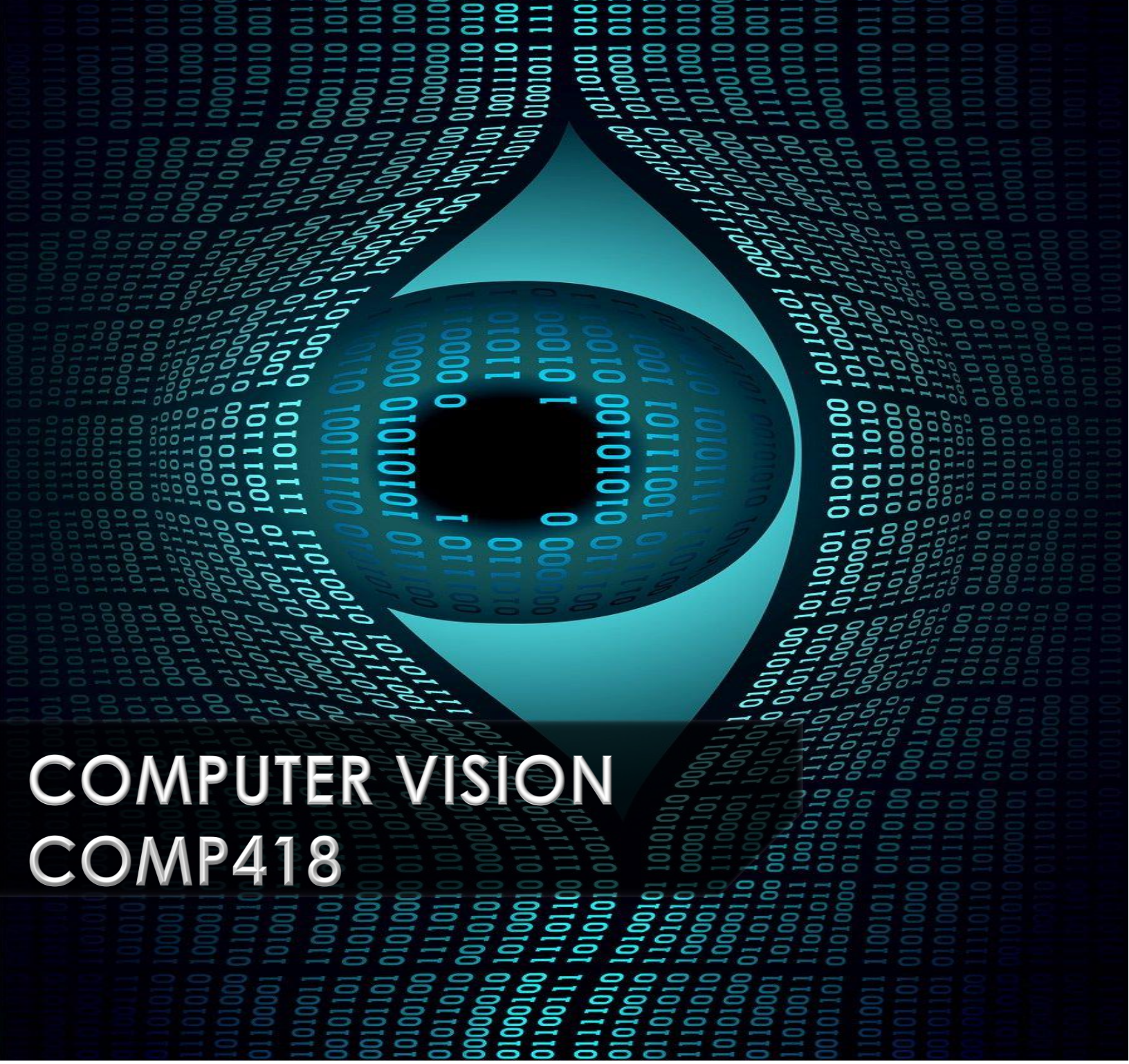


2nd Assignment

21/05/21

Θεοδωράκη Εμμανουέλα

2014030238

Ομάδα 19

HMMY

DESCRIPTION

Για την εργασία αυτή, υλοποιείται ένας αλγόριθμος για Laplacian Blob Detection ο οποίος ακολουθεί τα εξής στάδια:

1. Δημιουργία Laplacian of Gaussian φίλτρου
2. Δημιουργία ενός Laplacian Scale-Space επαναληπτικά
    a. Φιλτράρισμα εικόνας με ένα scale-normalized Laplacian για το τρέχον scale
    b. Αποθήκευση του τετραγώνου της Laplacian απόκρισης
    c. Υπολογισμός επόμενου scale, αυξάνοντας τον παράγοντα k
3. Εύρεση του extrema, με την εκτέλεση 3D Non-Maximum Suppression
4. Εμφάνιση των κύκλων που εντοπίστηκαν στις χαρακτηριστικές τους κλίμακες.

Αρχικά, ολόκληρη η διαδικασία του αλγορίθμου τρέχει σε επανάληψη για το κάθε ένα αρχείο τύπου .jpg που βρίσκεται στον υποφάκελο '../Data/' του project το οποίο διαβάζεται με την imread() και μετατρέπεται σε grayscale εικόνα χρησιμοποιώντας την rgb2gray, αλλά και σε double, με την im2double.

Επίσης, γίνεται η αρχικοποίηση των παραμέτρων:

- (σ) sigma=2 : μία τιμή ανάμεσα στο [1-3]
- (scale multiplier) k = 1.19 : μία τιμή αρκετά μεγαλύτερη από το 1 αλλά όχι τόσο ώστε να αυξάνει το scale αρκετά γρήγορα,
- (number of levels in scale-space) n=15 : επιλέχθηκε σύμφωνα με τις προτάσεις της εκφώνησης της εργασίας.

Αφού η αρχική εικόνα έχει φτάσει στην επιθυμητή μορφή για φιλτράρισμα, δίνεται ως όρισμα στην συνάρτηση **createLoG_slow.m** μαζί με τις μεταβλητές: n, sigma, k. Η συνάρτηση αυτή, έχει σκοπό να δημιουργήσει μία πυραμίδα από αποκρίσεις ενός Laplaccian of Gaussian φίλτρου, το οποίο εφαρμόζει στην εικόνα που δίνεται με τη βοήθεια της συνάρτησης MATLAB, fspecial(). Εδώ υλοποιείται η λιγότερο αποδοτική μεθοδολογία κατά την οποία εφαρμόζεται επαναληπτικά το φίλτρο στην εικόνα αυξάνοντας το μέγεθος του kernel κάθε φορά πολλαπλασιάζοντας με τον παράγοντα k (`newSigma = sigma * k^(scale-1))`. Η εφαρμογή του φίλτρου έγινε για `hsize = ceil(newSigma*3)*2+`1 δηλαδή περιττό μέγθος για την αποφυγή shifting όπως αναφέρεται και στη mathworks.com. Στη συνέχεια, γίνεται η κανονικοποίηση του LoGfilter και η αποθήκευση του τετραγώνου της απόκρισης στο filtI. Η χρήση των εντολών `tic;` και `toc;` Έγινε για να συγκριθούν με την πιο αποδοτική υλοποίηση στην createLoG_slow.m.

Σε επόμενο , για να εφαρμόσουμε Non-maximum suppression , αρχικά κάνουμε σε 2D slices χρησιμοποιώντας τη συνάρτηση MATLAB orfdilt2() λαμβάνοντας το τοπικό μέγιστο των γιε τονικών pixel. Στη συνέχεια, χρησιμοποιείται για να συγκριθούν οι γειτονιές των slices και να βρεθεί το μέγιστο. Εφαρμόζοντας ετσι μία 3D Non-maximum suppression σε όλο το sale space.

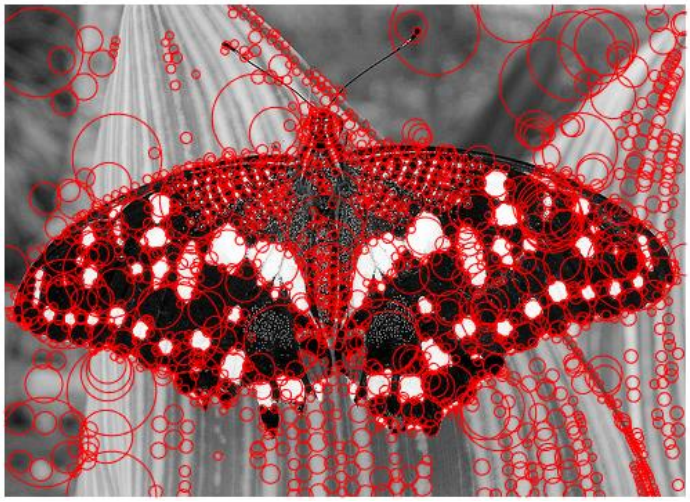
Τέλος, το threshold επιλέχθηκε 0.008 καθώς μετα από δοκιμές είχε καλύτερο αποτέλεσμα. Χρησιμοποιήθηκε στη συνάρτηση in2sub ώστε να προσδιοριστούν οι συντεταγμεενες που θα δοθούν στην show_all_circles και να προκύψει το τελικό αποτέλεσμα με τους κύκλους επάνω στις εικόνες και με ανάλογη ακτίνα.

RESULTS

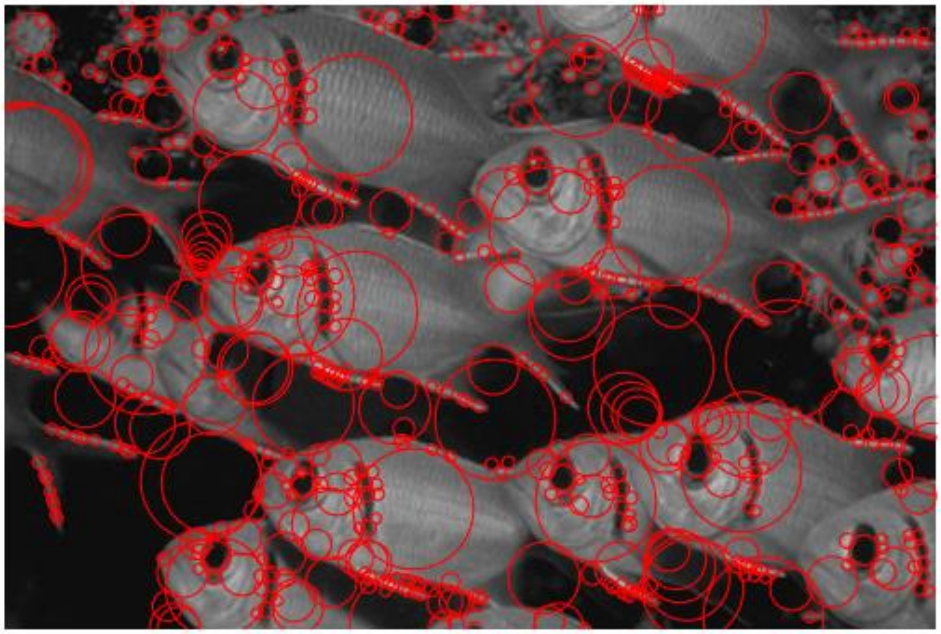
Στο σημείο αυτό παρουσιάζονται τα αποτελέσματα που προκύπτουν από την εκτέλεση του assignment2.m.

1st PAIR

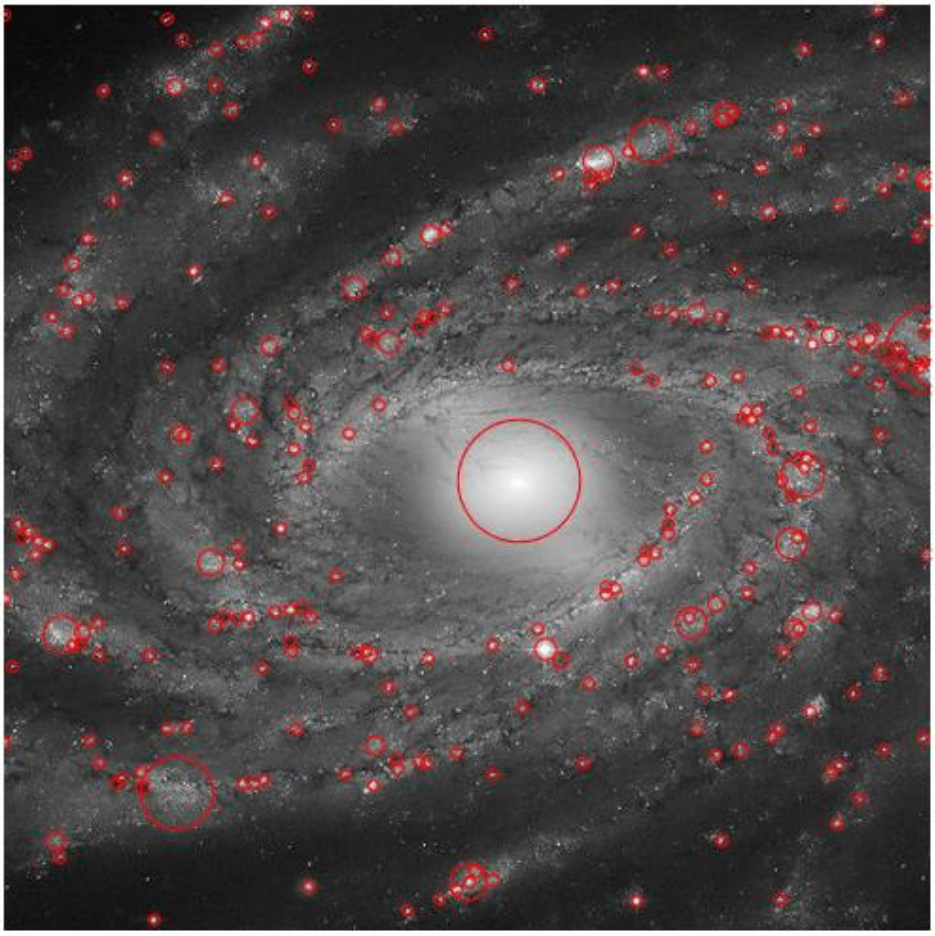
1495 circles



623 circles



281 circles



3115 circles

1478 circles