

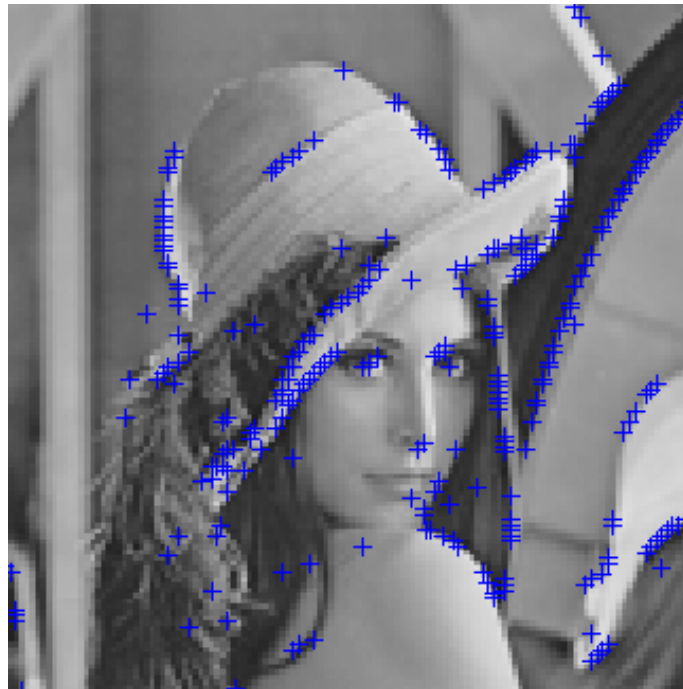
Ψηφιακή Επεξεργασία Εικόνας

-Εργασία 3-

Scale Invariant Feature Transform (SIFT)

Ομάδα Κατανόησης Πολυμέσων
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Άνοιξη 2021



Η γνωστή Lena Forsén με τα εντοπισμένα salient keypoints κατά SIFT.

Εισαγωγικά

Στην εργασία αυτή θα υλοποιήσετε τα ακόλουθα:

1. Υπολογισμός των space scales και των Difference-of-Gaussians (DoGs)
2. Εντοπισμός των salient keypoints
(α') Φιλτράρισμα των salient keypoints

Τα παραπάνω βήματα αποτελούν τον SIFT **detector**. Αξίζει να σημειωθεί πως τα εντοπισμένα salient keypoints μπορούν στην συνέχεια να χρησιμοποιηθούν σαν είσοδο στον SIFT **descriptor**. Ο συνδιασμός των SIFT detector/descriptor επιτρέπει την εξαγωγή εύρωστων χαρακτηριστικών (features) και κατά επέκταση την ανάπτυξη εφαρμογών όπως π.χ.,

image stitching για την κατασκευή πανοραμικών φωτογραφιών.

Μαζί με την εκφώνηση θα βρείτε και το βοηθητικό MATLAB αρχείο `dip_hw_3.mat` το οποίο περιλαμβάνει τις δύο εικόνες εισόδου (`mountains` και `roofs`) που θα χρησιμοποιήσετε στα πειράματά σας. Οι δύο εικόνες εισόδου είναι grayscale, κανονικοποιημένες, με διαστάσεις 128×128 η κάθε μία. Για επιπλέον μελέτη, σας δίνονται οι δύο δημοσιεύσεις του David G. Lowe (`lowe1999` και `lowe2004`) που περιγράφουν τον detector και τον descriptor σε μεγαλύτερο βάθος.

1 Gaussian Space Scale

Υλοποιήστε την ρουτίνα `my2DGaussianFilter` η οποία κατασκευάζει και επιστρέφει ένα 2D Gaussian φίλτρο σύμφωνα με την Εξίσωση 1.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (1)$$

```
1 function myGFilter = my2DGaussianFilter(K, sigma)
```

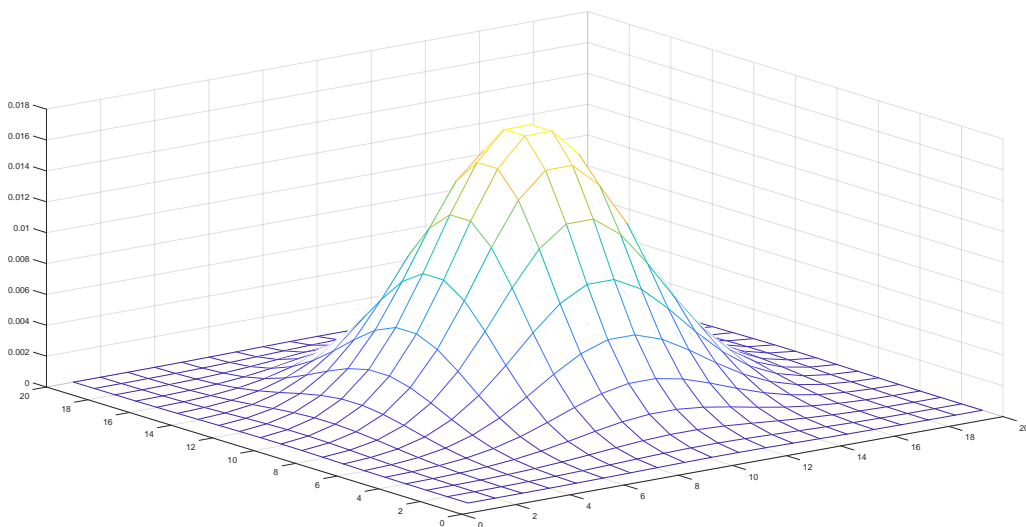
όπου

`K`: Το μέγεθος του φίλτρου.

`sigma`: Η τυπική απόκλιση της Gaussian κατανομής (σ).

`myGFilter`: Το $K \times K$ Gaussian φίλτρο.

Θεωρήστε πως το φίλτρο έχει μέση τιμή (είναι κεντραρισμένο γύρω από) το $(0, 0)$. Επιπλέον, το φίλτρο σας θα πρέπει να είναι *κανονικοποιημένο* - δηλαδή να έχει τιμές στο διάστημα $[0, 1]$. Για να το πετύχετε αυτό, πρέπει να διαιρέσετε κάθε τιμή του φίλτρου με το άθροισμα των όλων των τιμών του. Ένα παράδειγμα¹ Gaussian φίλτρου μπορείτε να βρείτε στην Εικόνα 2.



Σχήμα 2: Ένα 20×20 ($K = 20$) Gaussian φίλτρο δύο διαστάσεων με $\sigma = 3$.

Στην συνέχεια θα πρέπει να κατασκευάσετε το Gaussian Scale Space. Η διαδικασία ξεκινάει ως εξής. Έστω $I(x, y)$ μία *κανονικοποιημένη* (τιμές στο $[0, 1]$), grayscale εικόνα εισόδου. Εφαρμόζοντας, μέσω 2D convolution², το Gaussian φίλτρο $G(x, y, \sigma)$ πέρνουμε το scale space της εικόνας $L_1(x, y, \sigma)$ (Εξίσωση 2).

$$L_1(x, y, \sigma) = I(x, y) * G(x, y, \sigma) \quad (2)$$

¹Tip: χρησιμοποιήστε την MATLAB εντολή `mesh(myGFilter)` για να προβάλετε το φίλτρο σας

²Tip: χρησιμοποιήστε την MATLAB εντολή `conv2(img, aGaussianFilter, 'same')`

Στην πράξη, αυτό που κάνει η Εξίσωση 2 είναι ένα low-pass-filtering ή αλλιώς blurring στην εικόνα $I(x, y)$.

Η διαδικασία συνεχίζει χρησιμοποιώντας την blurred εικόνα $L_1(x, y, \sigma)$ σαν εικόνα εισόδου στην Εξίσωση 2 με $\sigma' = k\sigma$ όπου και παράγεται η εικόνα $L_1(x, y, \sigma')$. Για τις ανάγκες της εργασίας θα χρησιμοποιήσουμε $k = \sqrt{2}$. Η διαδικασία τερματίζει μετά από έναν αριθμό επαναλήψεων που ορίζεται από την μεταβλητή `levels`. Για παράδειγμα αν `levels = 4` τότε έχουν παραχθεί τα space scales:

- $L_1(x, y, 2\sqrt{2}\sigma)$
- $L_1(x, y, 2\sigma)$
- $L_1(x, y, \sqrt{2}\sigma)$
- $L_1(x, y, \sigma)$

με σειρά δημιουργίας από κάτω προς τα πάνω (δηλαδή πρώτα παράγεται το $L_1(x, y, \sigma)$, μετά το $L_1(x, y, \sqrt{2}\sigma)$, κ.ο.κ.) και $k = \sqrt{2}$. Τα παραπάνω τέσσερα space scales ανήκουν στην πρώτη *octave*.

Συνεχίζοντας στην δεύτερη *octave*, χρησιμοποιούμε το space scale που έχει τυπική απόκλιση (σ) διπλάσια του αρχικού (άρα το $L_1(x, y, 2\sigma)$, δηλαδή το τρίτο κατά σειρά δημιουργίας), το υποδειγματοληπτούμε με scale factor 2^3 και επαναλαμβάνουμε την διαδικασία Gaussian blurring όσες φορές υποδεικνύει η μεταβλητή `levels`. Μιας και από το προηγούμενο παράδειγμα `levels = 4`, η δεύτερη *octave* περιλαμβάνει τα:

- $L_2(x, y, 8\sigma)$
- $L_2(x, y, 4\sqrt{2}\sigma)$
- $L_2(x, y, 4\sigma)$
- $L_2(x, y, 2\sqrt{2}\sigma)$

με σειρά δημιουργίας από κάτω προς τα πάνω. Η μεταβλητή `octaves` προσδιορίζει τον συνολικό αριθμό των *octaves*.

Για να εντοπίσουμε τα stable keypoints πρέπει πρώτα να υπολογίσουμε difference-of-Gaussian (ή αλλιώς DoG) $D(x, y, \sigma)$, τα οποία είναι η διαφορά δύο γειτονικών scales στο ίδιο *octave* (Εξίσωση 3).

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3)$$

Το Σχήμα 3 απεικονίζει ένα παράδειγμα υπολογισμού των DoGs.

Υλοποιήστε λοιπόν την ρουτίνα `myDoGs` η οποία υπολογίζει τα space scales και τα DoGs.

```
1 function [spacescales, DoGs] = myDoGs(img, K, sigma, levels, octaves)
```

όπου

`img`: Η κανονικοποιημένη, grayscale εικόνα εισόδου $I(x, y)$.

`K`: Το μέγεθος του Gaussian φίλτρου.

`sigma`: Η τυπική απόκλιση της Gaussian κατανομής σ .

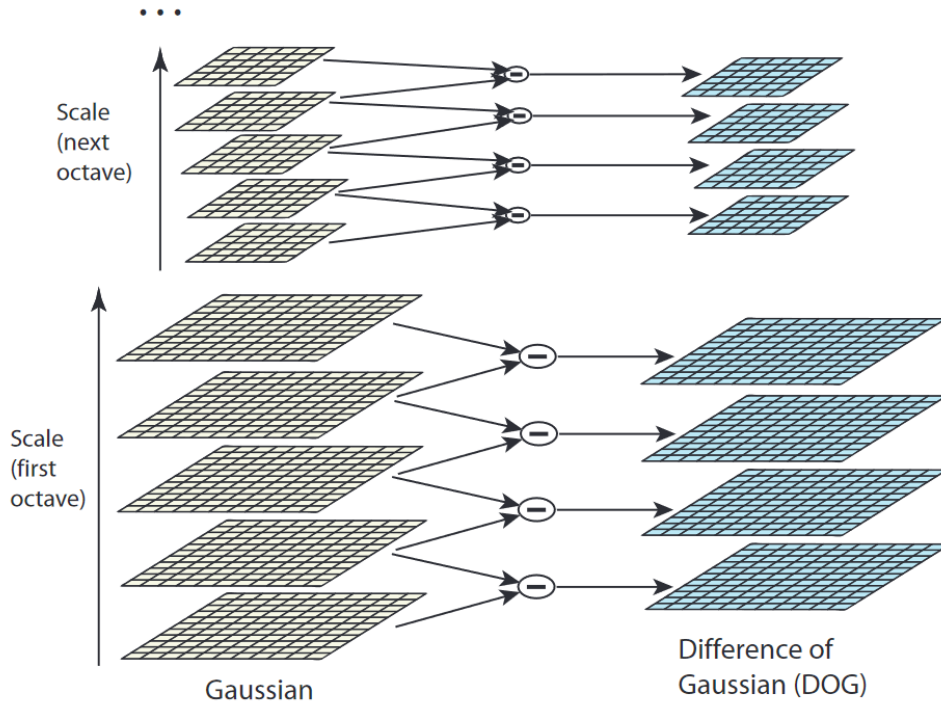
`level`: Ο συνολικός αριθμός των `levels` σε κάθε *octave*.

`octaves`: Ο συνολικός αριθμός των *octaves*.

`spacescales`: Ένα cell array με διαστάσεις $1 \times \text{octaves}$. Κάθε στοιχείο του cell είναι ένας πίνακας με διαστάσεις $N_i \times M_i \times \text{levels}$ που περιλαμβάνει όλα τα space scales του i -οστού *octave* με διαστάσεις $N_i \times M_i$.

`DoGs`: Ένα cell array με διαστάσεις $1 \times \text{octaves}$. Κάθε στοιχείο του cell είναι ένας πίνακας με διαστάσεις $N_i \times M_i \times (\text{levels} - 1)$ που περιλαμβάνει όλα τα DoGs του i -οστού *octave* με διαστάσεις $N_i \times M_i$.

³Tip: χρησιμοποιήστε την MATLAB εντολή `imresize(img, 0.5)`



Σχήμα 3: Παράδειγμα υπολογισμού των DoGs σε 2 *octaves* και *levels* = 5.

1.1 Demo 1

Υλοποιήστε το MATLAB script `demo1.m` στο οποίο να υπολογίζετε τα DoGs που προκύπτουν από την εικόνα εισόδου και $\sigma = \sqrt{2}$, $K = 7$, *levels* = 5, και *octaves* = 3. Επαναλάβετε το πείραμα για $\sigma = \sqrt{2}$, $K = 7$, *levels* = 3, και *octaves* = 7.

Να χρησιμοποιήσετε και τις δύο εικόνες εισόδου στα πειράματά σας.

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας με την χρήση MATLAB `subplot`. Τι παρατηρείτε;

2 Εντοπισμός των salient keypoints

Μετά των υπολογισμό των DoGs σειρά έχει ο εντοπισμός των salient keypoints.

Η διαδικασία εντοπισμού έχει ως εξής. Κάθε pixel των DoGs (εκτός από τα pixels που ανήκουν στο περίγραμμα) συγκρίνεται με τους $3^2 - 1$ γείτονες του καθώς και τους 3^2 γείτονες του ανώτερου DoG και τους 3^2 γείτονες του κατώτερου DoG (σύνολο $3^3 - 1$ γείτονες). Το σημείο αποτελεί *keypoint* αν είναι μεγαλύτερο ή μικρότερο από όλους τους γείτονές του, αν είναι δηλαδή τοπικό ακρότατο σε αυτόν τον 3D χώρο. Το Σχήμα 4 δείχνει την διαδικασία. Η σύγκριση είναι εφικτή μόνο μεταξύ των DoGs του ίδιου *octave*. Επιπλέον, δεν χρειάζεται να ελέξετε pixels τα οποία δεν έχουν όλους τους 26 γείτονες (αν δηλαδή ανήκουν στο πρώτο ή το τελευταίο DoG του *octave*).

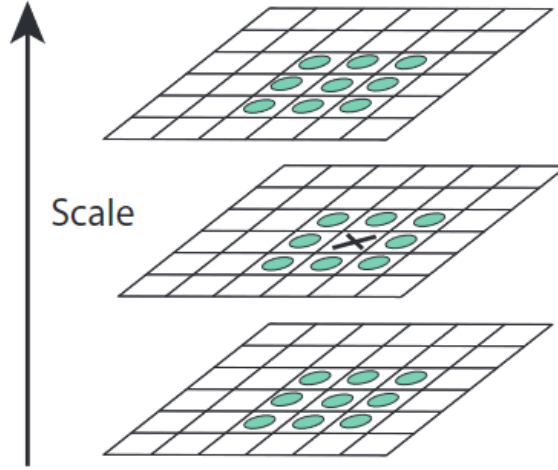
Υλοποιήστε λοιπόν την ρουτίνα `myKeypoints` η οποία υπολογίζει τα αρχικά keypoints με την χρήση των DoGs.

```
1 function keypoints = myKeypoints(DoGs)
```

όπου

DoGs: Ένα cell array με διαστάσεις $1 \times \text{octaves}$. Κάθε στοιχείο του cell είναι ένας πίνακας με διαστάσεις $N_i \times M_i \times (\text{levels} - 1)$ που περιλαμβάνει όλα τα DoGs του i -οστού *octave* με διαστάσεις $N_i \times M_i$.

keypoints: Μία λίστα με τα εντοπισμένα keypoints μήκους V . Κάθε ένα από τα V στοιχεία της λίστας είναι ένα tuple της μορφής (o, s, m, n) , όπου o και s το *octave* και *space scale* στο οποίο ανήκει το εντοπισμένο *keypoint* και m, n οι



Σχήμα 4: Σύγκριση του pixel που συμβολίζεται με μαύρο X με τους $3^2 - 1$ γείτονές του (πράσινοι κύκλοι).

συντεταγμένες του.

2.1 Απόρριψη των low contrasted keypoints

Σειρά έχει η απόρριψη των low contrasted keypoints. Έστω $w_{s,m,n}^o$ η τιμή της φωτεινότητας του pixel το οποίο είναι keypoint στο octave o , space scale s με συντεταγμένες m, n . Αν $|w_{s,m,n}^o| \leq p \times t$ τότε το keypoint απορρίπτεται, διαφορετικά το προσθέτουμε στην λίστα με τα φιλτραρισμένα keypoints. Η παράμετρος t είναι ένα κατώφλι και η παράμετρος p ένα ποσοστό που δείχνει πόσο συντηρητικοί θέλουμε να είμαστε σχετικά με την απόρριψη. Το κατώφλι t υπολογίζεται σύμφωνα με την Εξίσωση 4.

$$t = \frac{2^{1/n_{spo}} - 1}{2^{1/3} - 1} 0.015 \quad (4)$$

Όπου n_{spo} , ο αριθμός των levels -2 . Για $n_{spo} = 3$ το κατώφλι είναι 0.015.

Υλοποιήστε την ρουτίνα `discardLowContrasted` η οποία απορρίπτει τα keypoints με χαμηλή αντίθεση.

```
1 function keypointsHighC = discardLowContrasted(DoGs, keypoints, t, p)
```

όπου

keypoints: Η λίστα με τα εντοπισμένα keypoints μήκους V . Κάθε ένα από τα V στοιχεία της λίστας είναι ένα tuple της μορφής (o, s, m, n) , όπου o και s το octave και space scale στο οποίο ανήκει το εντοπισμένο keypoint και m, n οι συντεταγμένες του.

t: Το κατώφλι απόρριψης που υπολογίζεται σύμφωνα με την Εξίσωση 4.

p: Ποσοστό που δείχνει πόσο συντηρητικοί θέλουμε να είμαστε (συνήθως ≈ 0.8).

keypointsHighC: Η λίστα με τα φιλτραρισμένα keypoints, μήκους $V' < V$.

2.2 Demo 2

Υλοποιήστε το MATLAB script `demo2.m` στο οποίο να υπολογίζετε αρχικά τα keypoints από το πρώτο πείραμα του `demo1.m` (με $\sigma = \sqrt{2}$, $K = 7$, `levels` = 5, και `octaves` = 3). Παρουσιάστε τα (αφιλτράριστα) keypoints πάνω στα DoGs τα οποία ανήκουν. Στην συνέχεια φιλτράρετε τα low contrasted keypoints και παρουσιάστε τα πάνω στα DoGs.

Πειραματιστείτε με διαφορετικές τιμές της παραμέτρου p .

Όπως και στο πρώτο demo, να χρησιμοποιήσετε και τις δύο εικόνες εισόδου στα πειράματά σας.

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας με την χρήση MATLAB subplot. Τι παρατηρείτε;

Αξιολόγηση & παραδοτέα

Κατά την υποβολή της εργασίας θα πρέπει να παραδώσετε μια αναφορά και τα αρχεία με τις συναρτήσεις:

- my2DGaussianFilter.m
- myDoGs.m
- myKeypoints.m
- discardLowContrasted.m

Επιπλέον, θα πρέπει να παραδώσετε και τα scripts demo1.m και demo2.m, τα οποία θα εκτελούνται **χωρίς ορίσματα** και θα παρουσιάζουν τα ζητούμενα των ενότητων 1.1 και 2.2. Τέλος, στην αναφορά σας θα πρέπει επίσης να παρουσιάσετε όποιες σχεδιαστικές επιλογές έχετε κάνει.

Σχετικά με την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Η αναφορά θα πρέπει να επιδεικνύει την ορθή λειτουργία του κώδικά σας στις εικόνες που σας δίνονται περιλαμβανομένων των αποτελεσμάτων που προκύπτουν από τα demo.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλήσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για *οποιαδήποτε* είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για τις εικόνες που σας δίνονται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα, καθώς και να τηρούνται τα ακόλουθα:

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι AEM.zip, όπου AEM είναι τα τέσσερα ψηφία του Α.Ε.Μ. του φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα Matlab και το αρχείο report.pdf το οποίο θα είναι η αναφορά της εργασίας. **Κάθε συνάρτηση θα πρέπει να είναι ένα ξεχωριστό αρχείο .m με όνομα ίδιο με αυτό της συνάρτησης που υλοποιεί.**
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα report.pdf.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη m.
- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε τις εικόνες που σας δίνονται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ “Thumbs.db”, “DS_Store”, “.directory”.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “\$”, “%” κλπ.