



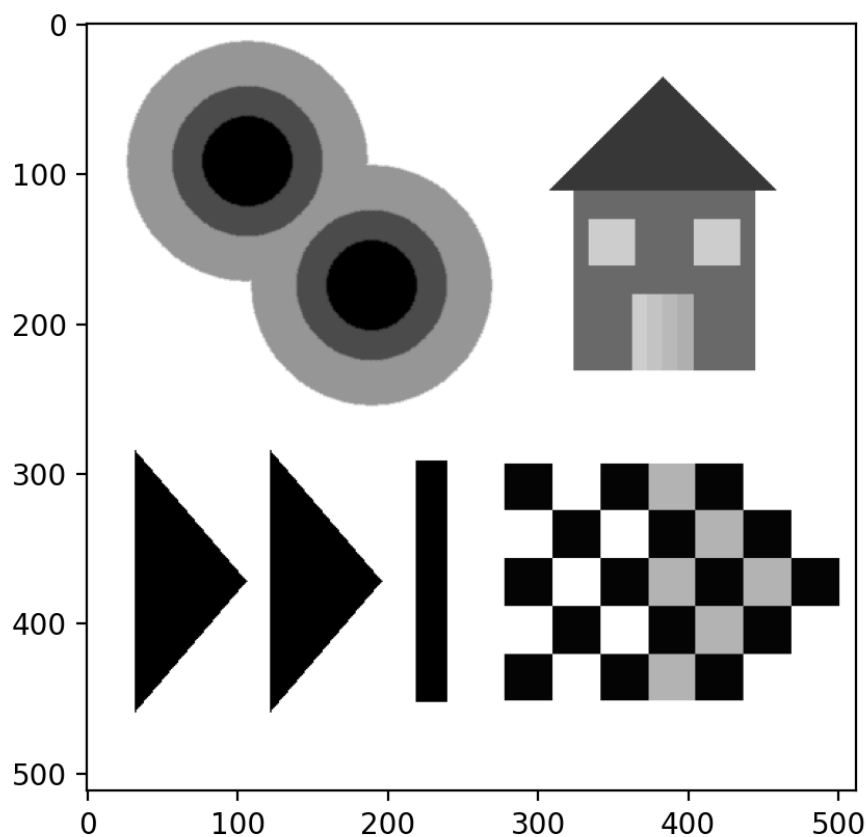
Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών &
Μηχανικών Υπολογιστών
Ροή Σ: Όραση Υπολογιστών
(8^ο Εξάμηνο)
Εργαστηριακή Άσκηση 1

Δημήτρης Μπακάλης 03118163

Μέρος 1

1.1. Δημιουργία Εικόνων Εισόδου:

Αρχικά θα επεξεργαστούμε την τεχνητή εικόνα edgetest 22.png, η οποία φαίνεται παρακάτω:

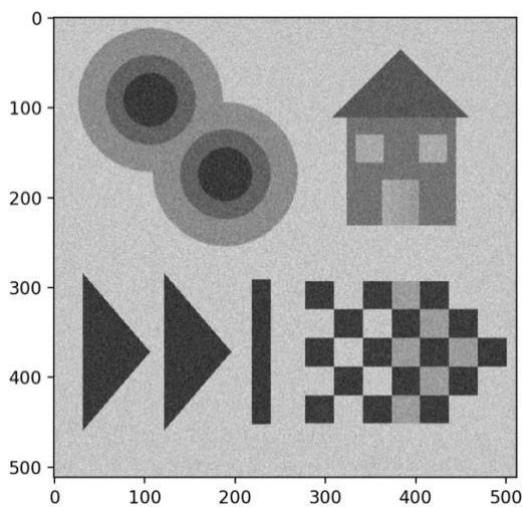


Συγκεκριμένα, θα προσπαθήσουμε να ανιχνεύσουμε τις ακμές της εικόνας, αφού, πρώτα, προσθέσουμε σε αυτή λευκό θόρυβο (μέση τιμή 0 και τυπική απόκλιση που υπολογίζεται από συγκεκριμένες τιμές του PSNR). Από τον τύπο (1) της εκφώνησης προκύπτει ο παρακάτω τύπος για τον υπολογισμό της τυπικής απόκλισης:

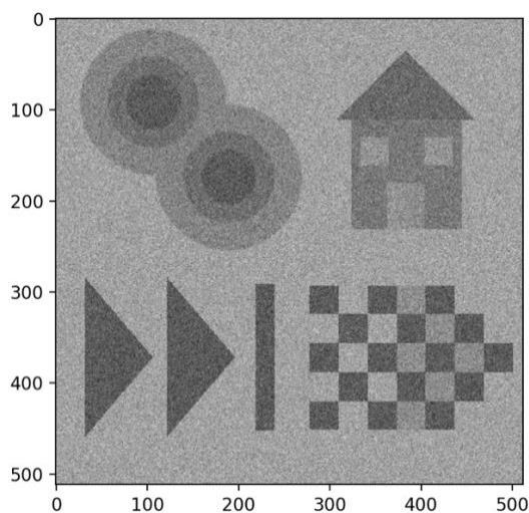
$$\sigma = \frac{I_{max} - I_{min}}{10^{PSNR/20}}$$

Επομένως, αφού στη δική μας περίπτωση έχουμε $I_{max}=1$, $I_{min}=0$ θα ισχύει:

i) Για PSNR = 20 dB: $\sigma = 0.1$



ii) Για PSNR = 10 dB: $\sigma = \frac{1}{\sqrt{10}} \approx 0.316$

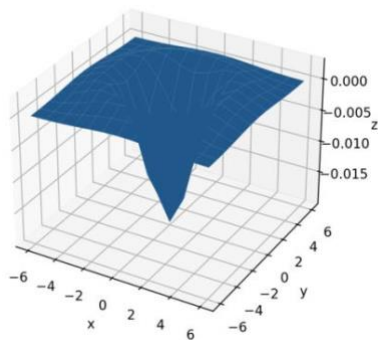


Στη συνέχεια, θα εξομαλύνουμε την εικόνα με γκαουσιανό φιλτράρισμα και θα επιχειρήσουμε να βρούμε τα zero crossings της λαπλασιανής της εξομαλυμένης εικόνας. Τα zero crossings, που θα εξάγουμε, αφορούν τις περιοχές, στις οποίες αλλάζει απότομα η φωτεινότητα στην εικόνα, και, επομένως, σηματοδοτούν τις ακμές της.

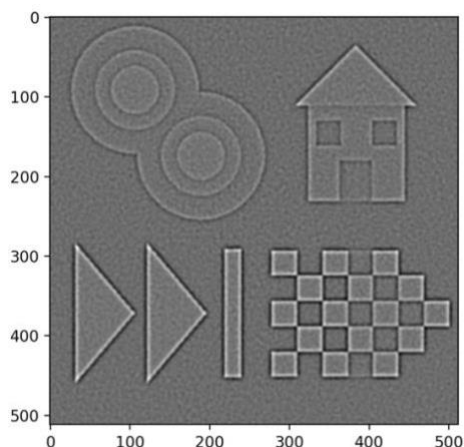
Αρχικά, θα προσεγγίσουμε τη λαπλασιανή της εξομαλυμένης εικόνας, τόσο με γραμμικά, όσο και με μη γραμμικά φίλτρα. Στα γραμμικά φίλτρα θα ισχύει η σχέση $L1 = \nabla^2 (G * I)$, το οποίο συνεπάγεται διπλή παραγωγή (άρα και διπλό διακριτό φιλτράρισμα), με αποτέλεσμα να έχουμε αρκετό θόρυβο στην έξοδο. Προκείμενου να είμαστε ακριβέστεροι, θα εκμεταλλευτούμε την αντιμεταθετική ιδιότητα της συνέλιξης και θα φιλτράρουμε μία, μόνο, φορά με τη λαπλασιανή της γκαουσιανής, η οποία θα δημιουργηθεί από τον ακόλουθο τύπο:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Η γραφική της παράσταση για $\sigma = 2$ είναι η παρακάτω:



Παρατίθεται η εικόνα που προκύπτει με φιλτράρισμα της αρχικής εικόνας με αυτό το φίλτρο:



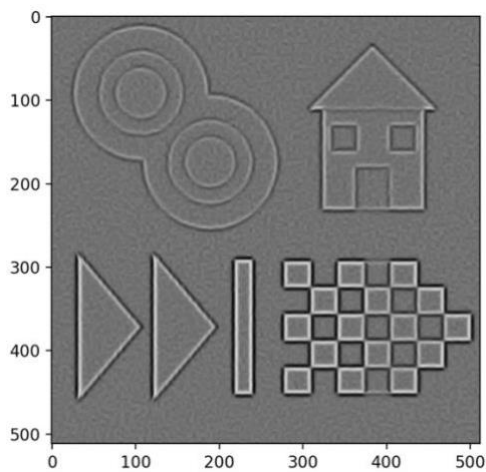
Με χρήση μη γραμμικών μορφολογικών φίλτρων, η λαπλασιανή της εξομαλυμένης εικόνας παράγεται ως εξής:

$$L2 = I\sigma \oplus B + I\sigma \ominus B - 2I\sigma, \text{ με } B \text{ έναν διακριτό δίσκο } 3 \times 3.$$

Ο παραπάνω τύπος αποτελεί απλούστευση ($r=1$) του παρακάτω τύπου, ο οποίος ορίζει ακριβώς τη λαπλασιανή μιας εικόνας:

$$M^2(I\sigma) = \lim_{r \rightarrow 0} \frac{(I\sigma \oplus rB - I\sigma) - (I\sigma - I\sigma \ominus rB)}{2r}$$

Με την εφαρμογή του μη γραμμικού φίλτρου λαμβάνουμε την παρακάτω εικόνα:



Για να επιλέξουμε τα zero crossings των παραπάνω εικόνων θα εφαρμόσουμε τον εξής, ίδιο και για τα δύο φίλτρα, αλγόριθμο. Αρχικά, θα δημιουργήσουμε τη δυαδική εικόνα πρόσημου της L . Στη συνέχεια, βρίσκουμε το περίγραμμα της εικόνας αξιοποιώντας τον τύπο:

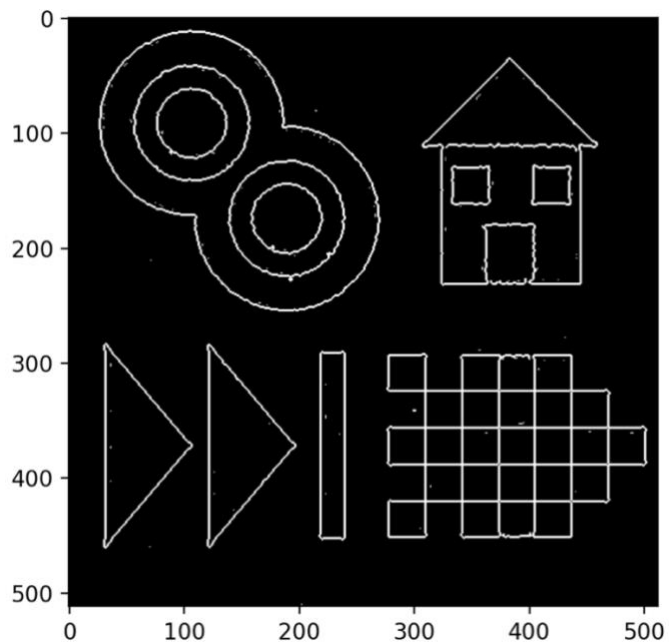
$$Y = (X \oplus B) - (X \ominus B) \approx \partial X$$

Επομένως, θα επιλέξουμε ως τελικά zero crossings ένα υποσύνολο των στοιχείων που ισχύει $Y[i, j] = 1$, καθώς θέλουμε να δώσουμε έμφαση σε αυτά με έντονη αλλαγή φωτεινότητας, άρα και μεγάλο $\|\nabla I\sigma\|$. Το δεύτερο κριτήριο, λοιπόν, που θα πρέπει να ικανοποιεί ένα σημείο, για να θεωρηθεί ότι ανήκει στα zero crossings είναι να ισχύει η παρακάτω ανισότητα για μία παράμετρο θ_{edge} που ορίζουμε εμείς:

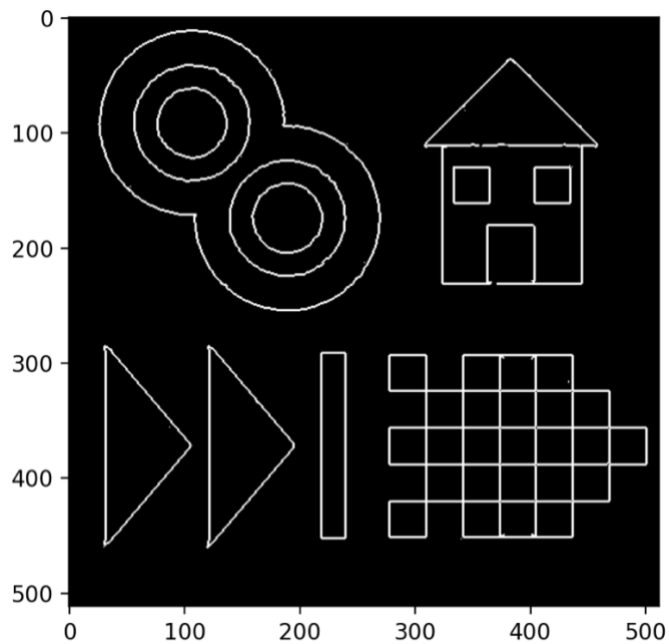
$$\|\nabla I\sigma[i, j]\| > \theta_{\text{edge}} \cdot \max_{x,y} \|\nabla I\sigma\|$$

Η τελική εικόνα με παράμετρο $\theta_{\text{edge}}=0.1$ είναι:

(i) Με χρήση γραμμικού φίλτρου ($\sigma=2$):

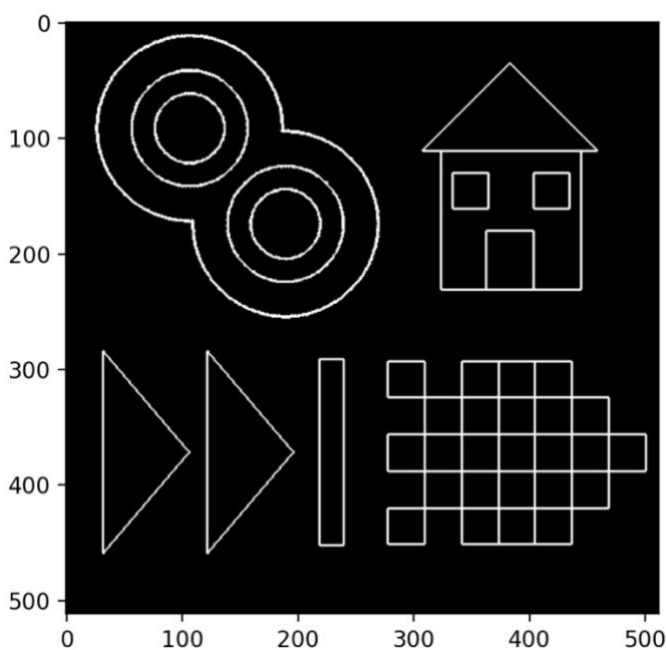


(ii) Με χρήση μη γραμμικού μορφολογικού φίλτρου:



Σημείωση: Η εικόνα που επεξεργάζεται, για να προκύψουν αυτές οι εικόνες (λαπλασιανές και δυαδικές) είναι αυτή με θόρυβο με $\text{PSNR} = 20\text{dB}$.

Τέλος, θα διαμορφώσουμε μια μετρική για την επίδοση του αλγορίθμου μας. Δεδομένου πως έχουμε μία τεχνητή εικόνα μπορούμε να υπολογίσουμε ακριβώς τις ακμές της εικόνας, αυτής με τον τύπο που χρησιμοποιήθηκε παραπάνω, για να εξάγουμε το περίγραμμα (προφανώς ο λόγος που αρκεί αυτό είναι και ότι θα το εφαρμόσουμε στην αρχική εικόνα χωρίς θόρυβο, σε αντίθεση με αυτή που επεξεργαζόμασταν σε όλα τα βήματα ως τώρα). Έπειτα, θα μετατρέψουμε την εικόνα σε δυαδική με κατωφλιοποίηση. Έτσι, η τελική μετρική θα προκύψει από τη μέση τιμή του precision και του recall μεταξύ των αποτελεσμάτων της εικόνας με θόρυβο και αυτής χωρίς θόρυβο. Οι ακμές που ανιχνεύτηκαν στην αποθορυβοποιημένη εικόνα (με κατώφλι $\theta_{\text{realedge}}=0.1$) φαίνονται παρακάτω:



Οι τελικές τιμές με τις παραμέτρους που έχουν χρησιμοποιηθεί στα προηγούμενα παραδείγματα είναι οι εξής:

Με μη γραμμικά μορφολογικά φίλτρα:

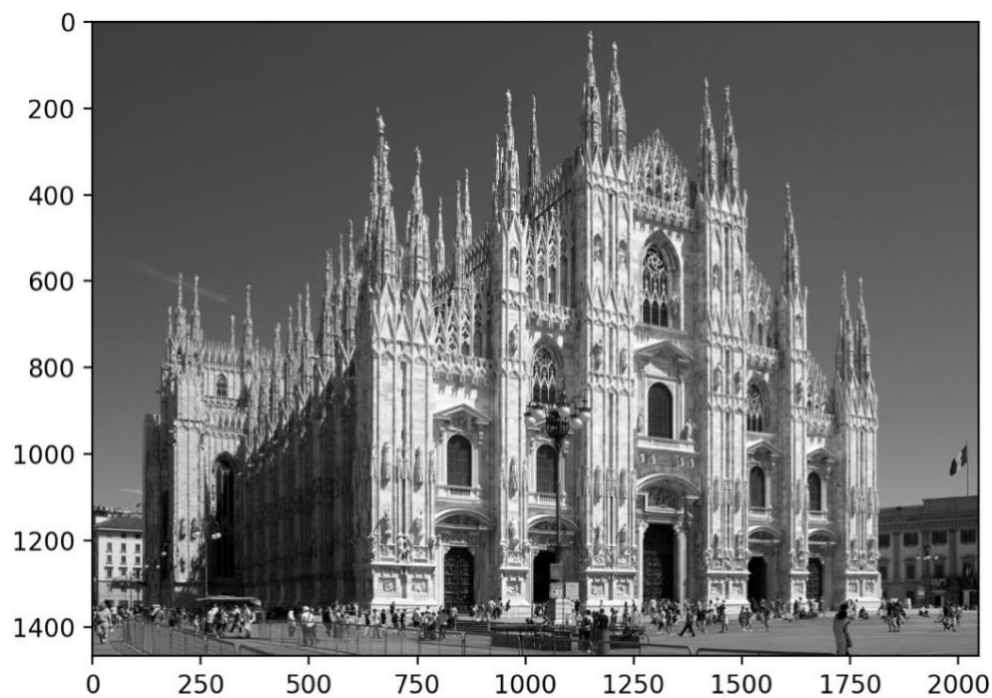
- Για PSNR = 20 dB ($\theta_{\text{edge}} = 0.15$):
Precision = 0.9098
Recall = 0.97762
C = 0.9437
- Για PSNR = 10 dB ($\theta_{\text{edge}} = 0.25$):
Precision = 0.7398
Recall = 0.8683
C = 0.8041

Με γραμμικά φίλτρα ($\sigma = 2$):

- Για PSNR = 20 dB ($\theta_{\text{edge}} = 0.15$):
Precision = 0.8814
Recall = 0.9472
C = 0.9143
- Για PSNR = 10 dB ($\theta_{\text{edge}} = 0.25$):
Precision = 0.6842
Recall = 0.8344
C = 0.7593

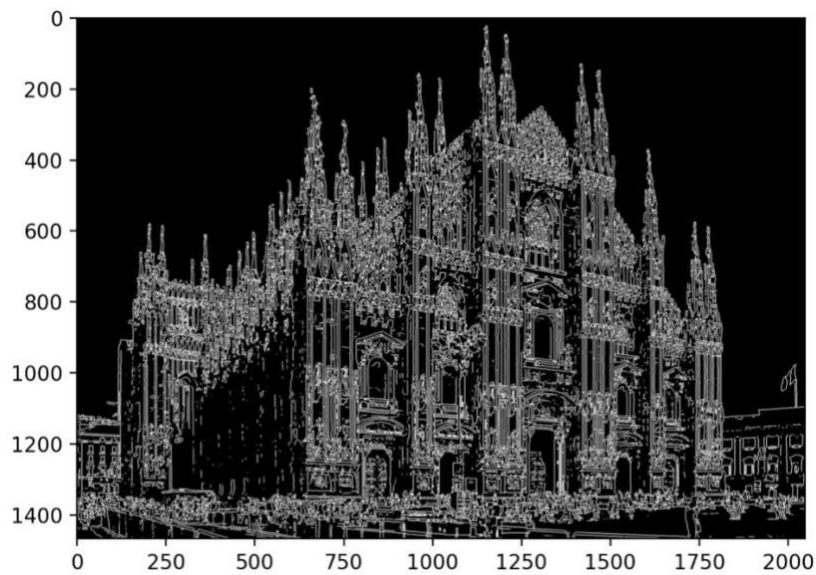
Επομένως, παρατηρούμε πως τα μη γραμμικά μορφολογικά φίλτρα έχουν καλύτερη επίδοση στη συγκεκριμένη περίπτωση.

Αντίστοιχα, επαναλαμβάνουμε την παραπάνω διαδικασία και στην εικόνα `duomo_edges.jpg`:



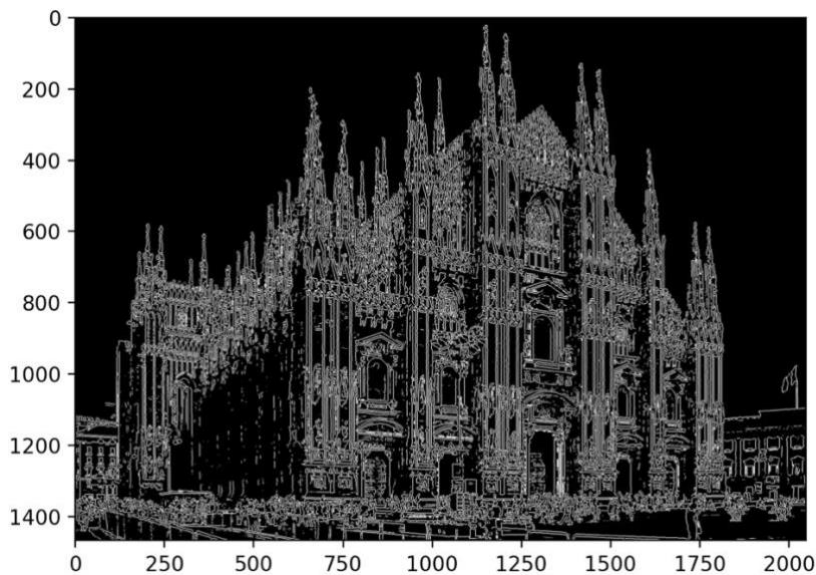
Η τελική εικόνα με παράμετρο $\theta_{\text{edge}}=0.1$ είναι:

Με μη γραμμικά μορφολογικά φίλτρα:



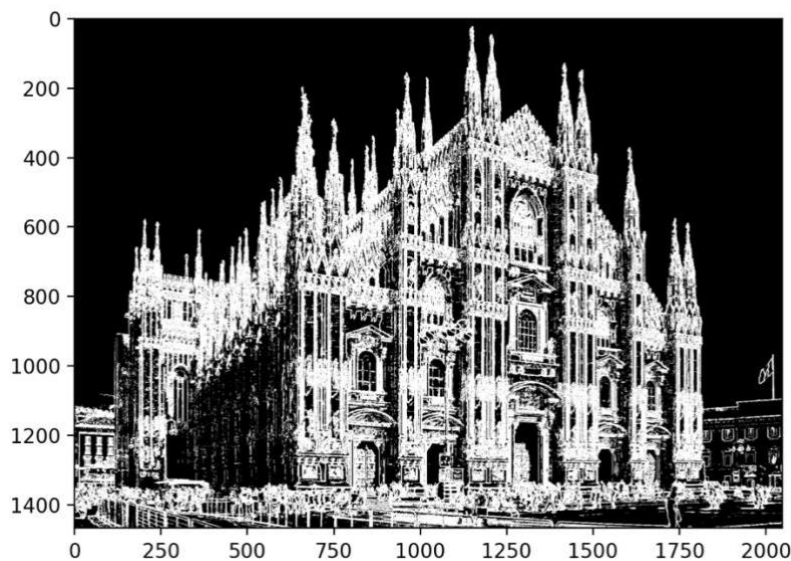
- Precision = 0.4308
- Recall = 0.8960
- C = 0.6634

Με γραμμικά φίλτρα ($\sigma = 2$):



- Precision = 0.3961
- Recall = 0.8925
- C = 0.6443

Η εικόνα που θεωρούμε ιδανική (στη συγκεκριμένη περίπτωση δεν είναι, καθώς πρόκειται για πραγματική εικόνα) για την εξαγωγή των μετρικών:



Μέρος 2

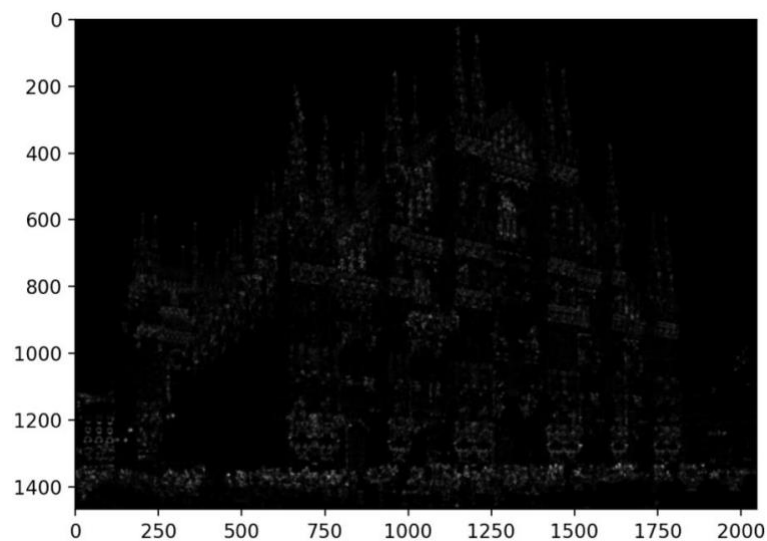
Στο συγκεκριμένο μέρος της άσκησης θα ασχοληθούμε με ανίχνευση σημείων ενδιαφέροντος (γωνίες, blobs) σε μια εικόνα. Αρχικά, θα υλοποιήσουμε τη μέθοδο των Harris-Stephens για την ανίχνευση γωνιών. Το πρώτο βήμα του αλγορίθμου είναι να υπολογίσουμε τα στοιχεία του δομικού τανυστή J για τα pixels της εικόνας. Οι ιδιοτιμές του εκφράζονται από τη σχέση (9) της εκφώνησης και είναι ιδιαίτερα κρίσιμες, καθώς η ιδιοτιμή λ_- είναι ενδεικτική του μέτρου της ελάχιστης αλλαγής φωτεινότητας προς μια κατεύθυνση, ενώ η λ_+ του μέτρου της μέγιστης αλλαγής φωτεινότητας προς μια κατεύθυνση. Το γεγονός αυτό αξιοποιείται ως εξής:

- Αν τα λ_- , λ_+ έχουν κοντινές τιμές και μικρές, τότε η περιοχή που εξετάζουμε ανήκει σε background.
- Αν τα λ_- , λ_+ έχουν μεγάλη διαφορά στις τιμές, τότε η περιοχή που εξετάζουμε ανήκει σε ακμή.
- Αν τα λ_- , λ_+ έχουν κοντινές τιμές και μεγάλες, τότε η περιοχή που εξετάζουμε επιδέχεται μεγάλες αλλαγές φωτεινότητας σε κάθε μεταβολή της στο χώρο, οπότε υπάρχει γωνία.

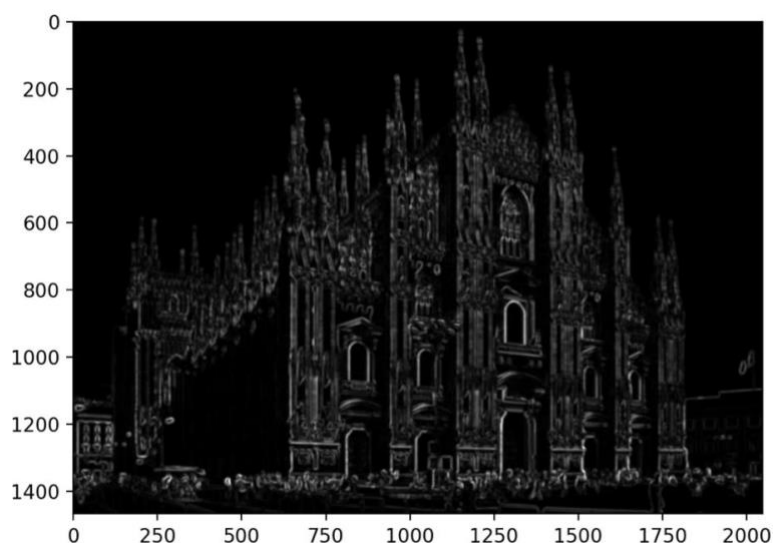
Το κριτήριο, για να καταλάβουμε πόσο «κοντά» ή όχι βρίσκονται οι ιδιοτιμές περιγράφεται από την σχέση (10) της εκφώνησης.

Για $\sigma = 2$ και $\rho = 2.5$ οι εικόνες των ιδιοτιμών είναι οι παρακάτω:

Για το λ_- :



Για το λ_+ :



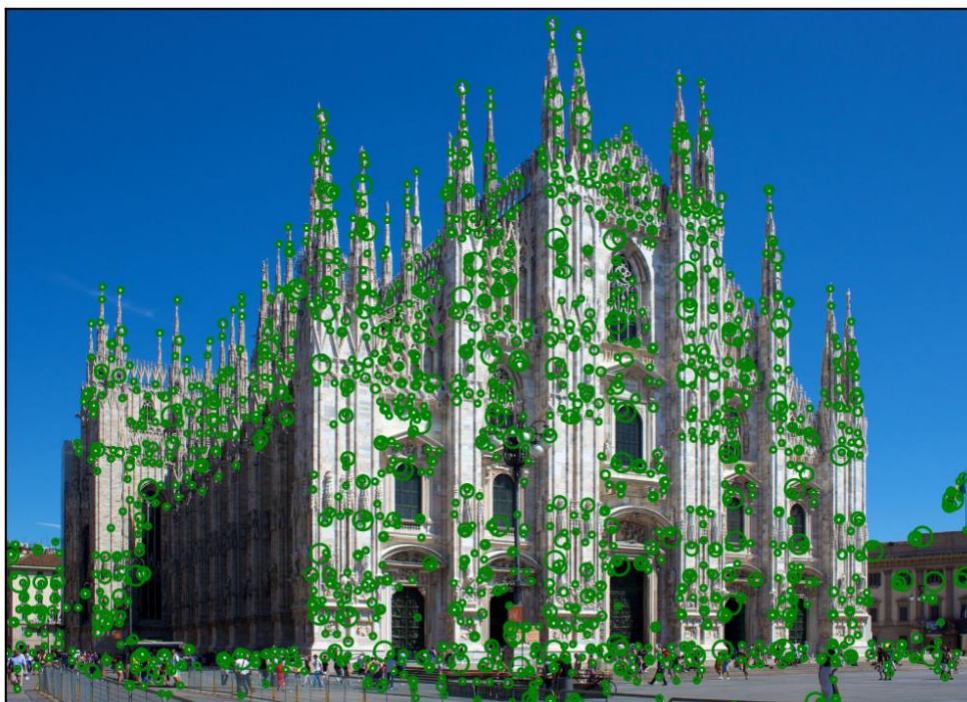
Παρατηρούμε πως το λ_+ υπολογίζει ένα υπερσύνολο των γωνιών, το οποίο προσεγγίζει τις ακμές της εικόνας. Αυτό το συμπέρασμα είναι λογικό, καθώς με δεδομένη τη μεγάλη τιμή του λ_+ , αν το λ_- έχει μικρή τιμή έχουμε ακμή, ενώ, αν έχει μεγάλη τιμή, έχουμε γωνία, όπως έχει ειπωθεί και παραπάνω. Αντίθετα, το λ_- περιέχει ένα σύνολο γωνιών διαφορετικής έντασης (φωτεινότητας).

Επομένως, με την εφαρμογή της σχέσης (10), θα έχουμε ένα σύνολο γωνιών της εικόνας, από το οποίο θα επιλέξουμε ένα υποσύνολο που ικανοποιεί τις συνθήκες $\Sigma 1$, $\Sigma 2$ με $\theta_{\text{corn}} = 0.005$.

Οι τελικές γωνίες που ανιχνεύονται φαίνονται παρακάτω:

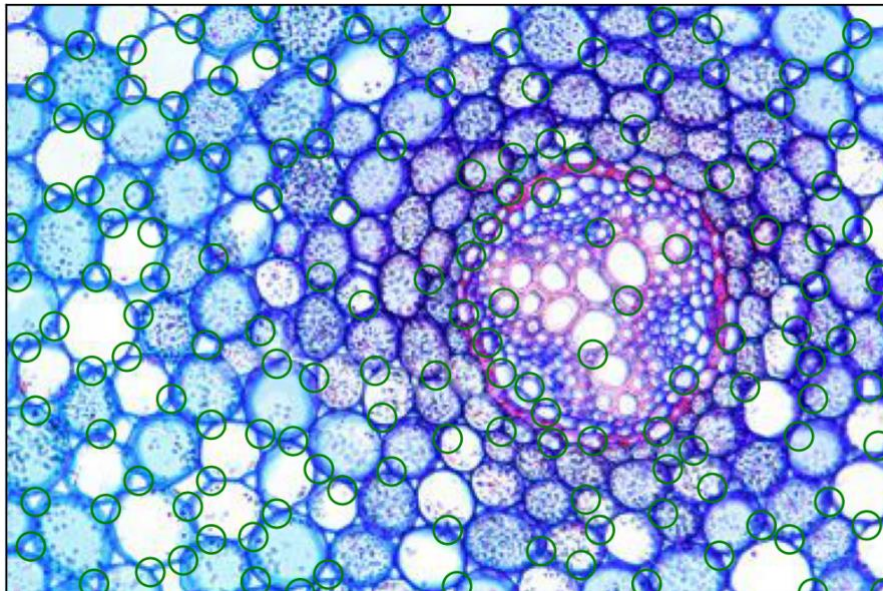


Στη συνέχεια, θα επαναλάβουμε την ίδια διαδικασία σε πολλές κλίμακες, ώστε να κρατάμε τις γωνίες που ανιχνεύονται σε κλίμακες που αποτελούν τοπικά μέγιστα των γειτονικών κανονικοποιημένων LoG. Στην παρακάτω εικόνα θα χρησιμοποιήσουμε 4 κλίμακες με αρχικά σ , ρ τα ίδια με πριν και $s=1.5$. Το αποτέλεσμα προκύπτει είναι το ακόλουθο:

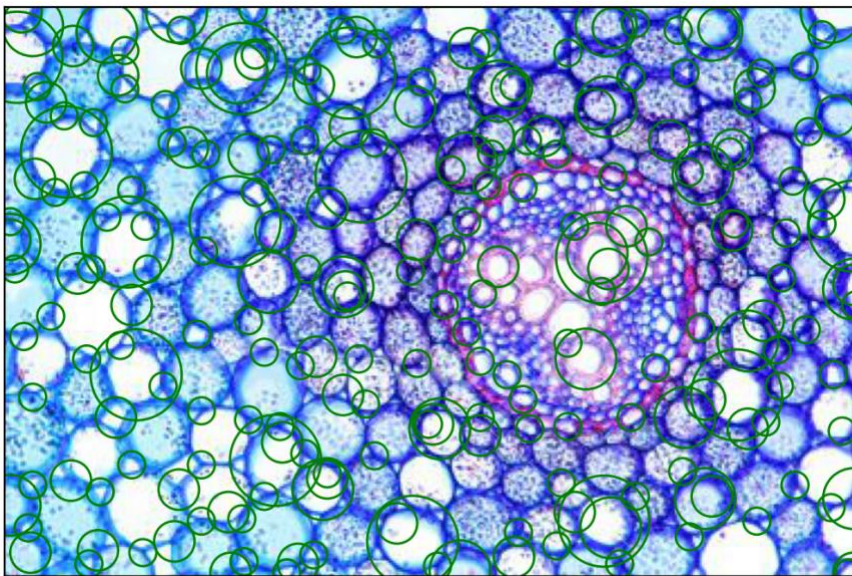


Μία άλλη σημαντική κατηγορία σημείων ενδιαφέροντος είναι τα blobs, τα οποία είναι περιοχές, που παρουσιάζουν μία διαφορετική ομοιογένεια, σε σχέση με αυτή της γειτονίας τους. Για την ανίχνευσή τους θα χρησιμοποιήσουμε έναν αλγόριθμο παρόμοιο με αυτόν που περιεγράφηκε προηγουμένως για την ανίχνευση γωνιών. Μόνη διαφορά είναι πως αυτή τη φορά το κριτήριο R ορίζεται ως η ορίζουσα του Εσσιανού πίνακα των μερικών παραγώγων δεύτερης τάξης της εξομαλυμένης εικόνας. Τελικά, λαμβάνουμε ένα υποσύνολο του R ανατρέχοντας στις ίδιες συνθήκες με την ανίχνευση γωνιών.

Δύο παραδείγματα ανίχνευσης blobs σε εικόνες με $\sigma=2$ είναι τα εξής:



Κατά αναλογία με πριν, εφαρμόζουμε πολυκλιμακωτή ανίχνευση blobs, με το ίδιο κριτήριο για την τελική επιλογή των blobs. Οι εικόνες που προκύπτουν με $s=1.5$, $N=4$ είναι οι παρακάτω:



Τέλος, θα επιχειρήσουμε να βελτιστοποιήσουμε τον αλγόριθμο μας με τη χρήση ολοκληρωτικών εικόνων και box filters. Με αυτά θα ακολουθήσουμε έναν πιο αποτελεσματικό τρόπο (trade off μεταξύ ακρίβειας των τιμών και υπολογιστικού χρόνου) για την προσέγγιση των φίλτρων των παραγώγων δεύτερης τάξης της εξομαλυμένης εικόνας. Ουσιαστικά, εκμεταλλευόμαστε το γεγονός, πως μπορούμε να υπολογίσουμε το άθροισμα των τιμών των pixels μιας «υποεικόνας» της ολοκληρωτικής σε $O(1)$. Για αυτό τον λόγο, θεωρούμε τις παραγώγους L_{xx} , L_{yy} ως ένα άθροισμα μεταξύ τριών τέτοιων «υποεικόνων» και την L_{xy} ως ένα άθροισμα τεσσάρων. Έπειτα, κατασκευάζουμε πάλι το κριτήριο R ως την ορίζουσα του Εσσιανού πίνακα με τις προσεγγίσεις των παραγώγων, αλλά στη θέση του L_{xy} βάζουμε το $0.9 \cdot L_{xy}$ (κάτι που έχει αποδειχθεί πειραματικά). Ακολουθούν μερικές οπτικοποιήσεις των blobs για διάφορες κλίμακες.

Για $\sigma=2$:



Για $\sigma=4$:



Για $\sigma=6$:



Για $\sigma=8$:



Για $\sigma=10$:



Από τα παραπάνω αποτελέσματα παρατηρούμε ότι, ενώ σε μικρές κλίμακες (πχ 2) έχουμε αρκετά καλές προσεγγίσεις των blobs, όσο μεγαλώνει η κλίμακα έχουμε μεγάλες ανακρίβειες, καθώς ο αριθμός των ανιχνευόμενων σημείων μειώνεται σημαντικά. Ωστόσο, δεν είναι αυτός ο μόνος λόγος που για $\sigma=8$ και $\sigma=10$ έχουμε 4 και 0 σημεία, αντίστοιχα. Αυτό οφείλεται και σε κάποιες τιμές στο περίγραμμα των παραγώγων δεύτερης τάξης της εικόνας που έχουν ακραίες τιμές. Αυξάνοντας την τιμή της μεταβλητής f της συνάρτησης `BlobDetectWithBoxFilters` κόβουμε μεγαλύτερο μέρος του περιγράμματος ενισχύοντας τα αποτελέσματα αυτά, αλλά, ταυτόχρονα, χάνουμε αρκετά σημεία σε μικρότερες κλίμακες. Θέτοντας $f=32$ έχουμε τα παρακάτω αποτελέσματα για $\sigma=8$ και $\sigma=10$:

Για $\sigma=8$:



Για $\sigma=10$:



Λόγω της προηγούμενης διαπίστωσης, η πολυκλιμακωτή ανίχνευση blobs θα πραγματοποιηθεί με τις ελαφρώς μειωμένες τιμές $\sigma=1.7$, $s=1.3$, $N=4$. Το αποτέλεσμα είναι το ακόλουθο:



Μέρος 3

Στο συγκεκριμένο μέρος, θα χρησιμοποιήσουμε τους προηγούμενους αλγορίθμους ανίχνευσης σημείων ενδιαφέροντος, για να κωδικοποιήσουμε τη γειτονιά αυτών των σημείων και να εξάγουμε χαρακτηριστικά για την εικόνα, που επεξεργαζόμαστε. Αυτή η διαδικασία θα πραγματοποιηθεί με τους δύο τοπικούς περιγραφητές HOG (Histogram of Oriented Gradients) και SURF (Speed Up Robust Features). Η εφαρμογή των εξαγόμενων χαρακτηριστικών θα είναι σε matching εικόνων διαφορετικής γωνίας περιστροφής (-20° , -10° , 0° , 10° , 20°) και κλίμακας (0.6, 0.8, 1.0, 1.2). καθώς και σε classification problem, δεδομένου ενός data set από τη βάση Pascal VOC2005 με εικόνες αυτοκινήτων, ανθρώπων και ποδηλάτων.

Η πρώτη εφαρμογή που θα εξεταστεί είναι η εκτίμηση της γωνίας περιστροφής και της κλίμακας στο πρόβλημα ταιριάσματος εικόνων που έχει προαναφερθεί. Χρησιμοποιώντας τις συναρτήσεις του αρχείου `cv22_lab1_part3_utils.py`, υπολογίζουμε τα μέσα σφάλματα γωνίας περιστροφής και κλίμακας για 3 εικόνες για κάθε συνδυασμό τοπικού περιγραφητή και μεθόδου ανίχνευσης σημείων ενδιαφέροντος. Τα αποτελέσματα που λαμβάνουμε συνοψίζονται στα επόμενα πίνακας:

Default παράμετροι που χρησιμοποιούνται:

Algorithm / Parameter	σ	r	k	θ	s	N
CornerDetect	2	2.5	0.05	0.005	-	-
MultiscaleCornerDetect	2	2.5	0.05	0.005	1.5	4
BlobDetect	2	-	-	0.005	-	-
MultiscaleBlobDetect	2	-	-	0.005	1.5	4
MultiscaleBlobDetectWithBoxFilters	1.7	-	-	0.005	1.3	4

Με HOG ως τοπικό περιγραφητή:

	Image 0		Image 1		Image 2	
Algorithm / Average Error	Scale	Theta	Scale	Theta	Scale	Theta
CornerDetect	0.162	21.570	0.223	16.795	0.256	23.739
MultiscaleCornerDetect	0.170	21.620	0.152	21.327	3.344	30.733
BlobDetect	0.186	7.231	0.100	13.674	0.154	27.219
MultiscaleBlobDetect	0.128	11.526	0.132	14.333	0.196	15.385
MultiscaleBlobDetectWithBoxFilters	0.166	18.986	0.245	21.205	0.114	13.194

Με SURF ως τοπικό περιγραφητή:

	Image 0		Image 1		Image 2	
Algorithm / Average Error	Scale	Theta	Scale	Theta	Scale	Theta
CornerDetect	0.004	2.854	0.002	0.402	0.097	12.908
MultiscaleCornerDetect	0.002	0.150	0.004	0.270	0.002	0.462
BlobDetect	0.027	7.759	0.010	0.229	0.001	0.054
MultiscaleBlobDetect	0.001	0.081	0.001	0.069	0.001	0.071
MultiscaleBlobDetectWithBoxFilters	0.003	0.152	0.003	0.190	0.002	0.162

Παρατηρούμε ότι με τον τοπικό περιγραφητή SURF λαμβάνουμε πολύ μικρότερα μέσα σφάλματα, γεγονός που ήταν αναμενόμενο, καθώς ένα κύριο προτέρημα, που χαρακτηρίζει τη μέθοδο των SURF είναι οι περιστροφικά ανεξάρτητοι περιγραφητές που εξάγονται ως χαρακτηριστικά. Επιπλέον, όπως είναι λογικό, οι πολυκλιμακωτές υλοποιήσεις έχουν, εν γένει, καλύτερα αποτελέσματα.

Στη συνέχεια, θα ασχοληθούμε με το πρόβλημα κατηγοριοποίησης εικόνων που περιγράφηκε παραπάνω. Αρχικά, θα καλέσουμε τις συναρτήσεις `p3.FeaturesExtraction` και `p3.CreateTrainTest`, για να εξάγουμε τα χαρακτηριστικά των εικόνων της βάσης και να τα χωρίσουμε να `train` και `test data set`, αντίστοιχα. Έπειτα, θα κληθεί η `p3.BagOfWords`, προκειμένου να γίνει το `clustering` και να λάβουμε το κανονικοποιημένο ιστόγραμμα χαρακτηριστικών. Η τελική κατηγοριοποίηση θα γίνει με χρήση της συνάρτησης `p3.svm`, η οποία καλεί ένα `Support Vector Machine`, για να γίνει η ταξινόμηση στις 3 κλάσεις. Η συνάρτηση, αυτή, επιστρέφει και το `accuracy` του `classification`, το οποίο θα είναι και η τελική μετρική για την αξιοπιστία του μοντέλου μας. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα (οι παράμετροι που χρησιμοποιήθηκαν είναι οι ίδιες με προηγουμένως):

	HOG		SURF	
	Accuracy	Time	Accuracy	Time
MultiscaleCornerDetect	61.793%	312	56.552%	269
MultiscaleBlobDetect	62.483%	283	58.069%	255
MultiscaleBlobDetectWithBoxFilters	65.379%	187	59.586%	146

Παρατηρούμε πως με τον τοπικό περιγραφητή HOG προκύπτουν καλύτερα αποτελέσματα (της τάξης του 5%) από ότι με τον SURF. Ο χρόνος που απαιτείται για την ολοκλήρωση της ταξινόμησης είναι παρόμοιος για τους δύο περιγραφητές (ίσως λίγο μικρότερος για τα SURFs), παρουσιάζει, ωστόσο σημαντική πτώση (40%-50%) στις υλοποιήσεις με τα box filters όπως αναμενόταν. Στη συγκεκριμένη περίπτωση, πέρα από καλύτερο χρόνο, πετυχαίνουμε και καλύτερο accuracy με τη συγκεκριμένη υλοποίηση. Επομένως, από τους 6 διαθέσιμους συνδυασμούς που έχουμε, θα διαλέγαμε αυτόν με τοπικούς περιγραφητές HOGs και αλγόριθμο ανίχνευσης σημείων ενδιαφέροντος την πολυκλιμακωτή ανίχνευση blobs με χρήση box filters και ολοκληρωτικών εικόνων, με accuracy 65.379%.