



ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ

Assignment 2: Blob Detection

Μαυρογιώργης Δημήτρης, AM:2016030016
Κολομβάκη Αφροδίτη, AM:2016030158
Δελατόλας Θάνος, AM:2016030074

May 20, 2021

1. Introduction

Σκοπός του αλγορίθμου scale-invariant feature transform (SIFT) που χρησιμοποιείται ευρέως στη Μηχανική όραση, είναι να εντοπίσουμε και να περιγράψουμε κάποια τοπικά χαρακτηριστικά στις ψηφιακές εικόνες, με απότερο στόχο να αναγνωρίσουμε δύο διαφορετικές λήψεις-περιοχές της ίδιας εικόνας. Για παράδειγμα, με βάση τις ίδιες περιοχές να καταλάβει ο αλγόριθμος ότι πρόκειται για την ίδια εικόνα τραβηγμένη από διαφορετική γωνία. Πιο συγκεκριμένα, στη δεύτερη εργαστηριακή άσκηση κληθήκαμε να υλοποιήσουμε έναν Laplacian blob detector. Ένας blob detector είναι υπεύθυνος να υπολογίζει περιοχές στις ψηφιακές εικόνες οι οποίες έχουν διαφορετικά χαρακτηριστικά, δηλαδή διαφέρουν ως προς κάποιες ιδιότητες. Για παράδειγμα, κάποια χαρακτηριστικά περιοχών είναι η φωτεινότητα της εικόνας ή και το χρώμα της. Επιπλέον, η μέθοδος που χρησιμοποιήσαμε για να υπολογίσουμε τα σημεία ενδιαφέροντος (key points) βασίζονται στα τοπικά ακρότατα, δηλαδή βασίζονται στην εύρεση τοπικών μεγίστων ή ελαχίστων της συνάρτησης.

2. Implementation

- **Generate a Laplacian of Gaussian filter.**

Το Laplacian of Gaussian filter δημιουργήθηκε με τη συνάρτηση fspecial με size $n \times n$ pixels όπου $n = 2 \cdot \text{ceil}(3 \cdot \sigma) + 1$. Το μέγεθος του φίλτρου καθορίζεται από το standard deviation από τη παραπάνω σχέση γιατί:

- Ο όρος $3 \cdot \sigma$ προσεγγίζει το μέγεθος της διαφοράς δυο gaussian φίλτρων
- Πολλαπλασιάζουμε με το 2 εφόσον οι εικόνες μας είναι 2-D.
- Προσθέτουμε 1 για να έχουμε περιττό μέγεθος φίλτρου. Ο λόγος που θέλουμε περιττό μέγεθος του φίλτρου είναι επειδή στην τελική εικόνα μετά τη συνέλιξη έχουμε μια συμμετρία στα pixel κάτι που δε συμβαίνει όταν έχουμε ζυγό αριθμό καθώς τότε θα πρέπει να υπολογίσουμε και κάποια πιθανή distortion.

Γνωρίζουμε πως κανονικοποιούμε την Gaussian πολλαπλασιάζοντας με σ και εφόσον η Laplacian of Gaussian είναι η δεύτερη παράγωγος της Gaussian κανονικοποιείται με τη διασπορά (σ^2).

- **Build a Laplacian scale-space.**

Αρχικά, υλοποιήσαμε την περίπτωση όπου εφαρμόζουμε downsampling στις εικόνες των επιπέδων. Για κάθε επίπεδο $i > 1$ γίνεται downsample της αρχικής εικόνας κατά $\frac{1}{k^{i-1}}$ χρησιμοποιώντας τη συνάρτηση imresize. Στη συνέχεια γίνεται συνέλιξη της downsampled εικόνας με το φίλτρο με τη συνάρτηση imfilter, ενώ αφού υπολογίσουμε το τετράγωνο της συνέλιξης επαναφέρουμε την εικόνα στις αρχικές της διαστάσεις έτσι ώστε στα επόμενα βήματα να μπορέσουμε να συγκρίνουμε ανα pixel τα layers.

Επιπλέον υλοποιήσαμε τη δεύτερη περίπτωση όπου αυξάνουμε το μέγεθος του φίλτρου κρατώντας το μέγεθος της εικόνας σταθερό. Από τη διάλεξη η τυπική απόκλιση για κάθε επίπεδο i υπολογίζεται από τη σχέση $\text{scaled_sigma} = \sigma \cdot k^{i-1}$. Τα υπόλοιπα βήματα που ακολουθήσαμε ήταν όμοια με την πρώτη περίπτωση με μοναδική διαφορά ότι δεν χρειάστηκε upscale στο τετράγωνο της συνέλιξης.

- **Find the extrema in the scale-space.**

Σε αυτό το στάδιο του αλγορίθμου βρίσκουμε τη μέγιστη τιμή ανάμεσα στους 26 γείτονες κάθε pixel, όπου αυτοί οι γείτονες βρίσκονται σε 3 διαδοχικά επίπεδα. Τα μέγιστα σημεία αποτελούν τα key points.

Όσον αφορά τη διαδικασία που ακολουθήθηκε είναι η εξής:

- Με τη συνάρτηση `ordfilt2` βρίσκουμε το μέγιστο pixel μιας γειτονίας 3 x 3 του ίδιου επιπέδου.
- Με ένα `for-loop` υπολογίζουμε το μέγιστο pixel αλλά σε 3 διαδοχικά επίπεδα.

Από το πρώτο βήμα παίρνουμε το μέγιστο 9 pixel και σε συνδιασμό με το δεύτερο βήμα παίρνουμε το μέγιστο 3 x 9 pixel, δηλαδή ενός pixel (i,j) και των 26 γειτόνων του.

- **Display the resulting circles at their characteristic scales.**

Όσον αφορά το συγκεκριμένο βήμα, βρίσκουμε τη θέση στο επίπεδο κάθε κύκλου η οποία είναι η θέση του αντίστοιχου maxima σημείου και υπολογίζουμε την ακτίνα του με βάση τη σχέση: $\sqrt{(2) \cdot \sigma \cdot k^{i-1}}$

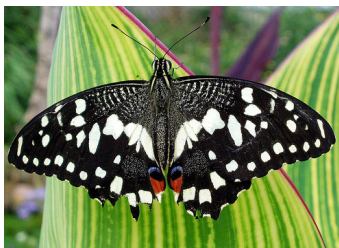
3. Επιλογή παραμέτρων

Έπειτα απο δοκιμές καταλήξαμε στις τιμές:

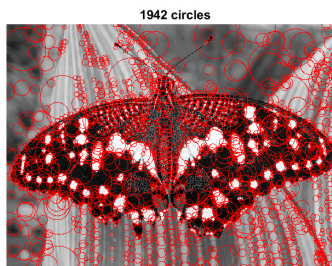
- $\sigma = 2$
- `threshold=0.005`
- `layers=15`

Επιπλέον, βρήκαμε πως το κατάλληλο k είναι $k = \sqrt{\sqrt{2}} \approx 1.2$. Όσον αφορά την επιλογή του σ , δεν επιλέξαμε τιμή 1, επειδή ναι μεν τα αποτελέσματα δεν ήταν και τόσο βέλτιστα αλλά ο χρόνος εκτέλεσης του αλγορίθμου ήταν πολύ μικρότερος. Επιπλέον, δεν χρησιμοποιήσαμε κάποια μεγαλύτερη τιμή σ καθώς ο χρόνος εκτέλεσης αυξανόταν αρκετά, ενώ παράλληλα δεν υπήρχε κάποια σημαντική βελτίωση των αποτελεσμάτων. Τέλος, για το `threshold` δεν επιλέξαμε κάποια μεγαλύτερη τιμή, γιατί δεν παρατηρούσαμε σημαντικές αλλαγές ως προς συνολικό αριθμό κύκλων.

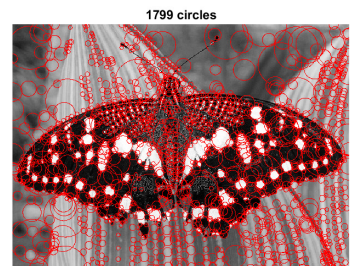
4. Σύγκριση



«i» Original Image



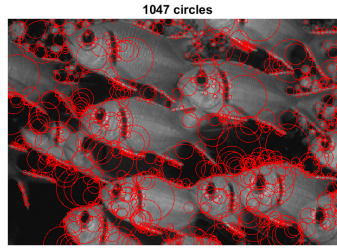
«ii» Image Downsampling



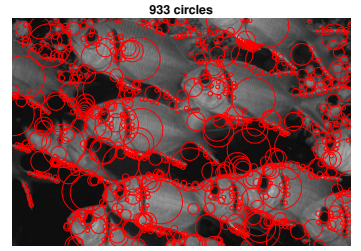
«iii» Filter Resizing



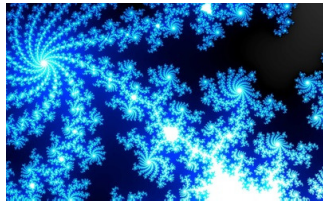
«i» Original Image



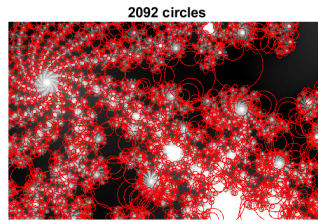
«ii» Image Downsampling



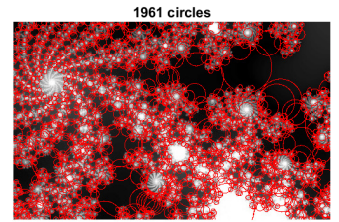
«iii» Filter Resizing



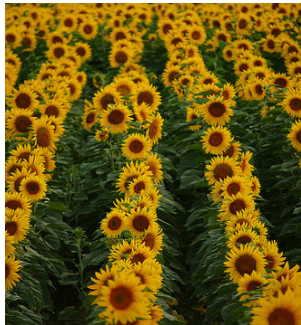
«i» Original Image



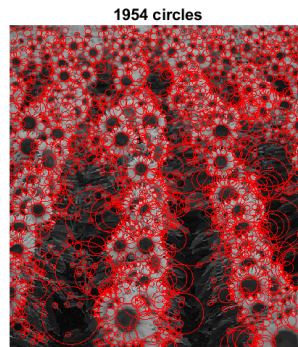
«ii» Image Downsampling



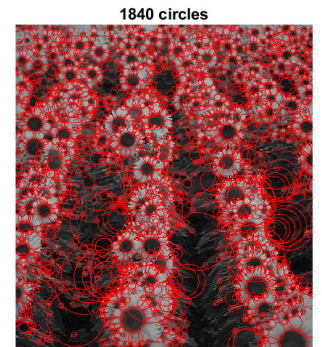
«iii» Filter Resizing



«i» Original Image



«ii» Image Downsampling



«iii» Filter Resizing

	Butterfly	Fishes	Snowflakes	Sunflowers
Image Downsampling	0.362362	0.393882	0.221580	0.265076
Filter Resizing	1.024760	1.042106	0.706753	0.749725

Πίνακας: 1. Χρόνοι εκτέλεσης σε seconds

Οι κύριες διαφορές μεταξύ των δυο υλοποιήσεων είναι η χρονική πολυπλοκότητα και η διαφορά στον αριθμό των συνολικών key points.

Πιο συγκεκριμένα, η χρονική πολυπλοκότητα είναι σημαντικά μικρότερη στην περίπτωση που κάνουμε down-sample την εικόνα και οφείλεται στο γεγονός πως σε αυτή τη μέθοδο σε κάθε scale μειώνεται το size της εικόνας, ενώ παραμένει σταθερό το size του φίλτρου με αποτέλεσμα οι πράξεις που απαιτούνται για τη συνέλιξη να μειώνονται σε κάθε στάδιο.

Αντιθέτως, στο Filter Resizing η εικόνα παραμένει σταθερή ενώ το μέγεθος του φίλτρου αυξάνει σε κάθε scale και κατ' επέκταση το μέγεθος της συνέλιξης αυξάνει σταδιακά. Επιπλέον, τα υπόλοιπα βήματα του αλγορίθμου SIFT για τη εύρεση των μεγίστων και το σχεδιασμό των κύκλων είναι κοινά και στις δυο περιπτώσεις.

Επιπρόσθετα, το πλήθος των κύκλων ανάμεσα στις δύο μεθόδους που υλοποιήσαμε διαφέρει. Ειδικότερα, παρατηρούμε ελαφρώς περισσότερους κύκλους στην περίπτωση που κάνουμε downsample την εικόνα κατά την κατασκευή του scale space. Παράλληλα, βλέπουμε ότι στην δεύτερη μέθοδο έχουμε κύκλους με μεγαλύτερη ακτίνα στη θέση κάποιων κύκλων με μικρότερη ακτίνα. Το γεγονός αυτό συμβαίνει καθώς όσο αυξάνουμε το σ έχουμε αύξηση του μεγέθους της Γκαουσιανής καμπάνας.

Τέλος, με βάση τη χρονική πολυπλοκότητα και τον αριθμό των key points, θα επιλέγαμε τη μέθοδο που κάνει image downsampling.