

Contents

1	Μέρος 1: Ανίχνευση Ακμών σε Γκριζες Εικόνες	3
1.1	Δημιουργία εικόνων εισόδου:	3
1.2	Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών:	3
1.3	Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών:	5
1.4	Εφαρμογή των Αλγορίθμων Ανίχνευσης Ακμών σε Πραγματικές εικόνες:	5
2	Ανίχνευση Σημείων Ενδιαφέροντος (Interest Point Detection)	6
2.1	Ανίχνευση Γωνιών:	6
2.2	Πολυκλιμακωτή Ανίχνευση Γωνιών:	8
2.3	Ανίχνευση <i>Blobs</i> :	9
2.4	Πολυκλιμακωτή Ανίχνευση <i>Blobs</i> :	10
2.5	Επιτάχυνση με την χρήση <i>Box Filters</i> και Ολοκληρωτικών Εικόνων (<i>Integral Images</i>):	10
3	Μέρος 3: Εφαρμογές σε Ταίριασμα και Κατηγοριοποίηση Εικόνων με Χρήση Τοπικών Περιγραφητών στα Σημεία Ενδιαφέροντος	13
3.1	Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας . . .	13
3.2	Κατηγοριοποίηση Εικόνων	14

1η Εργαστηριακή Άσκηση
Εντοπισμός Σημείων Ενδιαφέροντος και
Εξαγωγή Χαρακτηριστικών σε Εικόνες

Ανδρέας Βεζάκης AM: 03117186

Κωνσταντίνος Κωστόπουλος AM: 03117043

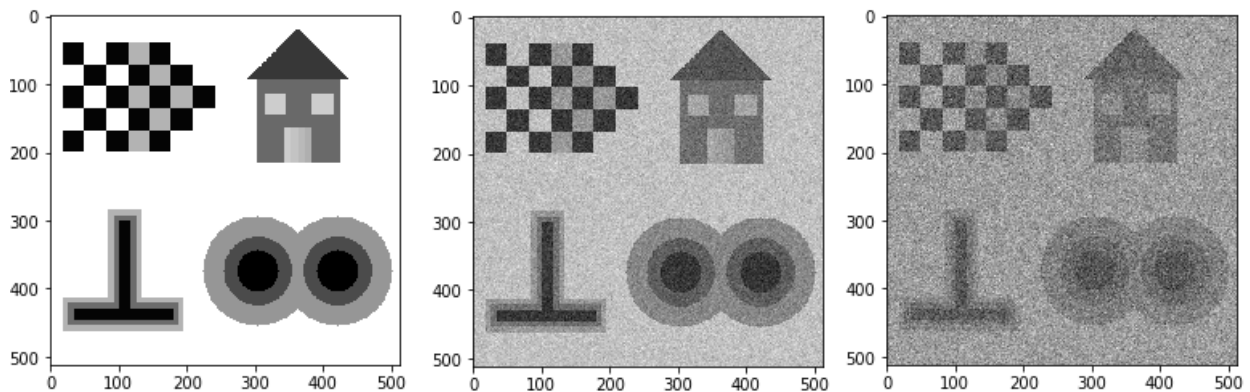
10 Απριλίου 2021

1 Μέρος 1: Ανίχνευση Ακμών σε Γκριζες Εικόνες

1.1 Δημιουργία εικόνων εισόδου:

Αρχικά, για απλότητα, κανονικοποιούμε την εικόνα μας στις τιμές $[0,1]$, και ύστερα προσθέτουμε *White Gaussian Noise*, με μηδενική μέση τιμή και τυπική απόκλιση τέτοια ώστε ο *PSNR* να ικανοποιεί τις τιμές:

- $PSNR = 20dB$
- $PSNR = 10dB$



Σχήμα 1: α)Κανονικοποιημένη εικόνα, β) $PSNR = 20dB$, γ) $PSNR = 10dB$

Για τον *PSNR* γνωρίζουμε ότι $PSNR = 20\log_{10} \frac{I_{max} - I_{min}}{\sigma_n}$

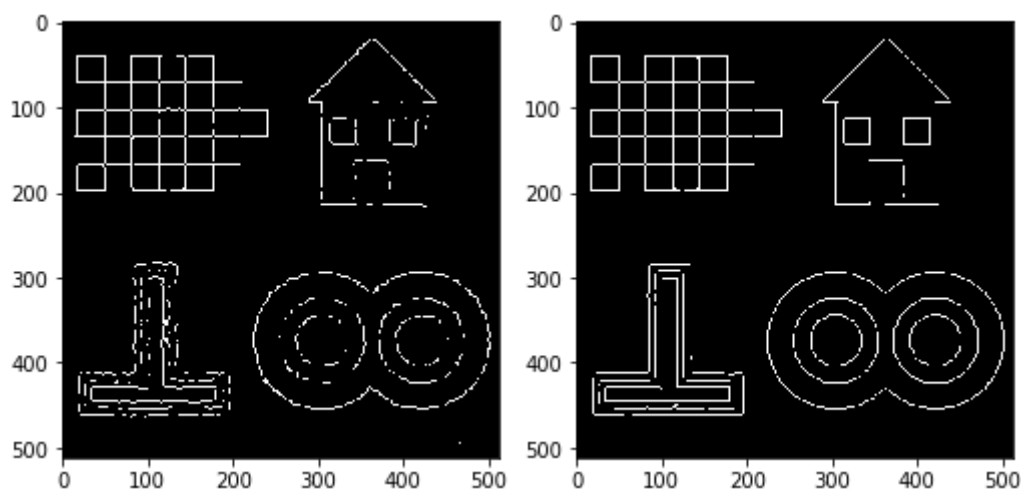
Επομένως, $\sigma_n = \frac{I_{max} - I_{min}}{10^{PSNR/20}}$

1.2 Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών:

Για να ανιχνεύσουμε τις ακμές στην παραπάνω θορυβώδη εικόνα, βασιζόμαστε στο ότι οι ακμές μια θορυβώδους εικόνας μπορούν να δοθούν από την εύρεση των σημείων μηδενισμού της Λαπλασιανής της εξομαλυμένης εικόνας τα οποία έχουν αρκετά μεγάλη κλίση. Ο αλγόριθμος που ακολουθήσαμε έχει τα παρακάτω βήματα.

1. Δημιουργούμε 2 *kernel*, που προσεγγίζουν τις κρουστικές αποκρίσεις των παρακάτω γραμμικών φίλτρων με τυπική απόκλιση σ , και διάσταση $n \times n$, με $n = 2\text{ceil}(3\sigma) + 1$.
 - Διδιάστατη *Gaussian*: $G_\sigma(x, y)$
 - *Laplacian of Gaussian*: $LoG = \nabla G_\sigma(x, y)$
2. Προσέγγιζουμε τη *Laplacian* L της εξομαλυμένης εικόνας $I_\sigma = G_\sigma * I(x, y)$, με 2 εναλλακτικές.
 - Γραμμική (L_1): $L_1 = (\nabla^2 G_\sigma) * I = h * I$
 - Μη Γραμμική L_2 : $L_2 = I_\sigma \oplus B + I_\sigma \ominus B - 2I_\sigma$
3. Εντοπίζουμε τα σημεία μηδενισμού της L .
 - Εφαρμόζοντας *threshold* στην L (βάζουμε τιμή 1, στις θέσεις που η Λαπλασιανή της έχει μη αρνητικές τιμές εικόνας), και υπολογίζοντας έτσι την δυαδική εικόνα προσήμου.
 - Εντοπίζουμε το περίγραμμα Y της X . $Y = (X \oplus B) - (X \ominus B) \approx \partial X$
4. Τέλος επιλέγουμε ως ακμές τα σημεία στα οποία η εξομαλυμένη εικόνα έχει σχετικά μεγάλη κλίση.

Παρατηρούμε ότι για $PSNR = 20dB$, η συνάρτησή μας έχει αρκετά καλή ακρίβεια, ενώ για $PSNR = 10dB$ δηλαδή όταν υπάρχει περισσότερος θόρυβος, η ακρίβεια πέφτει το οποίο ήταν αναμενόμενο. Να σημειώσουμε ότι η συνάρτησή μας δέχεται όρισμα για την επιλογή του τρόπου προσέγγισης της Λαπλασιανής (Γραμμικά είτε Μη Γραμμικά), και ότι οι εικόνες που παρουσιάζονται είναι τα καλύτερα αποτελέσματα που μπορούσαμε να εξάγουμε με αυτήν.



Σχήμα 2: Εφαρμογή της συνάρτησης στις εικόνες με α) $PSNR = 10dB$, β) $PSNR = 20dB$

1.3 Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών:

1. Για την εικόνα με $PSNR = 20dB$

- Με γραμμική προσέγγιση της Λαπλασιανής έχουμε ακρίβεια ≈ 0.91
- Με μη γραμμική προσέγγιση της Λαπλασιανής έχουμε ακρίβεια ≈ 0.97

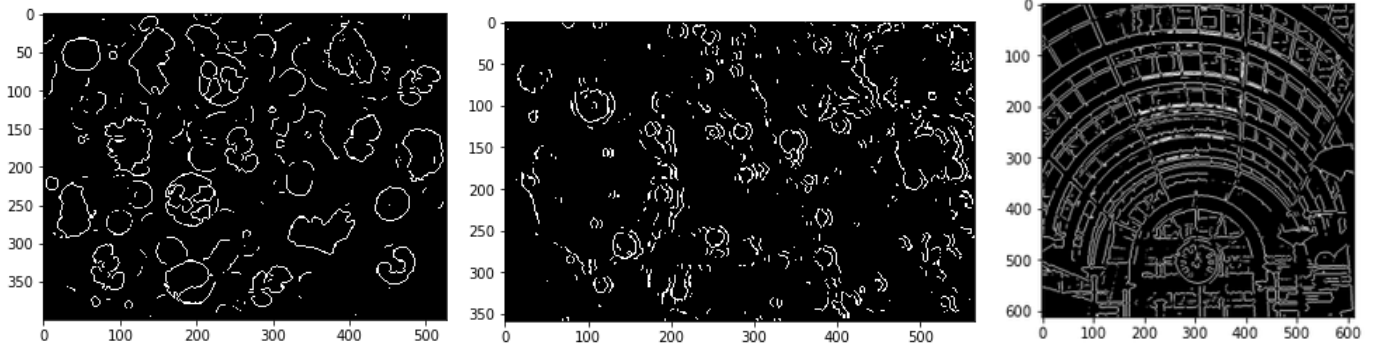
2. Για την εικόνα με $PSNR = 10dB$

- Με γραμμική προσέγγιση της Λαπλασιανής έχουμε ακρίβεια ≈ 0.70
- Με μη γραμμική προσέγγιση της Λαπλασιανής έχουμε ακρίβεια ≈ 0.77

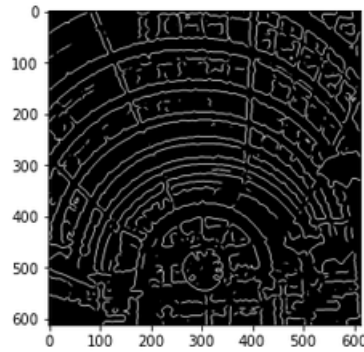
1.4 Εφαρμογή των Αλγορίθμων Ανίχνευσης Ακμών σε Πραγματικές εικόνες:

Εφαρμόζοντας την συνάρτηση σε πραγματικές εικόνες, και δοκιμάζοντας διαφορετικές τιμές παραμέτρων, βλέπουμε ότι η συνάρτηση μας, καταφέρνει να εντοπίσει ικανοποιητικά τις ακμές της κάθε εικόνας. Επίσης, όσο αυξάνουμε το

σ και το Θ_{edge} , υπολογίζει όλο και λιγότερες ακμές όπως φαίνεται στο Σχήμα 4.



Σχήμα 3: Εφαρμογή της συνάρτησης στις εικόνες "blood_smear", "mars" και urban_edges ($\Theta_{edge}=2.5$).



Σχήμα 4: Εφαρμογή της συνάρτησης στην εικόνα urban_edges, με $\Theta_{edge}=5$.

2 Ανίχνευση Σημείων Ενδιαφέροντος (Interest Point Detection)

2.1 Ανίχνευση Γωνιών:

Αρχικά κατασκευάζουμε τις Γαусσιαν συναρτήσεις πυρήνα G_σ , G_ρ , όπως κατασκευάσαμε το αντίστοιχο *kernel*, στα παραπάνω ερωτήματα. Φιλτράρουμε την

εικόνα μας με το $G_\sigma * I = I_\sigma$, υπολογίζουμε τις μερικές παραγώγους αυτής, και πολλαπλασιάζοντας κατάλληλα και φιλτράροντας με την G_ρ υπολογίζουμε τα J_1, J_2, J_3 του δομικού τανυστή J .

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

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

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

Έπειτα υπολογίζουμε τις ιδιοτιμές του δομικού τανυστή, και εφαρμόζουμε το κριτήριο γωνιότητας R :

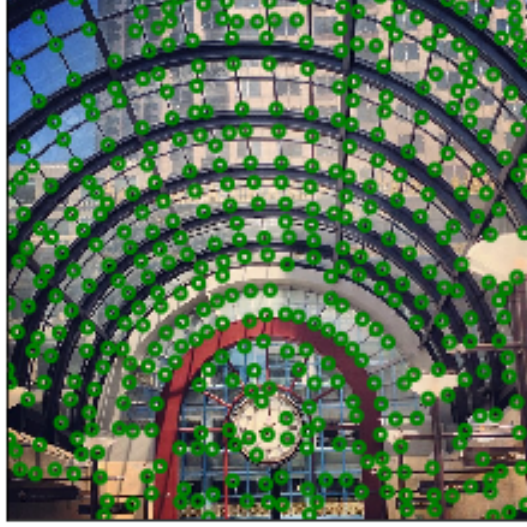
$$J = \begin{bmatrix} J_1(x, y) & J_2(x, y) \\ J_2(x, y) & J_3(x, y) \end{bmatrix}$$

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

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

Τέλος επιλέγουμε τα *pixels* που ικανοποιούν τις 2 παρακάτω συνθήκες:

- Είναι μέγιστα εντός τετραγωνικού παραθύρου του R . Για την υλοποίηση κατασκευάζουμε ένα παράθυρο δίσκου B και εφαρμόζουμε *dilation*.
 $Cond1 = R \oplus B$
- Η τιμή τους βρίσκεται πάνω από ένα κατώφλι που έχουμε επιλέξει.
 $R(x, y) > \theta_{corn} R_{max}$



Σχήμα 5: Εφαρμογή της μεθόδου *Harris-Stephens* με $\sigma=2$, $\rho=2$, $\vartheta_{corn} = 0.005$, $k = 0.05$

2.2 Πολυκλιμακωτή Ανίχνευση Γωνιών:

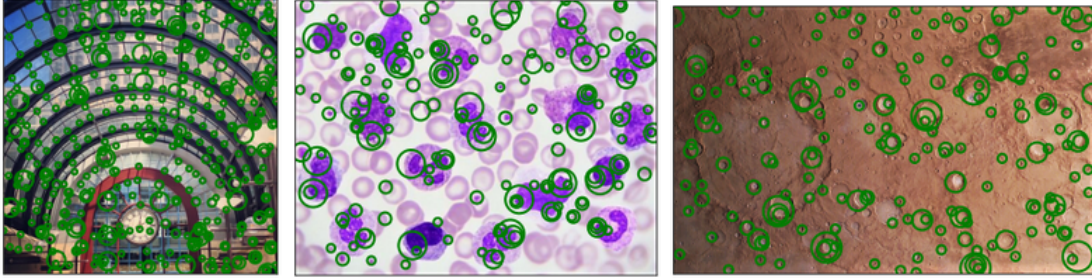
Στο βήμα αυτό, θέλουμε να ανιχνεύσουμε γωνίες σε διαφορετικές κλίμακες. Υλοποιούμε τη μέθοδο *Harris-Laplacian*, όπου αρχικά εφαρμόζουμε τον αλγόριθμο εύρεσης γωνιών μονής κλίμακας για διαφορετικές κλίμακες ολοκλήρωσης και διαφορίσης, ενώ στη συνέχεια επιλέγουμε τα σημεία τα οποία μεγιστοποιούν την κανονικοποιημένη Λαπλασιανή της Γκαουσιανής (*Laplacian of Gaussian LoG*) σε μια γειτονιά διαδοχικών κλιμάκων.

Τις διαφορετικές κλίμακες τις υπολογίζουμε με τη σχέση:

$$\begin{aligned}\sigma_0, \sigma_1, \dots, \sigma_{N-1} &= s^0 \sigma_0, s^1 \sigma_0, \dots, s^{N-1} \sigma_0 \\ \rho_0, \rho_1, \dots, \rho_{N-1} &= s^0 \rho_0, s^1 \rho_0, \dots, s^{N-1} \rho_0\end{aligned}$$

Η κανονικοποιημένη *LoG* ορίζεται ως:

$$|LoG(x, \sigma_i)| = \sigma_i^2 |L_{xx}(x, \sigma_i) + L_{yy}(x, \sigma_i)|$$



Σχήμα 6: Εφαρμογή της μεθόδου *Harris-Laplacian*

2.3 Ανίχνευση *Blobs*:

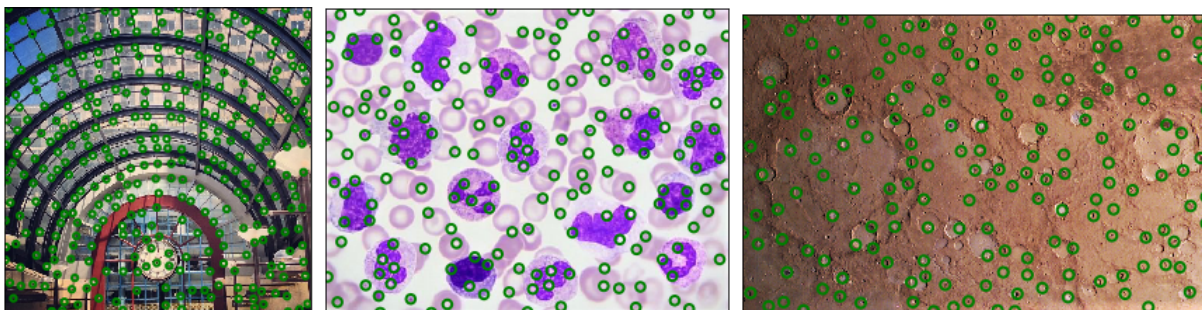
Εν γένει ως *blob* ορίζουμε την περιοχή μιας ψηφιακής εικόνας η οποία παρουσιάζει κάποια κοινά στοιχεία και διαφέρει ως προς το περιβάλλον της. Τα κοινά αυτά στοιχεία, δηλαδή τα σημεία ενδιαφέροντος, μπορεί να αφορούν το χρώμα ή τη φωτεινότητα της περιοχής. Επομένως τα σημεία αυτά βρίσκουν ιδιαίτερη εφαρμογή στην όραση υπολογιστών, καθώς με αυτά μπορούμε να προσδιορίσουμε αντικείμενα στο χώρο, πηγές φωτός, όπου η αναζήτησή τους δεν είναι εφικτή με μεθόδους ανίχνευσης ακμών ή γωνιών.

Για την εύρεση τέτοιων περιοχών, σε αντιστοιχία με το κριτήριο γωνιότητας της μεθόδου *Harris*, γίνεται χρήση των μερικών παραγώγων δεύτερης τάξης της εικόνας και συγκεκριμένα η ορίζουσα R του πίνακα *Hessian* για κάθε *pixel* της εικόνας:

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

Έπειτα, θα πρέπει να ικανοποιούνται οι 2 συνθήκες που περιγράψαμε στο μέρος 2.1.

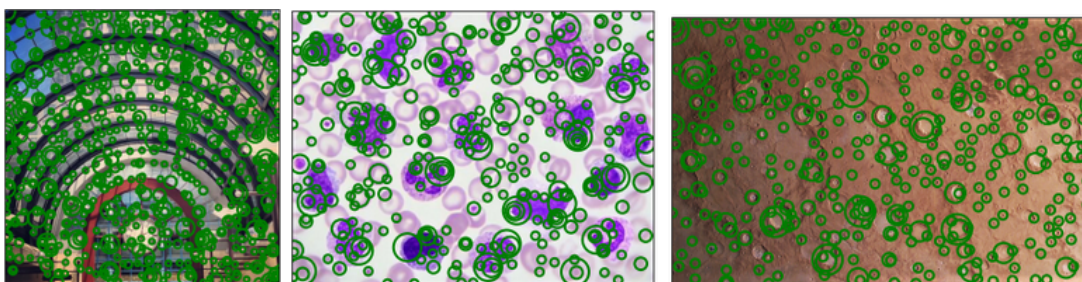
Με άλλα λόγια, στη μονοκλιμακωτή ανάλυση εξετάζουμε τις δεύτερες παραγώγους της φιλτραρισμένης εικόνας ως προς τους ορθογώνιους άξονες. Εφαρμόζοντας *dilation* ουσιαστικά εξετάζουμε τα γειτονικά *pixel* σε ένα παράθυρο δίσκου, αθροίζοντάς τα. Αν το άθροισμα αυτό βρίσκεται πάνω από μια προκαθορισμένη τιμή, τότε αποτελεί *blob* για την συγκεκριμένη κλίμακα σ .



Σχήμα 7: Εφαρμογή της *blob_detect* στις 3 εικόνες που μας δίνονται.

2.4 Πολυκλιμακωτή Ανίχνευση Blobs:

Όπως αναφέρθηκε τα *blobs* που ανιχνεύονται στην μονοκλιμακωτή ανάλυση εξαρτώνται από την κλίμακα σ που χρησιμοποιούμε στην εικόνα. Συνεπώς, ενδέχεται να εντοπιστούν *blobs* τα οποία ουσιαστικά ομαδοποιούνται σε μια μεγαλύτερη κατηγορία. Για να γίνει αυτό απαιτείται ανάλυση σε διαφορετικές κλίμακες. Θα ακολουθήσουμε τα ίδια ακριβώς βήματα με την πολυκλιμακωτή ανάλυση γωνιών, τρέχοντας για κάθε διαφορετική κλίμακα την συνάρτηση *blob_detect*, που φτιάξαμε για το ερώτημα 2.3.



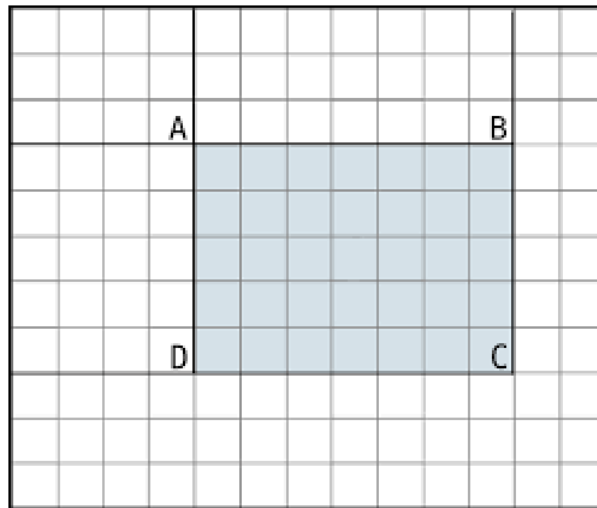
Σχήμα 8: Εφαρμογή της *multi_blob_detect*.

2.5 Επιτάχυνση με την χρήση Box Filters και Ολοκληρωτικών Εικόνων (Integral Images):

Καθώς, η παραπάνω διαδικασία (Ανίχνευση *blobs*) απαιτεί συνέλιξη της εικόνας με αυξανόμενου μεγέθους φίλτρα, καταλαβαίνουμε ότι είναι αρκετά χρονοβόρα.

Στο μέρος αυτό, θα βελτιώσουμε την πολυπλοκότητα του αλγορίθμου μας. Αρχικά, προσεγγίζουμε αυτά τα φίλτρα, με έτοιμα *box filters*. Δηλαδή με φίλτρα που βασίζονται σε αθροίσματα ορθογώνιων περιοχών. Έπειτα, δημιουργούμε την ολοκληρωτική εικόνα. Δηλαδή, μία εικόνα, κάθε *pixel* της οποίας, ισούται με το άθροισμα των πάνω και αριστερά από αυτό *pixel* της αρχικής εικόνας. Επίσης, γνωρίζουμε ότι το άθροισμα των στοιχείων ενός παραθύρου είναι το εξής:

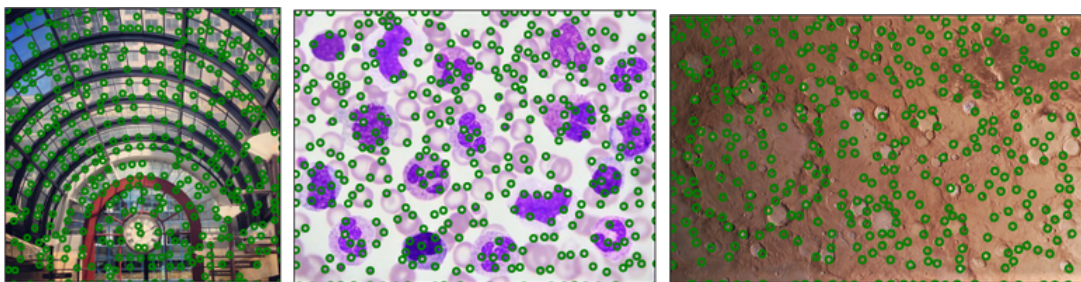
$$\sum_{A,B,C,D} I(i,j) = S_A + S_C - S_B - S_D$$



Σχήμα 9: Παράθυρο Υπολογισμού Αθροίσματος

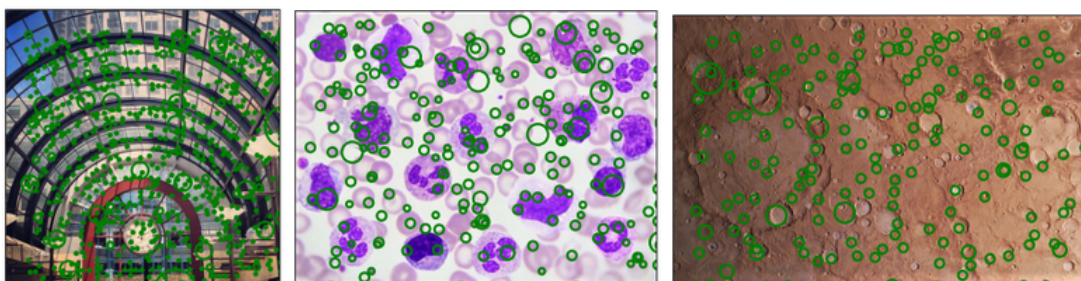
Χρησιμοποιώντας την παραπάνω σχέση, για κάθε *pixel*, μπορούμε να υπολογίσουμε το εμβαδόν του παραθύρου, βρίσκοντας τα 4 σημεία που το ορίζουν. Επομένως, αντιλαμβανόμαστε ότι βελτιώνεται η πολυπλοκότητα του αλγορίθμου μας, αφού δεν κάνουμε συνέλιξη και δεν περνάμε από όλα τα *pixels* του παραθύρου για τον υπολογισμό του αθροίσματος του.

Η ζητούμενη συνάρτηση υλοποιήθηκε, παίρνοντας την ολοκληρωτική εικόνα, και κάνοντας κατάλληλα *np.roll* ώστε τα 4 σημεία του κάθε παραθύρου να πέσουν το ένα πάνω στο άλλο. Έπειτα, με απλή πρόσθεση πινάκων, έχουμε το ζητούμενο αποτέλεσμα. (Στην ολοκληρωτική εικόνα έγινε κατάλληλο *padding* ώστε να συμπεριλάβουμε όλα τα *pixels* της.)



Σχήμα 10: Εφαρμογή της συνάρτησης "*box_ft*".

Τέλος, επαναλαμβάνοντας τον ίδιο αλγόριθμο με τις προηγούμενες πολυκλιμακωτές συναρτήσεις δημιουργούμε την *multi_box_ft*, και έχουμε τα παρακάτω αποτελέσματα.



Σχήμα 11: Εφαρμογή της συνάρτησης "*multi_box_ft*".

3 Μέρος 3: Εφαρμογές σε Ταίριασμα και Κατηγοριοποίηση Εικόνων με Χρήση Τοπικών Περιγραφητών στα Σημεία Ενδιαφέροντος

Στο τρίτο μέρος της άσκησης σκοπός μας είναι να συγκρίνουμε τα αποτελέσματα των παραπάνω μεθόδων εξαγωγής σημείων ενδιαφέροντος σε συνδυασμό με τη χρήση δύο τοπικών περιγραφητών, των *Speed-Up Robust Features (SURF)* και *Histogram of Oriented Gradient (HOG)*. Η εξαγωγή τοπικών περιγραφητών αφορά στην κωδικοποίηση μιας γειτονιάς γύρω από τα σημεία ενδιαφέροντος κατά τρόπο τέτοιο ώστε να επιτρέπεται το 'ταίριασμα' (*matching*) των σημείων ενδιαφέροντος με τους τοπικούς περιγραφητές, ακόμα κι αν στην εικόνα έχει εφαρμοσθεί κάποιος μετασχηματισμός ομοιότητας (δηλαδή περιστροφή ή/και κλίμακωση). Στόχος είναι επίσης ο προσδιορισμός του μετασχηματισμού ομοιότητας, δηλαδή η ικανότητά μας να προσδιορίσουμε την περιστροφή και τη διαφορά κλίμακας ανάμεσα σε μία αρχική εικόνα και μία παραμορφωμένη μορφή αυτής. Επιπλέον, καθώς οι παραπάνω τοπικοί περιγραφητές δίνουν σημαντική πληροφορία για τη μορφή των εικονιζόμενων αντικειμένων, θα μελετήσουμε την αποτελεσματικότητά τους στη διαδικασία κατηγοριοποίησης εικόνων σε κλάσεις.

3.1 Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας

Σε αυτό το σημείο θα ελέγξουμε την ικανότητά των δύο περιγραφητών, να βοηθούν στην επίτευξη της εκτίμησης της περιστροφής και της κλίμακας μιας παραμορφωμένης εικόνας, δεδομένου ότι γνωρίζουμε την αρχική μορφή της. Πιο συγκεκριμένα, θα εφαρμόσουμε τους τοπικούς περιγραφητές *SURF* και *HOG* συνδυαστικά με κάθε έναν από τους ανιχνευτές σημείων ενδιαφέροντος που υλοποιήσαμε στο δεύτερο μέρος της άσκησης, εκτός από τη μονοκλιμακωτή έκδοση των *box filters*. Για την απλούστερη υλοποίηση του συγκεκριμένου μέρους, τροποποιήσαμε τις συναρτήσεις, ώστε να επιστρέφουν τα ζητούμενα δεδομένα, και να δέχονται λιγότερα ορίσματα (καθώς μερικά δεν χρειαζόνταν). Λάβαμε τα ακόλουθα αποτελέσματα:

Μέθοδος Ανίχνευσης Σημείων Ενδιαφέροντος	<i>SURF</i>		<i>HOG</i>	
	<i>scale_error</i>	<i>theta_error</i>	<i>scale_error</i>	<i>theta_error</i>
1ο Σύνολο Εικόνων				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.003	1.968	0.186	22.619
Πολυκλιμακωτή Ανίχνευση Γωνιών	0.002	0.156	0.131	15.515
Μονοκλιμακωτή Ανίχνευση <i>Blobs</i>	0.027	7.759	0.186	7.231
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i>	0.002	0.120	0.212	18.952
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i> με <i>Box Filters</i>	0.051	2.701	0.142	20.306
2ο Σύνολο Εικόνων				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.002	0.318	0.351	19.199
Πολυκλιμακωτή Ανίχνευση Γωνιών	0.005	0.313	1.045	37.957
Μονοκλιμακωτή Ανίχνευση <i>Blobs</i>	0.010	0.229	0.100	13.674
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i>	0.003	0.152	0.241	19.897
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i> με <i>Box Filters</i>	0.023	3.612	0.332	31.674
3ο Σύνολο Εικόνων				
Μονοκλιμακωτή Ανίχνευση Γωνιών	0.097	12.909	0.285	23.699
Πολυκλιμακωτή Ανίχνευση Γωνιών	0.004	0.319	0.253	22.740
Μονοκλιμακωτή Ανίχνευση <i>Blobs</i>	0.001	0.054	0.154	27.219
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i>	0.002	0.137	0.255	23.719
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i> με <i>Box Filters</i>	0.043	11.951	0.202	26.599

Παρατηρούμε ότι σε κάθε μέθοδο η *SURF* δίνει σημαντικά καλύτερη εκτίμηση περιστροφής και κλίμακας των εικόνων σε σχέση με τη μέθοδο *HOG*. Αυτό συμβαίνει επειδή η μέθοδος *SURF*, ενσωματώνει την πληροφορία των κατευθυντικών παραγώγων σε μία γενικότερη πληροφορία, γεγονός που την κάνει περισσότερο αναλλοίωτη σε περιστροφή και κλιμάκωση σε σχέση με τη *HOG*, για την οποία οι κατευθυντικές παράγωγοι παίζουν πρωταρχικό ρόλο. Επίσης, παρατηρούμε ότι οι πολυκλιμακωτές μέθοδοι δίνουν πολύ βελτιωμένα αποτελέσματα σε σχέση με τις αντίστοιχες μονοκλιμακωτές μεθόδους. Η παρατήρηση αυτή ήταν αναμενόμενη, καθώς όπως διαπιστώθηκε και στο δεύτερο μέρος της άσκησης, οι πολυκλιμακωτές μέθοδοι επιτυγχάνουν με πολύ καλή ακρίβεια εντοπισμό σημείων ενδιαφέροντος και ως εκ τούτου επιτρέπουν στον περιγραφητή να εκτιμήσει καλύτερα τη γειτονιά γύρω από τα σημεία αυτά.

3.2 Κατηγοριοποίηση Εικόνων

Στο σημείο αυτό σκοπός μας είναι να μελετήσουμε την αποτελεσματικότητα των τοπικών περιγραφητών *SURF* και *HOG* στην κατηγοριοποίηση εικόνων

σε κλάσεις. Συγκεκριμένα, μας δίνεται ένα σύνολο εικόνων που μπορούν να ταξινομηθούν στις κλάσεις αυτοκίνητο, άνθρωπος και ποδήλατο. Αρχικά χρησιμοποιούμε την συνάρτηση *FeatureExtraction* που μας δίνεται, προκειμένου να εξάγουμε τα χαρακτηριστικά που θα χρησιμοποιηθούν ως αναγνωριστικά για την ταξινόμηση. Στη συνέχεια, τρέχουμε τη συνάρτηση *createTrainTest* ώστε το σύστημά μας να εκπαιδευτεί (*train*) στο τι αναμένεται να κατηγοριοποιηθεί σε κάθε μία από τις κατηγορίες. Ένα ποσοστό των δεδομένων χρησιμοποιείται για το *train*, και το υπόλοιπο για τον έλεγχο έγκυρης ταξινόμησης. Τέλος, χρησιμοποιώντας τις έτοιμες συναρτήσεις *BagOfWords* και *svm*, τυπώνουμε το συνολικό ποσοστό επιτυχίας.

Μέθοδος Ανίχνευσης Σημείων Ενδιαφέροντος	<i>BagOfWords</i>	
	<i>SURF</i>	<i>HOG</i>
Πολυκλιμακωτή Ανίχνευση Γωνιών	60.414	66.483
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i>	56.138	68.966
Πολυκλιμακωτή Ανίχνευση <i>Blobs</i> με <i>Box Filters</i>	61.931	68.138

Παρατηρούμε ότι ο περιγραφητής *HOG* μας δίνει καλύτερα αποτελέσματα σε σχέση με τον περιγραφητή *SURF*. Αυτό συμβαίνει γιατί η μέθοδος *HOG* δημιουργεί ένα "πλέγμα" το οποίο καλύπτει σε αδρές αλλά πολύ κατατοπιστικές γραμμές το αντικείμενό μας, με αποτέλεσμα να επιτρέπει την εύκολη αντιστοίχιση μεταξύ ίδιων αντικειμένων. Τέλος, (όπως φαίνεται και από τα παραπάνω αποτελέσματα) καταλληλότερος συνδυασμός μεθόδου ανίχνευσης σημείων ενδιαφέροντος και τοπικού περιγραφητή που επιλύει το πρόβλημα της κατηγοριοποίησης εικόνων είναι η πολυκλιμακωτή ανίχνευση *blobs* σε συνδυασμό με περιγραφητή *HOG*, καθώς λαμβάνουμε ποσοστό 68.966.