Scale Invariant Feature Transform and Feature Matching

Ρέππα Σμαράγδα - 1115201600143
Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών
Ακαδημαϊκό έτος: 2020-2021

I. Περίληψη

Σκοπός αυτής της εργασίας ήταν πρώτον, να γίνουν πειράματα πάνω στις παραμέτρους του μοντέλου SIFT, να εξαχθούν συμπεράσματα σχετικά με τα αποτελέσματα, την απόδοση του μοντέλου κατά τη διάρκεια των πειραμάτων. Δεύτερον, αν μια φωτογραφία περνούσε από ένα μοντέλο με τις προκαθορισμένες τιμές του και η ίδια φωτογραφία περνούσε από χαμηλοπερατό φίλτρο, τί παραμέτρους θα δίναμε στο δεύτερο μοντέλο από το οποίο θα περνούσε η επεξεργασμένη φωτογραφία , ώστε να έχουμε ιδανικά τα ίδια ή όσο το δυνατόν παρόμοια αποτελέσματα με το πρώτο μοντέλο. Για αυτή τη διαδικασία θα χρησιμοποιηθούν τα αποτελέσματα των πειραμάτων. Σε αυτή την αναφορά, περιέχεται συνοπτική περιγραφή του αλγορίθμου, λίγα λόγια για τα φίλτρα που χρησιμοποιήθηκαν και σχολιασμός των αποτελεσμάτων των πειραμάτων.

II. Εισαγωγή

Η τεχνική της ανίχνευσης χαρακτηριστικών (feature detection) και το ταίριασμα αυτών των χαρακτηριστικών (feature matching) σε μια εικόνα χρησιμοποιούνται σε πολλούς τομείς όπως στη μηχανική όραση, στη μηχανική μάθηση, στην επεξεργασία ή ανάκτηση εικόνων, στην τεχνητή νοημοσύνη.

Ως χαρακτηριστικό ορίζεται ενα κομμάτι πληροφορίας, για το περιεχόμενο της εικόνας ή πιο συγκεκριμένα για μια περιοχή της αποτελεί πληροφορία για ορισμένες ιδιότητες της. Μπορεί να είναι ακμές (edges), γωνίες (corners), περιοχές σημαντικών σημείων (blobs) ή μέρη της εικόνας. [(1)] Τέτοια σημεία ξεχωρίζουν με κριτήριο πού μεταβάλλεται απότομα ή παρατηρείται ασυνέχεια στη φωτεινότητα, στην αντίθεση. Η ύπαρξη τους μπορεί να οφείλεται σε ασυνέχειες στο βάθος, ασυνέχειες στις επιφάνειες που απεικονίζονται, διαφοροποίηση στη φωτεινότητα σε τοπικό επίπεδο.

Υπάρχουν πολλοί αλγόριθμοι για αυτό το σκοπό με ακρωνύμια, SIFT, SURF, BRISK, BRIEF, ORB, ο καθένας με τα δικά του χαρακτηριστικά, με τα δικά του προτερήματα και μειονεκτήματα έναντι κάποιου άλλου, αναλόγως ποιός είναι ο στόχος εκείνου που θα τον χρησιμοποιήσει. Στο συγκεκριμένο project έχει επιλεγεί ο SIFT.

Άλλη τεχνική που χρησιμοποιήθηκε για τους σκοπούς της εργασίας είναι το ταίριασμα χαρακτηριστικών εικόνων (feature matching). Για δύο εικόνες image και image' παίρνουμε ενα ζεύγος χαρακτηριστικών (Xi, Yi) και (Xi', Yi'), όπου (Xi, Yi) είναι ένα χαρακτηριστικό της image και (Xi', Yi') της image', διαδικασία που λαμβάνει μέρος για όλα τα χαρακτηριστικά της εικόνας.

III. SIFT

1. Scalar Invariant Feature Transform

Το SIFT είναι ένας αλγόριθμος που ανιχνεύει χαρακτηριστικά τοπικού χαρακτήρα σε εικόνες. Δημοσιεύτηκε από τον David Lowe, Καναδό ερευνητή και καθηγητή, στον τομέα της πληροφορικής, το 2004 με τίτλο "*Distinctive Image Features from Scale-Invariant Keypoints*". Έχει εφαρμογές πάνω σε object recognition, robotic mapping and navigation, image stitching, 3D modeling και σε άλλους τομείς. [(2)]

Πιο συγκεκριμένα ο αλγόριθμος εντοπίζει χαρακτηριστικά που ονομάζονται keypoints και υπολογίζει τους περιγραφείς τους (descriptors). Κάθε keypoint έχει κλίμακα, τοποθεσία και προσανατολισμό, ενώ ο descriptor περιγράφει την περιοχή τοπικά του keypoint. Αυτά τα σημεία είναι ανεξάρτητα της κλίμακας και της όποιας περιστροφής μιας εικόνας, εξου και η ονομασία του αλγορίθμου.

Τα σημεία που έχουν προκύψει από τη χρήση μοντέλου SIFT μπορούν να χρησιμοποιηθούν σαν χαρακτηριστικά για την εκπαίδευση άλλων μοντέλων, όπως νευρωνικά δίκτυα, μοντέλα ταύτισης χαρακτηριστικών κ.α.

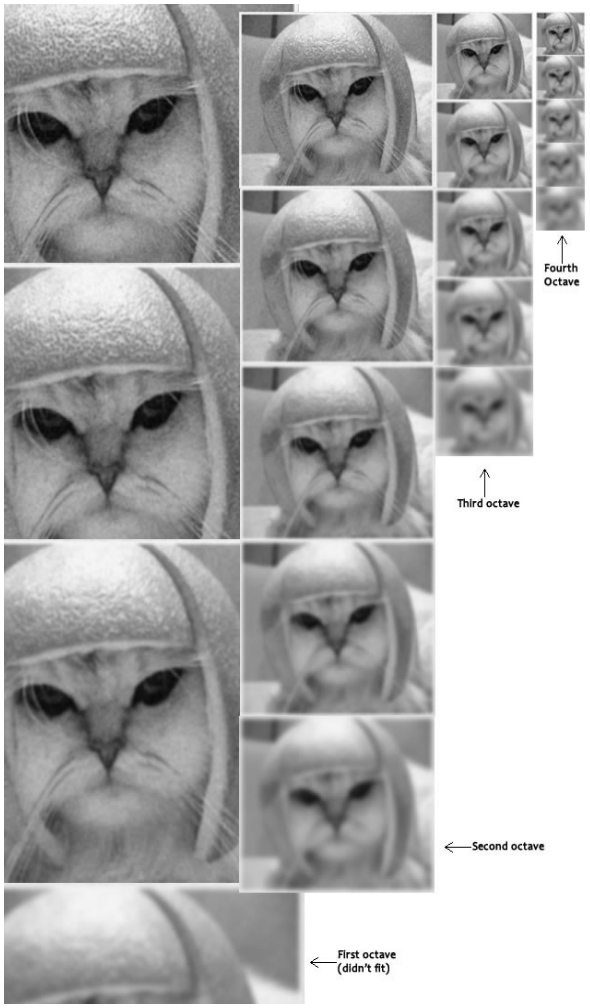
Στη συνέχεια θα δούμε 4 βασικά βήματα του αλγορίθμου και θα εξηγήσουμε συνοπτικά τις παραμέτρους με τις οποίες πειραματιστήκαμε.

2. Επιλογή κλίμακας (Scale-space Extrema Selection)

Τα αντικείμενα στην πραγματικότητα έχει νόημα να τα βλέπουμε από κάποια κλίμακα. Για παράδειγμα μπορούμε να δούμε εύκολα ένα στυλό πάνω σε ένα τραπέζι, αλλά προσπαθώντας να δούμε τον γαλαξία, τότε συγκριτικά το στυλό είναι σαν να μην υπάρχει. Έτσι και για τα keypoints. Το βήμα αυτό του αλγορίθμου υπάρχει, καθώς για να εντοπιστούν διαφορετικού μεγέθους keypoints, πρέπει να τα εξετάσουμε από διαφορετικές κλίμακες. Δηλαδή ένα μεγαλύτερο keypoint χρειάζεται να εξεταστεί υπο μεγαλύτερη κλίμακα.

Το scale-space, προσπαθεί να εισάγει αυτή την έννοια από τη φύση, στις ψηφιακές φωτογραφίες, στοχεύοντας στο να αναπαρασταθεί μια εικόνα σε διάφορες κλίμακες. Αυτό είναι το πρώτο βήμα αυτό του αλγορίθμου και αποτελείται απο τρία υποβήματα.

Ας ξεκινήσουμε με τη δημιουργία του scale – space. Μαθηματικά έχει αποδειχθεί ότι για την αλλαγή κλίμακας πρέπει μια εικόνα να περάσει μέσα από Gaussian Blur. Στο μοντέλο του SIFT η διαδικασία έχει ως εξής. Στο πρώτο στάδιο, ξεκινώντας απο την αρχική εικόνα, δημιουργούνται προοδευτικά Ν θολωμένες φωτογραφίες. Στο δεύτερο στάδιο, υποδιπλασιάζονται οι διαστάσεις της αρχικής εικόνας του πρώτου σταδίου (resized) και ύστερα δημιουργούνται προοδευτικά Ν θολωμένες φωτογραφίες. Στο τρίτο στάδιο υποδιπλασιάζονται οι διαστάσεις της αρχικής εικόνας του δεύτερου σταδίου και δημιουργούνται Ν θολωμένες εικόνες κ.ο.κ. Φωτογραφίες του ίδιου σταδίου, έχοντας τις ίδιες διαστάσεις, συντελούν μια οκτάβα. Όσο περισσότερο blur έχει υποστεί μια εικόνα στην οκτάβα, τόσο μεγαλύτερη και η κλίμακα.



Εικόνα 1: Δημιουργώντας το scale space με Gaussian Blur [3]

Η κλίμακα και ο αριθμός των οκτάβων εξαρτώνται από τις διαστάσεις της αρχικής εικόνας. Ωστόσο, σύμφωνα με τον D. Lowe 4 οκτάβες και 5 στρώματα blurring (N=5) η κάθε μια είναι ιδανικές για τη λειτουργία του αλγορίθμου.

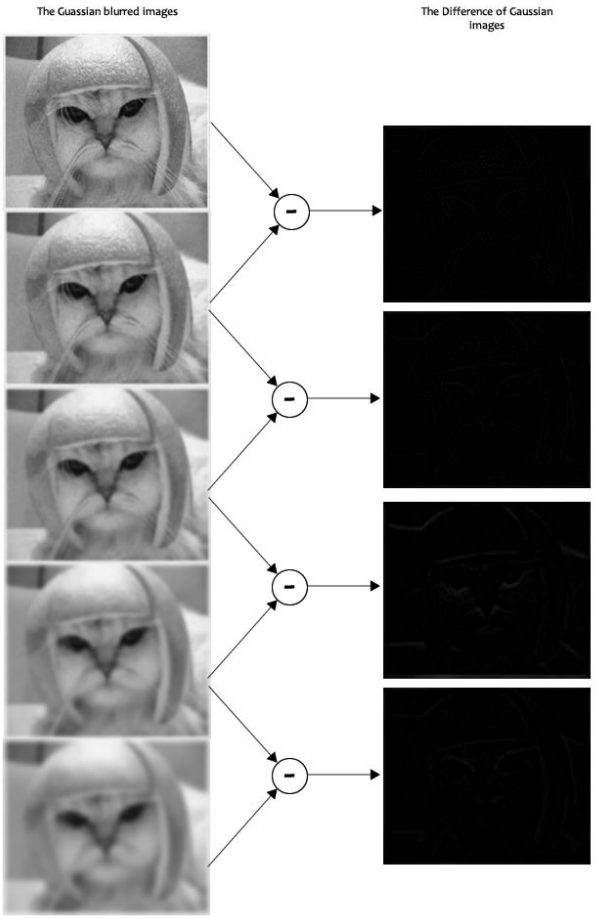
Όσον αφορά το θόλωμα των εικόνων, μαθηματικά ορίζεται ως η συνέλιξη ενός τελεστή Gauss και της εικόνας, ο οποίος εφαρμόζεται σε κάθε pixel.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Εξίσωση 1: Συνέλιξη της εικόνας με Gaussian Blur [3]

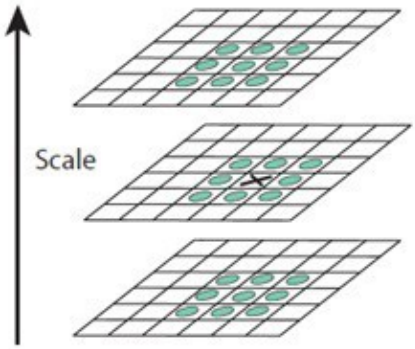
Όπου *G*, Gaussian Blur operator , *I* είναι η εικόνα, *x*,*y* είναι οι συντεταγμένες ενός σημείου και *σ* η παράμετρος για την κλίμακα ή το πόσο θα θολώσει η εικόνα στην οποία θα εφαρμοστεί το G. Όσο μεγαλύτερη η τιμή, τόσο πιο πολύ θα θολώσει. [3]

Εν συνεχεία, οι εικόνες αυτές θα χρησιμοποιηθούν για να βρεθεί ένα νέο σύνολο εικόνων, το Difference of Gaussian kernel. Το νέο αυτό σύνολο συμβάλλει στον εντοπισμό των keypoints. Ορίζεται ως η διαφορά μιας εικόνας που έχει υποστεί Gaussian Blurring με διαφορετικά *σ*. Η διαδικασία αυτή πραγματοποιείται για κάθε οκτάβα και βλέπουμε το αποτέλεσμα στην εικόνα 2. [3]



Εικόνα 2: Difference on Gaussians για μια οκτάβα [3]

Μέχρι τώρα έχει δημιουργηθεί ένας χώρος όπου η εικόνα βρίσκεται και σε διαφορετικές κλίμακες, ο οποίος χρησιμοποιήθηκε για να δημιουργηθεί ο χώρος των DoG φωτογραφιών. Πλέον, οι φωτογραφίες του τελευταίου εξετάζονται για τοπικά μέγιστα και ελάχιστα (local extrema). Για παράδειγμα ένα pixel συγκρίνεται με 8 γειτονικά του σε επόμενη κλίμακα και με 9 γειτονικά του σε προηγούμενη (εικόνα 3). Αν αποτελεί τοπικό μέγιστο ή ελάχιστο, αποτελεί και πιθανό key point και περισσότερο σημαίνει ότι αυτό το keypoint περιγράφεται καλύτερα στην συγκεκριμένη κλίμακα που βρίσκεται. Προτιμώνται οι ενδιάμεσες κλίμακες για τέτοιες διερευνήσεις, καθώς σε αυτές που βρίσκονται στην αρχή ή στο τέλος, δεν υπάρχουν πολλοί γείτονες για εξέταση. Για να είμαστε πιο συγκεκριμένοι, είναι “σχεδόν” ένα keypoint, γιατί ένα τοπικό μέγιστο ή ελάχιστο, συνήθως δεν βρίσκεται εξ ολοκλήρου σε ένα μόνο pixel, βρίσκεται σε περιοχή ανάμεσα σε pixel κάτι που μαθηματικά πρέπει να υπολογιστεί. [4]



Εικόνα 3: Εξέταση pixel για πιθανό keypoint [5]

3. Εντοπισμός Χαρακτηριστικών (Keypoint Localization)

Για τα keypoints που προέκυψαν από το προηγούμενο βήμα και είναι πολλά. Σε αυτό το βήμα πρέπει να υπολογιστούν οι περιοχές που προαναφέρθηκαν αλλά και να γίνει ένα “φιλτράρισμα” ώστε να κρατηθούν μόνο τα πιο δυνατά, καθώς κάποια βρίσκονται πάνω σε edge και άλλα δεν δημιουργείται αρκετή αντίθεση.

Χρησιμοποιώντας τις διαθέσιμες πληροφορίες για τα pixels, δημιουργούνται τιμές για τα subpixels. Αυτό συμβαίνει με επέκταση του θεωρήματος Taylor (όχι της σειράς Taylor) για το μέρος της εικόνας γύρω απο το πιθανό keypoint. [4]

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

Εξίσωση 2: Taylor expansion [4]

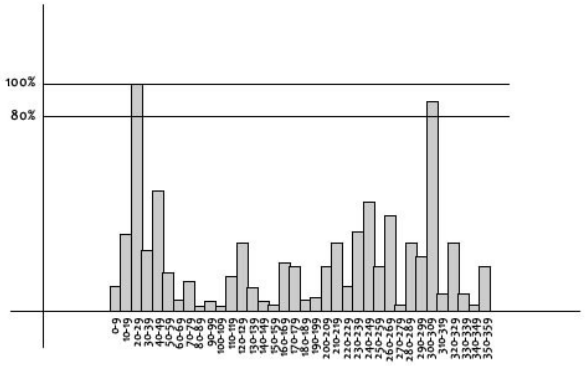
Την παραπάνω εξίσωση, αν την παραγωγίσουμε και την εξισώσουμε με μηδέν, μπορούμε να βρούμε τα τοπικά ακρότατα. Οι λύσεις της θα είναι οι τοποθεσίες του keypoint. Αν η ένταση του ακρότατου, είναι μικρότερη από κάποια τιμή (0.03 όπως λέγεται στο paper του Lowe, το keypoint απορρίπτεται. Αυτή η τιμή στο OpenCV, όπως θα δούμε παρακάτω ονομάζεται contrast Threshold.

Το DoG είναι πιο ανεκτικό στα edges, τα οποία ωστόσο θα πρέπει να απορριφθούν. Για να κριθεί αυτό θα χρησιμοποιηθεί ένας Hessian matrix διαστάσεων 2x2. [5] Από αυτή τη διαδικασία προκύπτουν 2 ιδιοδιανύσματα. Αν είναι και τα δύο είναι μεγάλα και είναι παρόμοια, τότε το σημείο είναι γωνία (corner) και το κρατάμε, ενώ αν το ένα σημείο είναι πολύ μεγαλύτερο από το άλλο, τότε πρόκειται για ακμή (edge) και το απορρίπτουμε. Η αναλογία αυτών των διανυσμάτων λέγεται edge Threshold στο OpenCV. [6]

4. Ανάθεση Προσανατολισμού σε Χαρακτηριστικό (Orientation Assignment)

Πλέον τα keypoints είναι έγκυρα, γνωρίζουμε και την κατάλληλη κλίμακα για το καθένα κι έχουμε εξασφαλίσει και την ανεξαρτησία ως προς την κλίμακα. Είναι απαραίτητο να βρεθεί ο προσανατολισμός ενός keypoint καθώς έτσι εξασφαλίζεται η ανεξαρτησία, που προσφέρει ο αλγόριθμος, κατά όποια πιθανή περιστροφή.

Επιλέγεται περιοχή γύρω απο το keypoint, της οποίας το μέγεθος εξαρτάται από την κλίμακα της εικόνας και λαμβάνει μέρος υπολογισμός των κατευθύνσεων των pixels που ανήκουν στην περιοχή. Δημιουργείται ένα ιστόγραμμα για αυτά τα στοιχεία.

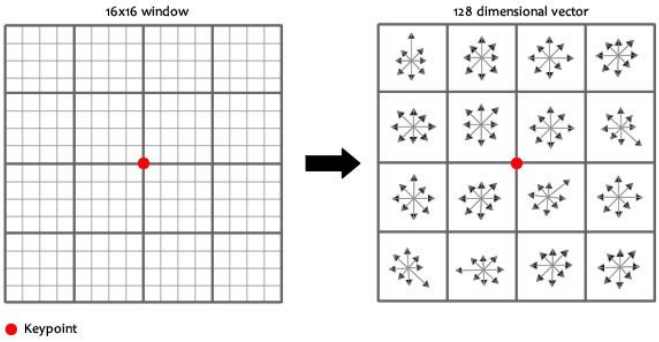


Εικόνα 4: Ιστόγραμμα του προσανατολισμού των επιμέρους στοιχείων της περιοχής [7]

Επιλέγεται η μεγαλύτερη κορυφή και οποιαδήποτε κορυφή ξεπερνά το 80% λαμβάνεται επίσης υπόψιν και δημιουργεί η κάθε μια ένα νέο keypoint με αυτό τον προσανατολισμό, αλλά με τοποθεσία και κλίμακα ίδια με αυτή του keypoint υπο εξέταση. [5]

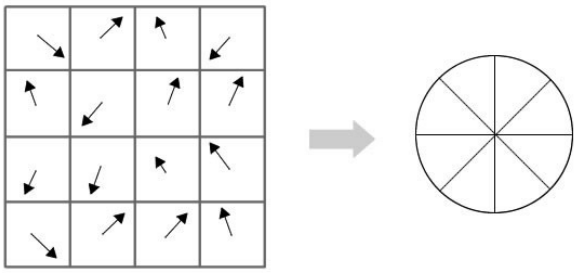
5. Περιγραφείς Χαρακτηριστικών (Keypoint Descriptor)

Κάθε keypoint χαρακτηρίζεται απο κλίμακα, τοποθεσία και προσανατολισμό. Παίρνουμε ένα παράθυρο 16x16 γύρω από το keypoint, το οποίο χωρίζεται σε 8 blocks 4x4 (εικόνα 5).



Εικόνα 5 [5]

Για κάθε block δημιουργείται ενα ιστόγραμμα για την εύρεση προσανατολισμού σε κάθε μπλοκ (εικόνα 6).



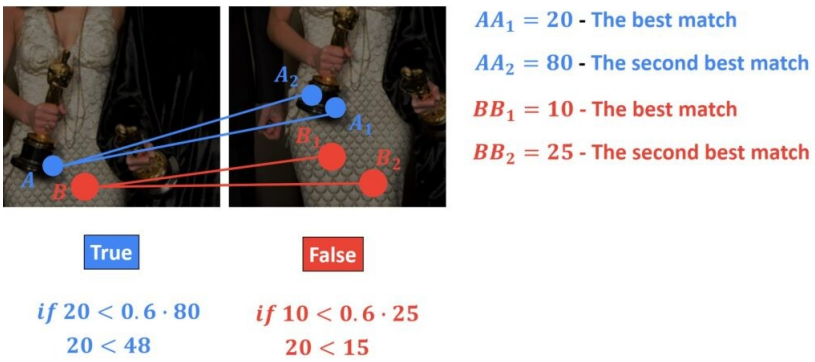
Εικόνα 6 [5]

Κι έτσι δημιουργήθηκε ένας περιγραφέας 4x4. Εφ'όσον το κάθε block έχει κι έναν προσανατολισμό, συνολικά υπάρχουν 8, θα προκύψουν 4x4x8=128 bin τιμές. Αυτές οι τιμές θα αναπαρασταθούν σαν διάνυσμα και συγκεκριμένα, το διάνυσμα που περιγράφει το χαρακτηριστικό (feature). Με κάποιες ακόμα διαδικασίες εξασφαλίζεται η ευρωστία του περιγραφέα όσον αφορά την όποια πιθανή περιστροφή, αλλαγή στη φωτεινότητα. [5], [6]

6. Ταίριασμα Χαρακτηριστικών (Keypoint Matching)

Για το ταίριασμα των keypoints δύο εικόνων εξετάζουμε τους πλησιέστερούς τους γείτονες. Σε κάποιες περιπτώσεις το δεύτερο καλύτερο ταίριασμα μπορεί να είναι πολύ κοντά στο πρώτο, λόγω θορύβου στις εικόνες ή άλλων παραγόντων. Για αυτό το λόγο ο

Lowe πρότεινε το παρακάτω ratio test, με στόχο να αυξηθεί η ευρωστία του αλγορίθμου. Αυτό θα γίνει απορρίπτοντας σημεία που δεν είναι αρκετά διακριτά. Η γενική ιδέα είναι ότι πρέπει να υπάρχει αρκετή διαφορά σε απόσταση, ανάμεσα στο πρώτο καλύτερο match και στα δευτερεύοντα matches.



Εικόνα 7: Παράδειγμα για το D. Lowe's ratio test [10]

Τότε υπολογίζεται ο λόγος των δύο αποστάσεων, της πρώτης καλύτερης απόστασης προς τη δεύτερη καλύτερη απόσταση. Αν είναι μεγαλύτερος από μια σταθερά που θα ορίσουμε εμείς (στην δημοσίευση η σταθερά αυτή είναι 0.7 - 0.8), δηλαδή οι αποστάσεις δεν έχουν αρκετή διαφορά μεταξύ τους, τα ταιριάσματα απορρίπτονται [10]. Σύμφωνα με τη δημοσίευση του Lowe αυτό περιορίζει τα λανθασμένα ταιριάσματα κατά 90%, ενώ απορρίπτει μόλις το 5% των σωστών. [5]

7. Παράμετροι SIFT στο OpenCV

Το μοντέλο στο OpenCV της Python δίνει τη δυνατότητα στον χρήστη να έχει πρόσβαση σε αυτές τις 5 παραμέτρους. Δίνεται και η default τιμή για κάθε μια απο αυτές. [8]

a) nfeatures = 0 : Ο μέγιστος αριθμός χαρακτηριστικών που μπορούν να κρατηθούν στα αποτελέσματα. Κρίνονται με βάση τη βαθμολογία τους, στην περίπτωση του SIFT με την αντίθεση που υπάρχει τοπικά.
b) nOctaveLayers = 3 : Αριθμός στρωμάτων ανα οκτάβα. Ο αριθμός 3 προκύπτει από το paper του D. Lowe. Ο αριθμός των οκτάβων υπολογίζεται αναλόγως την ανάλυση της εικόνας.
c) contrastThreshold = 0.04 : Αυτή η παράμετρος φιλτράρει για να αποκλείσει τα αδύναμα χαρακτηριστικά σε περιοχές με χαμηλή αντίθεση.
d) edgeThreshold = 10 : Χρησιμοποιείται για να φιλτράρει και να αποκλείσει χαρακτηριστικά που αποτελούν ακμές (edges). Όσο μεγαλύτερο είναι, τόσο λιγότερα χαρακτηριστικά φιλτράρονται. Η προκαθορισμένη τιμή προκύπτει από το paper του D. Lowe.
e) sigma = 1.6 : Το σίγμα του Gauss εφαρμόζεται στην εικόνα που δίνεται στο μοντέλο στην πρώτη οκτάβα. Αν η εικόνα έχει χαμηλή ανάλυση, θα χρειαστεί να μειωθεί η τιμή της παραμέτρου.

Παρακάτω θα σχολιάσουμε τι συμβαίνει στο μοντέλο και κυρίως στα αποτελέσματα όταν αυτές μεταβάλλονται.

IV. Επεξεργασία Εικόνων

Για τις ανάγκες της εργασίας χρησιμοποιήθηκαν φίλτρα που εφαρμόστηκαν στις εικόνες, με σκοπό απο μια εικόνα να προκύψει ένα query set εικόνων. [9] Όπως σε μονοδιάστατα σήματα, έτσι και οι εικόνες μπορούν να περάσουν μέσα από βαθυπερατά ή υψιπερατά φίλτρα. Τα μεν συμβάλλουν στην απομάκρυνση του θορύβου από την εικόνα, τα δε στο να εντοπίζονται πιο εύκολα όποια edges. Στην εργασία αυτή χρησιμοποιήθηκαν χαμηλοπερατά φίλτρα.

1. Filtering : Εφαρμόστηκε με συνέλιξη πυρήνας 5x5.

2. Averaging : Ακολουθεί την ίδια διαδικασία με το προηγούμενο, επίσης, με στόχο την απομάκρυνση του θορύβου. Χρησιμοποιεί κανονικοποιημένο box filter.

3. Gaussian Blurring : Με χρήση γκαουσιανού πυρήνα 5x5. Πολύ αποτελεσματικό στην απομάκρυνση γκαουσιανού θορύβου.

4. Median Blurring : Απομακρύνει τον θόρυβο πιο αποτελεσματικα, συγκεκριμένα τον θόρυβο του τύπου “salt-and-pepper”.

5. Bilateral Filtering : Πολύ αποτελεσματικό για τον θόρυβο, ενώ παράλληλα κρατάει τα αιχμηρά χαρακτηριστικά, ωστόσο πιο αργό στον υπολογισμό του από άλλα φίλτρα.

Το 1 ανήκει στην κατηγορία του image filtering ενώ τα υπόλοιπα σε αυτήν του image blurring, παρ’ όλα αυτά έχουν τον ίδιο στόχο. Η φωτογραφία με την οποία επιλέχθηκε να διεξαχθεί ένα πείραμα θα περάσει από το κάθε φίλτρο μια αλλά και δύο φορές.

V. Σχολιασμός Αποτελεσμάτων

Σε αυτό το κεφάλαιο θα αναφερθούν τα αποτελέσματα των πειραμάτων και των εξόδων του κώδικα και θα σχολιαστούν συνοπτικά. Θα ξεκινήσουμε πρώτα με τα πειράματα που έγιναν πάνω στις παραμέτρους, τις συνέπειες και τα αποτελέσματα και ύστερα θα αναφερθουμε σε κάποια από τα πειράματα που έγιναν στα αποτελέσματα ζευγαριών μοντέλων με διαφορετικές παραμέτρους, στην προσπάθεια να βγάλουν παρόμοια αποτελέσματα.

1. Πειράματα με τις παραμέτρους του μοντέλου SIFT

Για να πειραματιστούμε με κάποια παράμετρο, κρατήθηκαν σταθερές στις προκαθορισμένες τους τιμές όλες οι υπόλοιπες. Τα διαγράμματα που θα δούμε, εκτυπώνονται και αποθηκεύονται σε φάκελο από το πρόγραμμα, ανάλογα με τη βούληση του χρήστη. Οι φωτογραφίες που είναι σχεδιασμένες με τα keypoints και τα ζεύγη με τα matches τους αποθηκεύονται από το πρόγραμμα, επίσης αν το επιθυμεί ο χρήστης, σε αντίστοιχους φακέλους. Όλα αυτά γίνονται από το script sift.py.

Ο όρος success rate, είναι κάτι που έχω σχεδιαστεί για να βγούν κάποια συμπεράσματα όσον αφορά τη συμπεριφορά του μοντέλου, όχι κάτι που χρησιμοποιείται ευρέως για την αξιολόγηση του μοντέλου. Το συνολικό ποσοστό

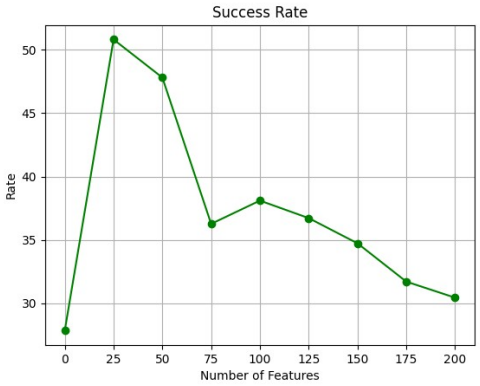
επιτυχίας ορίζεται σαν το μέσο όρο των ποσοστών επιτυχίας για κάθε ζεύγος φωτογραφιών, ενώ ως ποσοστό επιτυχίας για κάθε ζεύγος, ορίζω ως:

success_rate_of_matching = (good_matches / total_matches) * 100

Το feature matching επιστρέφει ζεύγη keypoints που ταιριάζουν ανάμεσα σε δύο φωτογραφίες. Τα αποδεκτά ζευγάρια (good matches) υπολογίζονται με ενα ratio test από το paper του D. Lowe. Κάθε keypoint από την πρώτη εικόνα, θα ταιριάξει με έναν αριθμό keypoints της δεύτερης εικόνας. Στην προκειμένη περίπτωση με 2 το πολύ, καθώς χρησιμοποιείται συνάρτηση για πλησιέστερους γείτονες με k=2. Κρατάμε τα δύο καλύτερα ταιριάσματα, από άποψη απόστασης. Το test αυτό εξετάζει αν αυτές οι δύο αποστάσεις είναι σημαντικά διαφορετικές. Αν δεν είναι, τότε το keypoint της δεύτερης εικόνας απορρίπτεται και δεν λαμβάνεται υπόψιν για άλλους υπολογισμούς.

a) Number of features

Η default τιμή της παραμέτρου είναι 0, δηλαδή το μοντέλο, αν δεν πειράξουμε αυτή την παράμετρο δεν θέτει φράγμα στο πλήθος των χαρακτηριστικών που μπορεί να εντοπιστούν και να επιστραφούν. Στη συνέχεια θα δούμε ένα διάγραμμα με το ποσοστό επιτυχίας συναρτήσει του πλήθους των keypoints που βρέθηκαν.



Διάγραμμα 1

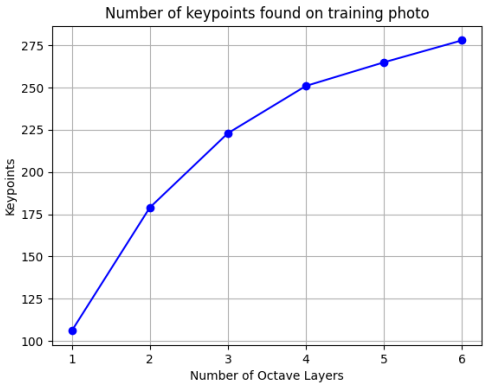
Ενώ μειώνονται τα keypoints που εντοπίστηκαν στην φωτογραφία, τα matches δεν μειώνονται τόσο, άρα το κλάσμα είναι μεγαλύτερο όταν ο περιορισμός των χαρακτηριστικών δεν είναι πολύ μικρός (μικρή τιμή παραμέτρου), ούτε πολύ μεγάλος (μεγάλη τιμή παραμέτρου ή μηδενισμός της)

b) Number of octave layers

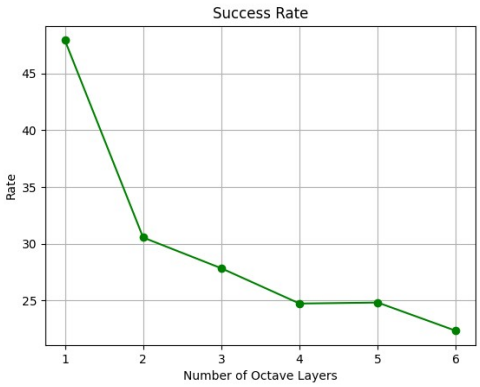
Ενώ στην προηγούμενη ενότητα δεν υπήρχε νόημα να δούμε διάγραμμα για τη μεταβολή του πλήθους χαρακτηριστικών που προκύπτουν από τον πειραματισμό με την παράμετρο, καθώς το πλήθος χαρακτηριστικών που επιστρεφόταν το ορίζαμε εμείς, από εδώ και πέρα θα το βλέπουμε.

Παρατηρούμε ότι όσο τα octave layers αυξάνονται, αυξάνονται και τα keypoints (διάγραμμα 2.α), κάτι αναμενόμενο αν και όσο περισσότερα στρώματα, τόσο πιο πολύ

κοστίζει υπολογιστικά η διαδικασία. Ωστόσο, το ποσοστό επιτυχίας πέφτει καθώς μαζί με την αύξηση των keypoints δεν έχουμε ανάλογη αύξηση των matches.



Διάγραμμα 2.α

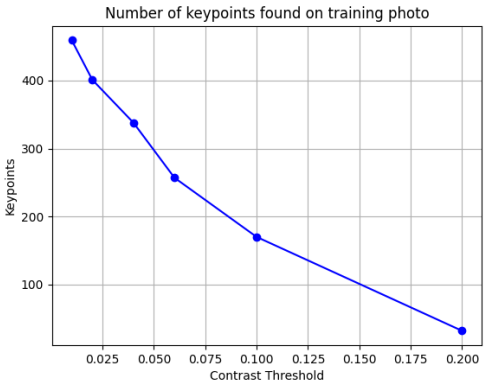


Διάγραμμα 2.β

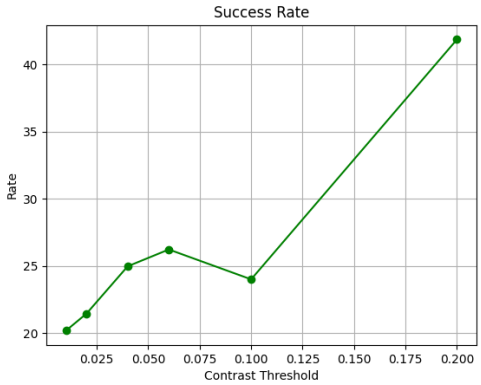
c) Contrast threshold

Αφού βρεθούν πιθανά keypoints, φιλτράρονται για την καταλληλότητά τους. Όπως εξηγήθηκε παραπάνω, ένα από τα κριτήρια είναι η αντίθεση που υπάρχει σε κάποιο keypoint, η οποία αν βρίσκεται κάτω από ένα threshold, το keypoint απορρίπτεται. Το contrastThrseshold αντιπροσωπεύει αυτό το όριο. Ενώ η τιμή της παραμέτρου αυξάνεται, φιλτράρονται και απορρίπτονται όλο και περισσότερα σημεία, καθώς για να γίνουν αποδεκτά πρέπει να υπάρχει στο σημείο μεγαλύτερη αντίθεση από το όριο, οπότε τελικά επιστρέφονται λιγότερα κι αυτό φαίνεται στο διάγραμμα 3.α. Το φιλτράρισμα δηλαδή γίνεται πιο αυστηρό όσο η τιμή της παραμέτρου μεγαλώνει.

Αντίθετα, το ποσοστό επιτυχίας ανεβαίνει. Ενώ σαν αποτέλεσμα είναι επιθυμητό, δεν είναι αποδεκτός ο τρόπος με τον οποίο επιτυγχάνεται, καθώς το ιδανικό σενάριο θα ήταν να έχουμε υψηλά ποσοστά επιτυχίας σε συνδυασμό με την εύρεση αρκετών χαρακτηριστικών, κάτι που θα σήμαινε ότι θα είχαμε αρκετά matches. Αυτό βέβαια εξαρτάται και από τις παραμέτρους του Brute Force Matcher που χρησιμοποιήθηκε, άλλα δεν έγιναν πειράματα με εκείνες.



Διάγραμμα 3.α

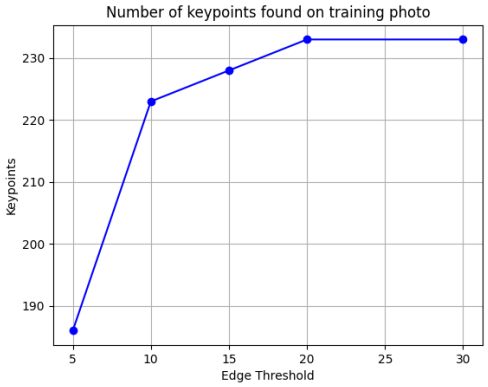


Διάγραμμα 3.β

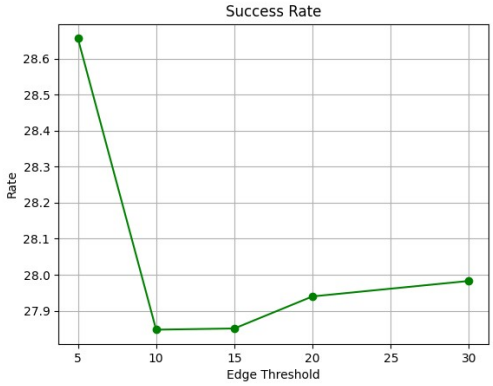
d) Edge threshold

Το edge Threshold αποτελεί όριο για την αναλογία 2 ιδιοδιανυσμάτων που προκύπτουν κατά το φιλτράρισμα keypoints που βρίσκονται πάνω σε ακμές (edges), ώστε αυτά να απορριφθούν. Όσο μεγαλύτερο το όριο, τόσο πιο δύσκολο να απορριφθούν πιθανά keypoints. Άρα το κριτήριο απόρριψης χαλαρώνει. Γι'αυτό, όπως βλέπουμε και στο διάγραμμα 4.α, όσο το threshold αυξάνεται, τόσο αυξάνονται και τα keypoints. Επειδή πρόκειται για όριο σε αναλογία δεν βλέπουμε μεγάλη αύξηση.

Το ποσοστό επιτυχίας μειώνεται όσο αυξάνεται η παράμετρος (διάγραμμα 4.β), καθώς περνούν keypoints, τα οποία δεν είναι απολύτως έγκυρα καθώς δεν είναι κατάλληλα για τον αλγόριθμο. Και σε αυτή την περίπτωση η μείωση είναι αμελητέα, γι'αυτό αργότερα δεν θα πειράξουμε την τιμή αυτης της παραμέτρου, ή απλά μπορεί να την μειώσουμε ελάχιστα.



Διάγραμμα 4.α

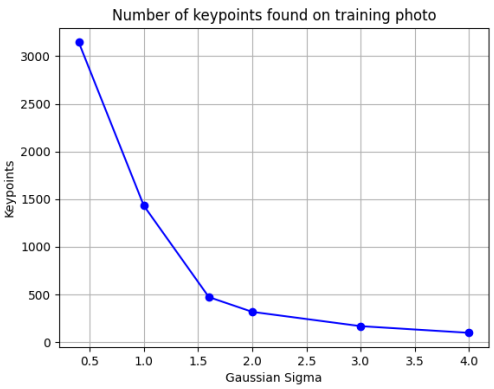


Διάγραμμα 4.β

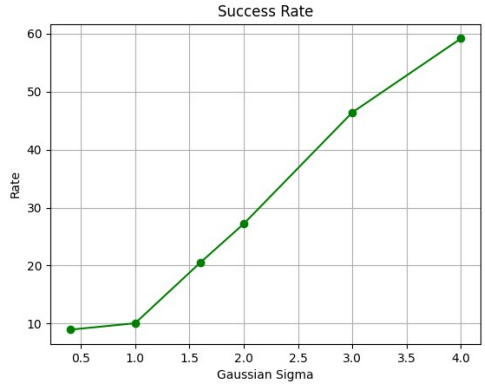
e) Gaussian sigma

Η τιμή της παραμέτρου εξαρτάται περισσότερο από την ποιότητα της φωτογραφίας. Για μια συγκεκριμένη φωτογραφία βλέπουμε ότι τα keypoints πέφτουν σημαντικά καθώς αυξάνουμε την τιμή. Αν παρατηρήσουμε το διάγραμμα (διάγραμμα 5.α), μέχρι να φτάσουμε στο 1.6, που είναι η τιμή που προτείνει το paper του Lowe να χρησιμοποείται, τα keypoints μειώνονται δραματικά.

Στο διάγραμμα 5.β βλέπουμε το ποσοστό επιτυχίας να αυξάνεται, όμως ισχύει το ίδιο με το ποσοστό επιτυχίας της παραμέτρου contrast Threshold πιο πάνω.



Διάγραμμα 5.α



Διάγραμμα 5.β

2. Αποτελέσματα ταιριασμάτων μεταξύ διαφορετικών μοντέλων

Σε αυτή την ενότητα θα σχολιαστούν κάποια αποτελέσματα πειραμάτων με το script compare.py. Θα επιλεγεί μια φωτογραφία για όλα τα πειράματα, τυχαία επιλέγω την 5 που σύμφωνα με το ReadMe έχει υποστεί μια φορα Gaussian blur, με αρχική εικόνα αυτή του δένδρου. Στόχος είναι να βρεθούν όσα περισσότερα matches γίνεται ανάμεσα σε αυτήν και την αρχική.

a) Δίνουμε deafult τιμές και στο δεύτερο μοντέλου

Τα δύο μοντέλα που δημιουργούνται έχουν τις ίδιες παραμέτρους. Είναι αναμενόμενο να μην έχουμε αρκετά matches εταξύ των δυο εικόνων.

Keypoints στην αρχική φωτογραφία

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 165 out of 473

Θα προσπαθήσουμε να εντοπίσουμε περισσότερα ζεύγη keypoints

b) Μειώνουμε το Gaussian sigma

Ενώ θεωρητικά, αν δουλεύουμε με ήδη θολωμένες φωτογραφίες θα ήταν καλό να χρησιμοποιούμε μικρότερες τιμές sigma, το δοκιμάσαμε και δεν είχαμε περισσότερα matches απο πριν. Με sigma = 0.8 :

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 119 out of 473

c) Αυξάνουμε το sigma

Με μια μικρή αύξηση του sigma έχουμε λιγότερη μείωση στα matches, ενώ αν η αύξηση ήταν μεγαλύτερη πάλι τα matches μειώνονται αρκετά. Για sigma = 2 :

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 150 out of 473

d) Μειώνουμε στο ελάχιστο το contrastThreshold

Μειώνοντας το contrastThreshold, όπως είναι αναμενόμενο βρίσκουμε περισσότερα keypoints. Τα matches αυξήθηκαν επίσης, συνεπώς είναι ένα σχετικά καλό αποτέλεσμα. Για contrastThreshold = 0.02:

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 194 out of 473

e) Αυξάνουμε ελάχιστα το contrastThreshold

Αυξάνουμε το threshold και όπως είναι αναμενόμενο θα βρεθούν λιγότερα keypoints. Λιγότερα keypoints προς σύγκριση, λιγότερα matches. Για contrastThreshold = 0.07:

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 118 out of 473

f) Μειώνουμε τα OctaveLayers

Μειώσαμε τα octave layers κατα 1 και η διαφορά στα matches δεν ήταν θεαματική, βρέθηκαν 161 matches. Μειώσαμε κατα 2 και τα matches μειώθηκαν κατα πολύ, όπως και τα keypoints της εικόνας 5. Για nOctaveLayers = 1 :

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 107 out of 473

g) Αυξάνουμε τα OctaveLayers

Αυξήσαμε τα στρώματα ανα οκτάβα και τα matches μειώθηκαν. Όσο αυξάνουμε τα στρώματα τόσο μειώνονται τα matches. Ενδεικτικά για nOctaveLayers = 5:

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 139 out of 473

h) Συνδυασμός αλλαγής παραμέτρων

Δεν θα κάνουμε πειράματα με τις παραμέτρους nfeatures και edgeThreshold. Η πρώτη περιορίζει το πλήθος χαρακτηριστικών κι εμείς θέλουμε να βρούμε όσα η αρχική εικόνα, ίσως και λίγα περισσότερα και η δεύτερη, όπως είδαμε στα πειράματα, ακόμα κι αν διπλασιστεί, τριπλασιαστεί, υποδιπλασιαστεί, γενικά αλλάξει κατα πολύ, δεν προσφέρει κάποια σοβαρή αλλαγή στα αποτελέσματα, πόσο μάλλον βελτίωση.

Άλλαξαμε 3 παραμέτρους και έθεσα nOctaveLayers = 4, contrastThreshold = 0.03, sigma = 1.5. Ενώ φάνηκε να βρέθηκαν πολλά keypoints στην εικόνα 5, δεν

βρέθηκαν πολλά παραπάνω matches απο ότι αν δεν είχαμε αλλάξει καμία παράμετρο και είχαμε δώσει σε όλες την default τιμή τους.

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 167 out of 473

Δοκιμάσαμε επίσης να αλλάξουμε μόνο 2 παραμέτρους. Με contrastThreshold = 0.02 και sigma = 1.2:

Keypoints στην εικόνα 5

Matches ανάμεσα στις 2 φωτογραφίες

Output:
Matches found on initial and query photo: 159 out of 473

Γενικά με αυτό το contrast threshold και με sigma απο 1.2 εώς 1.7 τα matches κυμαίνονται σε αυτή την τιμή.

VI. Επίλογος

Το μοντέλο SIFT εξηγήθηκε και είναι ένα πολύ χρήσιμο μοντέλο για την κατηγορία του. Έγιναν πειράματα με τις παραμέτρους που προσφέρει το OpenCV στον χρήστη για αλλαγή, συγκρίθηκαν τα αποτελέσματα του μοντέλου με τις default παραμέτρους με αυτά άλλων μοντέλων, με διαφορετικές παραμέτρους. Ωστόσο δεν καταφέραμε σε ένα ζεύγος εικόνων να βρούμε τα περισσότερα δυνατά ταιριάσματα, αντ' αυτού καταλήξαμε να βρίσκουμε περίπου τα ίδια ταιριάσματα είτε δεν αλλάζαμε καθόλου τις παραμέτρους, είτε με κάποιους θεωρητικά καλούς συνδυασμούς, όπως στο τελευταίο πείραμα.

VII. Βιβλιογραφία – Πηγές

(1) Feature (computer vision) - Wikipedia
(2) Scale-invariant feature transform - Wikipedia
(3) SIFT: Theory and Practice: The scale space - AI Shack
(4) SIFT: Theory and Practice: Finding keypoints - AI Shack
(5) Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium
(6) OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform)
(7) SIFT: Theory and Practice: Keypoint orientations - AI Shack
(8) OpenCV: cv::SIFT Class Reference
(9) OpenCV: Smoothing Images
(10) Feature Matching