

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



## Πτυχιακή εργασία

Μελέτη και υλοποίηση σε λογισμικό ανιχνευτών σημειακών χαρακτηριστικών από εικόνες (point feature detectors), για την αναγνώριση αντικειμένων σε εφαρμογές μηχανικής όρασης

Μπαδέκα Ευτυχία (ΑΕΜ 1037)

Επιβλέπων Καθηγητής: Δρ. Ιωάννης Καλόμοιρος

Σέρρες 2015

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

## **Πτυχιακή εργασία**

Μελέτη και υλοποίηση σε λογισμικό ανιχνευτών σημειακών χαρακτηριστικών από εικόνες (point feature detectors), για την αναγνώριση αντικειμένων σε εφαρμογές μηχανικής όρασης

Μπαδέκα Ευτυχία (ΑΕΜ 1037)

Επιβλέπων Καθηγητής: Δρ. Ιωάννης Καλόμοιρος

Σέρρες 2015

## Περίληψη

Στην εργασία αυτή γίνεται αρχικά, σύντομη εισαγωγή στη ψηφιακή επεξεργασία εικόνας και ειδικότερα, γίνεται αναφορά σε τεχνικές εξαγωγής χαρακτηριστικών σημείων (keypoints) από εικόνες.

Στη συνέχεια, γίνεται η μελέτη ενός ανιχνευτή εξαγωγής χαρακτηριστικών SIFT, δηλαδή χαρακτηριστικών σημείων αναλλοίωτων ως προς την κλιμάκωση (scale invariant feature transform). Στη συνέχεια, ο αλγόριθμος υλοποιείται στην πράξη με αναλυτικό τρόπο, στο περιβάλλον του Matlab, που παρέχει πολλά μαθηματικά εργαλεία για την επεξεργασία της εικόνας. Η εργασία στηρίζεται στη μελέτη και υλοποίηση του D. Lowe. Τα στάδια του αλγόριθμου εφαρμόζονται βήμα-βήμα. Η τελική εφαρμογή που αναπτύχθηκε μπορεί να κάνει την ανίχνευση χαρακτηριστικών σημείων σε μία κλίμακα και σε μία οκτάβα. Επίσης, εξάγονται οι περιγραφείς (descriptors) των χαρακτηριστικών σημείων και με βάση αυτούς γίνεται η αντιστοίχιση των χαρακτηριστικών σημείων που εξάγονται από δύο εικόνες, που διαφέρουν μεταξύ τους στην κλίμακα και στη γωνία λήψης. Τέλος, παρατίθεται ο συνολικός αλγόριθμος με την επεξήγησή του και τα συμπεράσματα.

# Περιεχόμενα

Περίληψη .....	2
Εισαγωγή .....	5
<b>1. Εισαγωγή στην ψηφιακή εικόνα .....</b>	<b>7</b>
1.1 Η εικόνα και η φωτογραφική μηχανή .....	7
1.2 Ψηφιακή φωτογραφία .....	7
1.3 Χρωματικά μοντέλα .....	8
1.4 Εικόνα αποχρώσεων του γκρι (Grayscale) .....	9
1.5 Ιστόγραμμα .....	10
1.6 Προβλήματα ψηφιακής εικόνας .....	11
1.6.1 Θόρυβος .....	11
<b>2. Ψηφιακή επεξεργασία εικόνας.....</b>	<b>12</b>
2.1. Βασικές λειτουργίες στην ψηφιακή επεξεργασία εικόνας .....	12
2.2 Αντίθεση.....	13
2.3 Συνέλιξη .....	13
2.4 Φίλτρο Gauss .....	14
<b>3. Εξαγωγή χαρακτηριστικών .....</b>	<b>16</b>
3.1 Εισαγωγή.....	16
3.1.1 Παραδείγματα αναγνώρισης αντικειμένων .....	17
3.2 Ανιχνευτές Χαρακτηριστικών.....	18
3.2.1 Χαρακτηριστικά (features).....	18
3.2.2 Ανίχνευση Χαρακτηριστικών .....	18
3.2.3 Αναλλοίωτα (invariant) χαρακτηριστικά .....	19
3.3 Η έννοια της χωροκλίμακας (Scale Space) .....	19
3.4.1 Η δημιουργία της χωροκλίμακας με φίλτρο Gauss .....	20
3.4.2 Πυραμίδα Gauss .....	21
3.5 Σχετική έρευνα.....	21
3.6 Αλγόριθμοι εξαγωγής χαρακτηριστικών.....	23
RIFT (rotation-invariant Feature Transform) .....	23
SURF (Speeded Up Robust Features) .....	23
SPIN (Sorting points into neighborhoods) .....	24
Harris.....	24
SUSAN (Smallest Univalued Segment Assimilating Nucleus) .....	24
Harris-Laplace .....	24

<b>4. Ανιχνευτής SIFT (Scale Invariant Feature Transform)- Μετασχηματισμός αναλλοίωτων ως προς την κλίμακα χαρακτηριστικών σημείων.....</b>	<b>25</b>
4.1 Εισαγωγικά.....	25
4.2 Ιδιότητες.....	26
1) Ανίχνευση ακρότατων στο χώρο της κλιμάκωσης (Scale-space extrema) .....	26
2) Εντοπισμός σημείων-κλειδιών (Patch Localization) .....	26
3)Ανάθεση προσανατολισμού (Orientation Assignment) .....	26
4) Περιγραφή των σημείων-κλειδιών (Patch Description) .....	27
4.2.1 Προβλήματα και πλεονεκτήματα του SIFT .....	28
4.3 Στάδια του ανιχνευτή SIFT .....	29
4.3.1 Εύρεση της χωροκλίμακας.....	29
4.3.2 Εύρεση τοπικού μέγιστου (Keypoint Localization) .....	32
4.3.3 Ανάθεση προσανατολισμού (Orientation Assignment) .....	33
4.3.4 Περιγραφέας.....	35
4.3.5 Αναπαράσταση Περιγραφέα .....	36
<b>5. Δημιουργία Εφαρμογής Αλγόριθμου SIFT .....</b>	<b>38</b>
5.1 Μαθηματικά προγράμματα .....	38
5.2 Matlab.....	38
5.3 Διάγραμμα Ροής.....	40
5.4 Λειτουργία του προγράμματος.....	41
<b>Συμπεράσματα.....</b>	<b>50</b>
<b>Παράρτημα Κώδικα .....</b>	<b>53</b>

## Εισαγωγή

Η εργασία αυτή είναι μια μελέτη και υλοποίηση σε λογισμικό ανιχνευτών σημειακών χαρακτηριστικών από εικόνες για την αναγνώριση αντικειμένων σε εφαρμογές ρομποτικής όρασης, βασισμένη στον ανιχνευτή SIFT (Scale Invariant Feature Transform). Ξεκινάει με τις εισαγωγικές έννοιες της εικόνας στην ψηφιακή εποχή, εξηγεί σύντομα τι πετυχαίνουμε με την ψηφιακή επεξεργασία της εικόνας και επικεντρώνεται σε μια τεχνική με την οποία εξάγουμε την απαραίτητη πληροφορία, ώστε να μπορούμε να κάνουμε αναγνώριση αντικειμένων.

Στην αρχή, υπάρχει αναφορά στην ψηφιακή εικόνα. Γίνεται μια περιγραφή της εικόνας στο ψηφιακό περιβάλλον του υπολογιστή και μια αναφορά σε τεχνικές που μπορούν να εφαρμοστούν με τη βοήθεια απλών προγραμμάτων επεξεργασίας εικόνας. Στη σημερινή εποχή, τέτοιες εφαρμογές είναι διαδεδομένες ακόμη και στα κινητά τηλέφωνα (smartphones). Επίσης, γίνεται μια εισαγωγή στη ρομποτική όραση ως επιστήμη που συνδέεται με την ψηφιακή επεξεργασία εικόνας. Δίνονται ορισμένα παραδείγματα πάνω σε αυτό και παρουσιάζονται κάποιες εφαρμογές. Τέλος, γίνεται μια συζήτηση για τα βασικά πλεονεκτήματα της ρομποτικής όρασης και εξηγείται ποιος είναι ο σκοπός της εξέλιξης αυτής της επιστήμης.

Στη συνέχεια, γίνεται αναφορά σε βασικούς αλγόριθμους εξαγωγής χαρακτηριστικών σημείων και αναγνώρισης αντικειμένων, που έχουν προταθεί και έχουν εξελιχθεί την τελευταία δεκαετία. Εξηγείται τι είναι ο χώρος της κλιμάκωσης, γιατί υποστηρίζεται αυτή η μέθοδος, τι ιδιότητες έχει και ποια είναι τα αμετάβλητα χαρακτηριστικά που βοηθάνε στην αναγνώριση αντικειμένου, ανεξάρτητα από το φωτισμό και την περιστροφή. Μέσα από όλη αυτή τη μελέτη και την αναφορά στους ανιχνευτές, η εργασία μας καταλήγει στο βασικό αντικείμενο της μελέτης, δηλαδή στον αλγόριθμο του μετασχηματισμού που παράγει αναλλοίωτα χαρακτηριστικά στο χώρο της κλιμάκωσης, δηλαδή στον μετασχηματισμό SIFT.

Γίνεται μια εκτενέστερη ανάλυση στα στάδια του αλγόριθμου, στα πλεονεκτήματα του και δίνεται με λεπτομέρεια η λειτουργία του ανιχνευτή. Η λειτουργία του έχει κατανοηθεί μέσα από την υλοποίηση του στο προγραμματιστικό περιβάλλον του προγράμματος Matlab. Η τελική εφαρμογή δείχνει τελικά πως λειτουργεί ένας

ανιχνευτής αντικειμένων. Ο ανιχνευτής αυτός λειτουργεί έτσι ώστε να γίνεται μια ταύτιση χαρακτηριστικών σε εικόνα που έχει αλλαχθεί ο προσανατολισμός της. Είναι επίσης δυνατό να ανιχνευθεί αντικείμενο σε εικόνα που έχει σε διαφορετική κλίμακα από την αρχική. Επίσης, είναι δυνατό να εξαχθούν γνωρίσματα, όπως οι προσανατολισμοί των αντικειμένων. Ένας τέτοιος ανιχνευτής έχει σαν βάση τη εξαγωγή κατάλληλων περιγραφέων που χρησιμεύουν στην ταυτοποίηση αντικειμένων.

# **1. Εισαγωγή στην ψηφιακή εικόνα**

## **1.1 Η εικόνα και η φωτογραφική μηχανή**

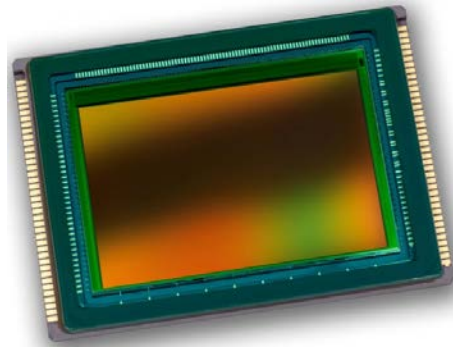
Η εικόνα συνδέεται στενά με τη φωτογραφία. Αποτελεί ένα πολύ καλό εργαλείο αποτύπωσης της πραγματικότητας και είναι το αποτέλεσμα της καταγραφής μιας φωτογραφικής μηχανής. Εδώ και πάνω από έναν αιώνα υπάρχει η αναλογική φωτογραφία, που ακόμη και σήμερα θεωρείται ότι είναι καλύτερη από την ψηφιακή, γιατί δεν έχει πρόβλημα με την ανάλυση, όπως έχει σε πολλές περιπτώσεις η ψηφιακή φωτογραφία.

Η κάθε φωτογραφική μηχανή έχει τρεις βασικές ρυθμίσεις: εστίαση, διάφραγμα και χρόνος έκθεσης. Το διάφραγμα είναι ένας σημαντικός παράγοντας στην πληροφορία της εικόνας, γιατί σχετίζεται και με την απόσταση του αντικειμένου. Το αποτέλεσμα του μεγάλου διαφράγματος είναι μια φωτογραφία που έχει θολά τα αντικείμενα του υποβάθρου, ενώ όταν κλείνει το διάφραγμα παίρνει λιγότερο φως, αποδίδοντας μια εικόνα πιο καθαρή.

## **1.2 Ψηφιακή φωτογραφία**

Στην ψηφιακή εποχή, ότι υπάρχει σε αναλογική μορφή, μετατρέπεται με μία δειγματοληψία σε ψηφιακή μορφή. Πλέον, οι ψηφιακές φωτογραφικές μηχανές έχουν αντικαταστήσει όλη τη διαδικασία της λήψης και της εμφάνισης με το μηχανισμό μιας σύγχρονης φωτογραφικής μηχανής. Το μόνο που δεν έχει αλλάξει λειτουργικά είναι ο φακός. Το αναλογικό φιλμ έχει αντικατασταθεί από τον αισθητήρα όπου είναι το βασικό μέσο απεικόνισης σε μια φωτογραφική μηχανή, και το οποίο είναι υπεύθυνο για την ανάλυση και την ποιότητα της εικόνας που λαμβάνεται. Οι αισθητήρες που χρησιμοποιούνται στις περισσότερες ψηφιακές φωτογραφικές μηχανές είναι CCD ή CMOS. Οι CMOS είναι ένα πλήθος ημιαγωγικών διατάξεων, που χρησιμοποιούνται για την λήψη φωτεινών σημάτων, χωρίζονται σε pixels και χρησιμοποιούνται πιο συχνά από τα CCD, διότι καταναλώνουν λίγη ενέργεια και έχουν χαμηλό κόστος.

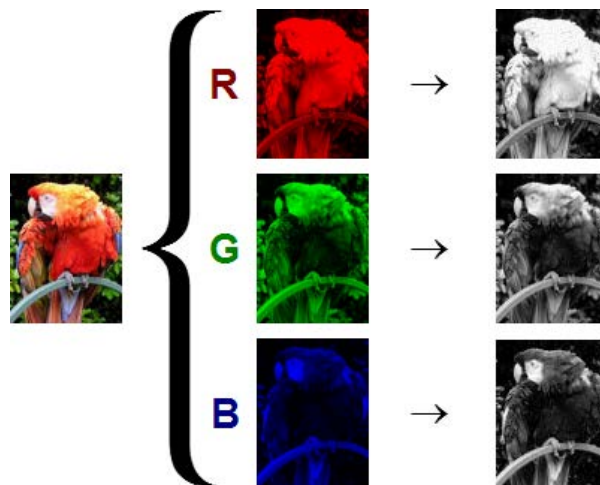




Εικόνα 1.3 Αισθητήρας CMOS

### 1.3 Χρωματικά μοντέλα

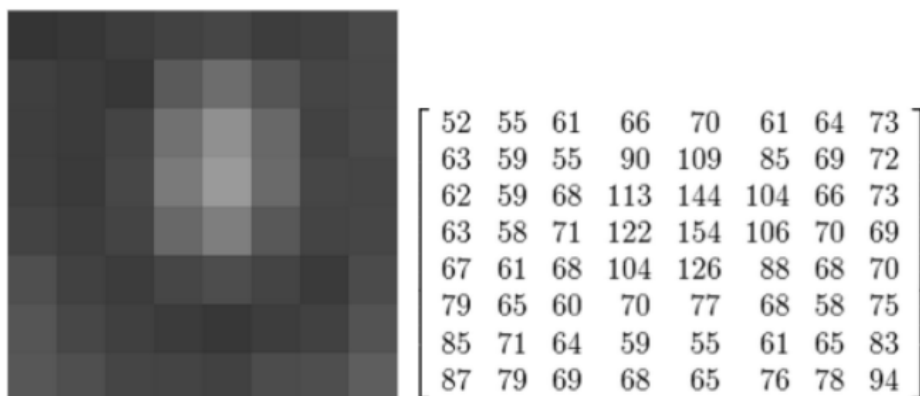
Μια εικόνα αποχρώσεων του γκρι αποτελεί στην γλώσσα του υπολογιστή έναν πίνακα τιμών από το 0 έως το 255. Μια έγχρωμη εικόνα αποτελεί έναν τριπλό πίνακα τιμών στον χρωματικό χώρο RGB. Έχουν προταθεί πολλοί χρωματικοί χώροι για την αναπαράσταση των χρωμάτων στην ψηφιακή εικόνα, όπως τα αντίθετα χρώματα του RGB το CMY και μοντέλα όπως το HSV, YIQ, κτλ. Ο χρωματικός χώρος RGB υπερισχύει στην ψηφιακή επεξεργασία. Το στοιχειώδες τμήμα του πίνακα της εικόνας ονομάζεται εικονοστοιχείο (Pixel) και σε μια έγχρωμη εικόνα φαίνονται οι μείξεις από τους συνδυασμούς τιμών του RGB. Μπορεί να αναφερθεί μεταφορικά σαν μια παλέτα 3 χρωμάτων, κόκκινου, πράσινου και μπλε, όπου οι ποσότητες που αναμιγνύονται, που στην ψηφιακή απεικόνιση λέγονται τιμές φωτεινότητας, δίνουν τις τελικές αποχρώσεις.



Εικόνα 1.1 Το RGB χρωματικό μοντέλο και η grayscale μορφή

## 1.4 Εικόνα αποχρώσεων του γκρι (Grayscale)

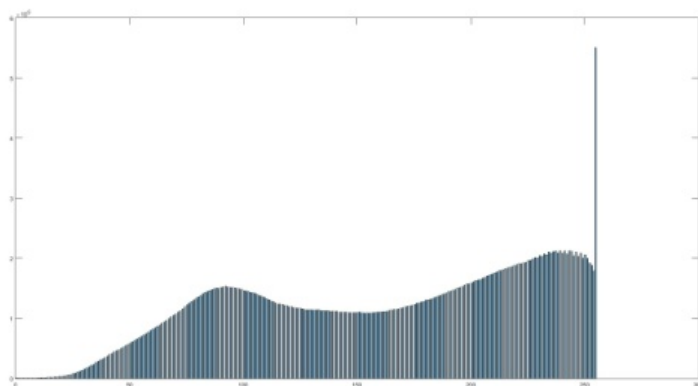
Σε πολλές περιπτώσεις, για την απλότητα των πράξεων σε προγράμματα λογισμικού, η εικόνα μετατρέπεται σε grayscale, δηλαδή μια ενσωμάτωση των τριών πινάκων χρωμάτων σε έναν πίνακα, όπου η τελική φωτεινότητα προκύπτει από τους συνδυασμούς των τριών φωτεινότητων. Η πληροφορία που είναι αναγκαία είναι καθαρά οι φωτεινότητες της εικόνας. Τα αντικείμενα εμφανίζονται ως σύνολα εικονοστοιχείων με κάποιες αντιθέσεις φωτεινότητων. Το χρώμα στη συγκεκριμένη τεχνική είναι μια περιττή πληροφορία. Έτσι, υπάρχει ένας δισδιάστατος πίνακας όπου οι διαστάσεις της εικόνας αναλογούν στο μέγεθος των γραμμών και των στηλών. Μπορεί να φανεί η εικόνα αποχρώσεων μαύρου-γκρι-άσπρου σαν πίνακας τιμών που το κάθε στοιχείο αντιστοιχεί στην τιμή της απόχρωσης. Όταν υπάρχει μια εικόνα με ανάλυση 320x240 pixels, αυτό σημαίνει ότι ο πίνακας της εικόνας είναι με 320 γραμμές και 240 στήλες. Όταν σε μια περιοχή της εικόνας υπάρχουν πολύ φωτεινές περιοχές, αυτό σημαίνει ότι σε εκείνη τη γραμμή και στήλη οι τιμές του αντίστοιχου πίνακα είναι περίπου 200-255. Σε έναν πίνακα grayscale εικόνας είναι εφικτό να γίνουν πολλές αλλαγές, όπως: μπορεί να πειραχτεί η αντίθεση της εικόνας, να γίνει εξισορρόπηση ιστογράμματος, εφαρμογή κάποιας διορθωτικής μάσκας, κάποιου φίλτρου μείωσης του θορύβου, και πολλά άλλα. Έτσι, ανάμεσα στα άλλα, υπάρχει και η δυνατότητα, μέσω μαθηματικών πράξεων και επεξεργασίας, να γίνει και μια αναγνώριση αντικειμένου.



Εικόνα 1.2 Αριστερά: ένα τμήμα εικόνας απόχρωσης γκρι, δεξιά η ψηφιακή μορφή του

## 1.5 Ιστόγραμμα

Μια βασική έννοια είναι και το ιστόγραμμα της εικόνας. Είναι ένα πολύ βοηθητικό εργαλείο για την πληροφορία που μεταφέρει η εικόνα, ένας χάρτης που δείχνει το που κυμαίνονται στατιστικά οι τιμές φωτεινότητων, δηλαδή το πλήθος των εικονοστοιχείων που αντιστοιχούν σε κάθε συγκεκριμένη τιμή φωτεινότητας. Μια εξισορροπημένη ως προς το ιστόγραμμα εικόνα δίνει το αποτέλεσμα μιας εξισορροπημένης κατανομής φωτεινότητων σε όλο το εύρος των δυνατών τιμών.



**Εικόνα 2.3 Μορφή ιστογράμματος φωτεινότητων της εικόνας**

Στην εργασία αυτή, το ιστόγραμμα χρησιμοποιείται με εναλλακτικό τρόπο, καθώς εξάγεται από τους προσανατολισμούς των μεταβολών των φωτεινότητων, δηλαδή των κλίσεων (gradients) που εξάγονται σε κάθε σημείο της εικόνας. Οι κλίσεις αυτές και οι προσανατολισμοί χρησιμοποιούνται για την εξαγωγή των περιγραφέων. Με βάση τα ιστογράμματα προσανατολισμού (orientation histograms) έχουμε τη δυνατότητα να κάνουμε τις ταυτοποιήσεις ανάμεσα σε χαρακτηριστικά σημεία διαφορετικών εικόνων.

## 1.6 Προβλήματα ψηφιακής εικόνας

### 1.6.1 Θόρυβος

Ο θόρυβος είναι ένα κοινό πρόβλημα της ψηφιακής εικόνας και προέρχεται από εγγενείς παράγοντες, όπως ο αισθητήρας της ψηφιακής μηχανής, η ευαισθησία του αισθητήρα (ISO), η λανθασμένη λήψη αποχρώσεων pixel και η μείωση της ποιότητας της εικόνας. Στην πλειοψηφία των εικόνων με αυτό το πρόβλημα του θορύβου, ανήκουν οι φωτογραφίες νυχτερινής λήψης, όταν δηλαδή δεν υπάρχει αρκετός φωτισμός για να ληφθεί η κατάλληλη πληροφορία. Σε άλλες περιπτώσεις, ο θόρυβος μπορεί να εισαχθεί σκόπιμα, από ψηφιακό πρόγραμμα επεξεργασίας εικόνας. Μπορεί να προστεθεί επίτηδες ή και άθελα με μια μείωση της ποιότητας της εικόνας. Όταν υπάρχει αλλοίωση της πληροφορίας σε μια εικόνα, η οποιαδήποτε επεξεργασία επηρεάζεται σημαντικά από τις λανθασμένες τιμές των pixel.



Εικόνα 2.4.1 Φωτογραφία με θόρυβο

## 2. Ψηφιακή επεξεργασία εικόνας

Στην ψηφιακή επεξεργασία εικόνας είναι εφικτό να χρησιμοποιηθούν μαθηματικές πράξεις, με σκοπό τη βελτίωση της εικόνας. Επίσης, μπορεί να γίνει επεξεργασία με σκοπό την εξαγωγή χαρακτηριστικών σημείων ή παραμέτρων. Ως είσοδος, μπορεί να είναι μια φωτογραφία ή μια ακολουθία βίντεο. Οι μέθοδοι επεξεργασίας εικόνας λαμβάνουν την εικόνα σαν ένα δισδιάστατο σήμα, δηλαδή ως ένα δισδιάστατο πίνακα τιμών.

Ότι ισχύει σε μια εικόνα, ισχύει και σε βίντεο, όπου έχουμε έναν αριθμό εικόνων τοποθετημένων πάνω σε χρονική κλίμακα, σε ψηφιακή μορφή. Όταν ο ρυθμός πλαισίων σε ένα βίντεο είναι 25 frames per second, αυτό σημαίνει ότι μέσα σε ένα δευτερόλεπτο έχουμε μια ακολουθία 25 εικόνων. Η επεξεργασία του βίντεο είναι ίδια με αυτή της εικόνας, απλά η διαφορά είναι ότι στο βίντεο γίνεται με απαίτηση την επεξεργασία πολλών εικόνων ανά δευτερόλεπτο.

### 2.1. Βασικές λειτουργίες στην ψηφιακή επεξεργασία εικόνας

Όλες οι βασικές λειτουργίες σε προγράμματα επεξεργασίας εικόνας είναι κυρίως μαθηματικές εφαρμογές πάνω στην ψηφιακή μορφή της εικόνας. Ένα πρόγραμμα ψηφιακής επεξεργασίας μπορεί να δώσει με ιδιαίτερη ευκολία τη δυνατότητα επεξεργασίας στην αντίθεση, στη φωτεινότητα, στα χρώματα, στην εξισορρόπηση του ιστογράμματος, στη θολότητα και σε πολλές άλλες μετατροπές της εικόνας. Αυτά τα εργαλεία, εκτός του ότι πετυχαίνουν τη βελτίωση της αισθητικής της εικόνας, είναι και μια καλή βάση για να χτιστούν αλγόριθμοι για την αναγνώριση αντικειμένων. Οι βασικές λειτουργίες είναι πράξεις στο κάθε pixel της εικόνας, όπου κάθε pixel αντικαθίσταται με μια καινούρια τιμή, έπειτα από την επεξεργασία.

## 2.2 Αντίθεση

Στην ψηφιακή επεξεργασία της εικόνας, η αντίθεση είναι ένα δυνατό εργαλείο στην οπτική παρουσίαση της εικόνας και βοηθάει στην εφαρμογή αλγορίθμων αναγνώρισης αντικειμένων. Η έννοια της αντίθεσης στην ψηφιακή επεξεργασία λαμβάνει τη μορφή φίλτρου που εφαρμόζεται πάνω στο ψηφιακό σήμα της φωτογραφίας. Προγραμματιστικά, είναι απλά μια συνθήκη επανάληψης που ελέγχει την κάθε τιμή του εικονοστοιχείου, κρίνει αν πρέπει να γίνει πιο σκούρο ή πιο ανοιχτό και προσθέτει ή αφαιρεί κάθε φορά μια τιμή, μέσα σε ένα όριο. Επειδή μια εικόνα είναι ένας πίνακας με τιμές απόχρωσης από 0 έως 255, ο αλγόριθμος βελτίωσης της αντίθεσης παίρνει την κάθε τιμή, την συγκρίνει με κάποιο κατώφλι και ανάλογα, την ενισχύει, ή την αποδυναμώνει.

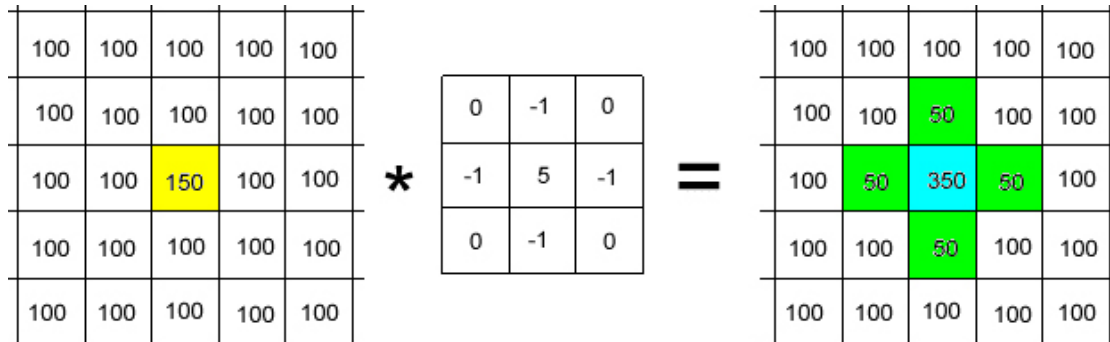


Εικόνα 2.1 αριστερά η αρχική φωτογραφία, δεξιά με εφαρμογή αύξησης του contrast

## 2.3 Συνέλιξη

Στην κατηγορία των ψηφιακών σημάτων βασική εφαρμογή αποτελεί η συνέλιξη μεταξύ δύο σημάτων. Για ένα δισδιάστατο πίνακα όπως είναι οι τιμές φωτεινότητας μιας εικόνας, η συνέλιξη θα κάνει χρήση ενός πίνακα συντελεστών σε μορφή μάσκας ή παραθύρου. Τα παράθυρα αυτά είναι συνήθως τετραγωνικά με συμμετρικούς συντελεστές. Υπάρχουν κατηγορίες βαθυπερατών φίλτρων που φιλτράρουν τις

υψηλές συχνότητες (σήματα θορύβου) και λειαινουν απότομες μεταβολές στην ένταση, όπως φίλτρα της μεσαίας τιμής, butterworth, ενδιάμεσης τιμής και τα Gaussian.



**Εικόνα 2.2.** Αναπαράσταση της πράξης της συνέλιξης σε εικόνα. Στα έγχρωμα pixels (δεξιά) έχει εφαρμοστεί η συνέλιξη με τη μάσκα των συντελεστών.

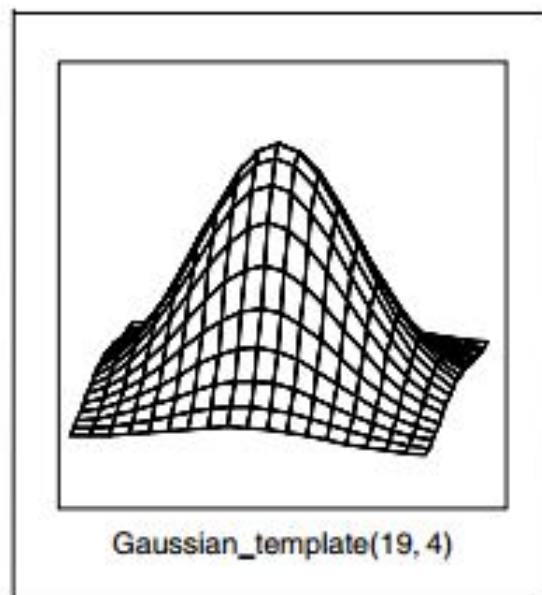
## 2.4 Φίλτρο Gauss

Στα ηλεκτρονικά και στην επεξεργασία σήματος, ένα Φίλτρο Gauss είναι ένα φίλτρο στο οποίο η κρουστική απόκριση είναι μια συνάρτηση Gauss. Η παράμετρος  $\sigma$  (τυπική απόκλιση) της συνάρτησης αυτής είναι υπεύθυνη για το επίπεδο του φίλτρου. Ο συντελεστής Gauss έχει θεωρηθεί ότι είναι το πιο ιδανικό εργαλείο για την εξομάλυνση της εικόνας. Η εφαρμογή του φίλτρου στην εικόνα μας δίνει ένα αποτέλεσμα θολότητας, δηλαδή όσο πιο μεγάλη είναι η παράμετρος  $\sigma$ , τόσο πιο έντονη θα είναι η θολότητα σε μια εικόνα. Στη μελέτη αυτή, το φίλτρο λειτουργεί με διάφορες τιμές  $\sigma$ , για τη δημιουργία της γκαουσιανής πυραμίδας. Το φίλτρο υλοποιείται με τη συνέλιξη με μια τετραγωνική μάσκα συντελεστών. Όσο μεγαλύτερο είναι το παράθυρο, τόσο πιο θολό είναι το αποτέλεσμα της εφαρμογής του φίλτρου στη φωτογραφία.

Η δισδιάστατη απεικόνιση της συνάρτησης Gauss μοιάζει πολύ με καμπάνα. Οι τιμές της συνάρτησης σε διακριτά σημεία είναι οι τιμές του Gauss. Ο πίνακας τιμών μιας μάσκας Gauss είναι ο υπολογισμός των αθροισμάτων των περιοχών όπου τα κεντρικά σημεία έχουν περισσότερη βαρύτητα από τα περιφερειακά σημεία.

Αναλόγως με το μέγεθος της μάσκας έχουμε και την κατάλληλη διακύμανση στο αποτέλεσμα.

Όταν εφαρμοσθεί συνέλιξη με την εικόνα το αποτέλεσμα πάνω στην εικόνα είναι ανάλογο με το πόσο μεγάλο είναι το  $\sigma$  και τι διαστάσεων είναι το παράθυρο. Όσο μεγαλύτερες αυτές οι παράμετροι, τόσο πιο μεγάλη θα είναι και η θολότητα. Ο λόγος που εφαρμόζεται συνέλιξη Gauss στην εικόνα είναι ότι προσφέρει θετικά στοιχεία, ένα από τα οποία είναι και ότι μειώνεται ο θόρυβος της εικόνας.



Εικόνα 2.4 Το φίλτρο Gauss σε δισδιάστατη απεικόνιση

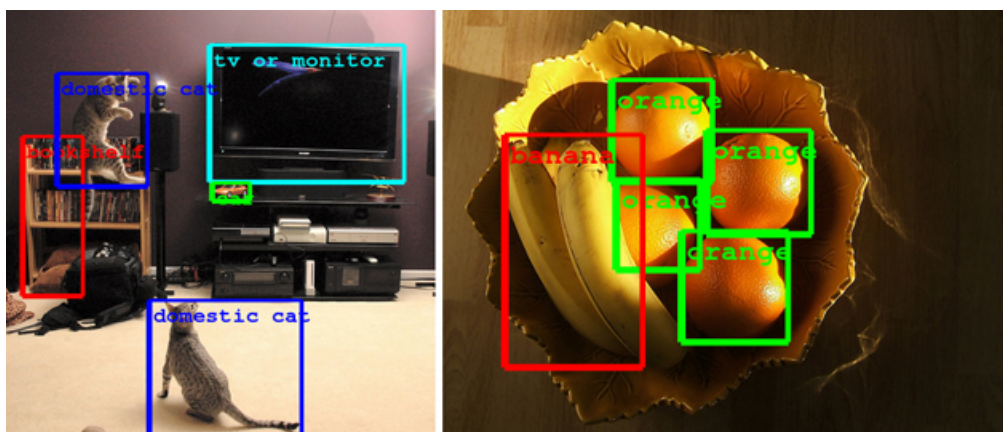


### 3. Εξαγωγή χαρακτηριστικών

#### 3.1 Εισαγωγή

Από το 1980 και μετέπειτα άρχισε να εξελίξη του τομέα της πληροφορικής που λέγεται ρομποτική όραση. Η ρομποτική όραση είναι στενά υνδεμένη με την ψηφιακή επεξεργασία εικόνας και βασίζεται στην αναπαράσταση της φυσικής όρασης. Ο κλάδος της ρομποτικής συνδέθηκε στενά με την ψηφιακή επεξεργασία εικόνας για την επίτευξη της αναγνώρισης σημείων και αντικειμένων στο χώρο. Εκτός από τον τομέα της ρομποτικής, η ρομποτική όραση συνδέεται στενά και με άλλους κλάδους της επιστήμης, λόγω κοινών εφαρμογών. Από το 1990 και μετά, η ρομποτική όραση εξελίχθηκε και με τη βοήθεια της εξέλιξης της τεχνολογίας, άρχισαν να εμφανίζονται οι αλγόριθμοι επεξεργασίας σε πραγματικό χρόνο (real-time).

Οι υπολογιστική όραση δεν έχει νόημα, αν δεν μπορεί ένας υπολογιστής να πάρει μια εικόνα και να κάνει αναγνώριση στο αντικείμενο που είναι αποτυπωμένο σε αυτή. Η αναγνώριση αντικειμένων είναι χρήσιμη στην παραγωγή και στη βιομηχανία. Παραδείγματος χάριν, ένα εργοστάσιο, μέσα σε ένα πολύ σύντομο χρονικό διάστημα πρέπει να παράγει πολλές ποσότητες αντικειμένων. Μια κάμερα αποτυπώνει την εικόνα που παίρνει πληροφορία από τη γραμμή παραγωγής, αναγνωρίζει με τη βοήθεια λογισμικού τον αριθμό κάθε είδους αντικειμένων και δίνει τις κατάλληλες οδηγίες για την εξέλιξη της παραγωγής.



Εικόνα 3.1 Παράδειγμα αναγνώρισης αντικειμένων σε εικόνα

### 3.1.1 Παραδείγματα αναγνώρισης αντικειμένων

Αναγνώριση εικόνας αποτελεί επίσης και η εικόνα ενός κειμένου. Σε κάθε κάτοχο ενός scanner δίνεται ένα αναγνωριστικό πρόγραμμα σε περίπτωση που θέλει να εισάγει, σε ψηφιακή μορφή, αρχεία κειμένου. Από τη στιγμή που εισάγεται η εικόνα σε ψηφιακή μορφή, ένα απλό πρόγραμμα αναγνώρισης κειμένου χρησιμοποιεί αλγόριθμους αναγνώρισης χαρακτήρων. Αφού αναγνωριστούν οι χαρακτήρες γίνεται και η μεταφορά της πληροφορίας σε ένα κειμενογράφο, σε μορφή ASCII.

Η αναγνώριση προσώπου είναι επίσης ένα σημαντικό παράδειγμα για τη χρήση αναγνώρισης εικόνων. Πολλές διαδικτυακές εφαρμογές προσφέρουν διάφορες υπηρεσίες επεξεργασίας, με το ανέβασμα μιας φωτογραφίας. Το πρόσωπο αναγνωρίζεται και αναλόγως με το είδος της προσφερόμενης υπηρεσίας εφαρμόζονται διάφορες τεχνικές. Σε τόπους κοινωνικής δικτύωσης, όταν ο χρήστης ανεβάσει μια φωτογραφία με περιεχόμενο τεσσάρων ατόμων, η εφαρμογή του δίνει τη δυνατότητα να επιλέξει τα πρόσωπα, για να δώσει περισσότερες πληροφορίες γι' αυτά, αν το επιθυμεί. Επίσης, υπάρχουν ιστοσελίδες όπου μπορεί κάποιος να ανεβάσει μια φωτογραφία με ένα πρόσωπο και να βγουν συμπεράσματα, όπως την ηλικία του προσώπου, τα συναισθήματα, αναλογίες προσώπου κλπ.

Για να μπορέσει ένα ρομπότ να κινηθεί σε ένα χώρο άγνωστο για αυτό, χρειάζεται η ρομποτική όραση. Αφού μπορέσει και αναγνωρίσει αντικείμενα, τότε με τις κατάλληλες εντολές που του έχουν δοθεί μπορεί να φερθεί ανάλογα. Υπάρχουν και άλλων ειδών αισθητήρες που είναι αναγκαίοι για την κατανόηση του περιβάλλοντος, όπως σαρωτές laser. Αν και είμαστε στα αρχικά βήματα για την ανάπτυξη των ρομπότ, η όραση μηχανή είναι απαραίτητος τομέας που εξελίσσεται διαρκώς.

Η αναγνώριση αντικειμένων παίζει πολύ σημαντικό ρόλο και στην τεχνητή νοημοσύνη, στη μηχανική μάθηση και στη λήψη αποφάσεων. Η αναγνώριση αντικειμένων, εκτός από την όραση, μπορεί να γίνει συμπληρωματικά και με άλλους αισθητήρες, όπως αφής και ακοής. Ο άνθρωπος λαμβάνει κυρίως οπτικά ερεθίσματα από ένα αντικείμενο, την πληροφορία του χρώματος και την πληροφορία της αποστάσεως.

Εκτός από τα παραπάνω παραδείγματα, υπάρχουν και άλλες εφαρμογές της υπολογιστικής όρασης όπως

- Πλοήγηση (αυτόνομα οχήματα)
- Εύρεση γεγονότων (π.χ. οπτική έρευνα ή μέτρηση ανθρώπων)

- Οργάνωση πληροφοριών (τακτοποίηση βάσεων σχετικών με τις εικόνες)
- Μοντελοποίηση αντικειμένων ή περιβαλλόντων (ιατρική εικονική ανάλυση ή τοπογραφικά μοντέλα)
- Αυτόματη επιτήρηση, Έλεγχος εργασιών κτλ.

## 3.2 Ανιχνευτές Χαρακτηριστικών

### 3.2.1 Χαρακτηριστικά (features)

Χαρακτηριστικά (features) ονομάζονται τα μεγέθη που προέρχονται από την αρχική εικόνα και έχουν υποστεί επεξεργασία, έτσι ώστε να δώσουν άλλες ποσότητες που μπορούν πλέον να βοηθήσουν στον εντοπισμό ενός αντικειμένου. Μια εικόνα δεν είναι άλλο από κάποια νούμερα. Πρώτο μέρος ενός αλγόριθμου είναι συνήθως η προ-επεξεργασία της εικόνας, η βελτίωση και το φιλτράρισμα. Στη συνέχεια, λαμβάνοντας πληροφορία για κάθε εικονοστοιχείο από τη γειτονιά του, μπορούμε να έχουμε περισσότερη πληροφορία. Ένα εικονοστοιχείο μόνο του δεν πρόκειται να γίνει χαρακτηριστικό μιας γωνίας, αν δεν υπάρχει κάτι γύρω του. Μετά την προ-επεξεργασία υπάρχει η εξαγωγή των χαρακτηριστικών. Τα χαρακτηριστικά αυτά μπορούν να είναι ότι τύπου χρειάζεται, αναλόγως με τη λειτουργία του ανιχνευτή: μπορούν να είναι αρνητικά νούμερα, λογικές μεταβλητές, ακέραιοι και ότι άλλο είναι εφικτό ώστε να το επεξεργαστεί ένα μαθηματικό πρόγραμμα. Τα χαρακτηριστικά χωρίζονται σε διάφορες κατηγορίες και υποκατηγορίες όπως χαρακτηριστικά χώρου, χαρακτηριστικά από μετασχηματισμό, γεωμετρικά, στατιστικά και υφής. Πρέπει να πληρούν κάποιες προϋποθέσεις για την καλή λειτουργία του ανιχνευτή. Αυτές οι προϋποθέσεις είναι η διακριτικότητα, η αξιοπιστία, το μικρό υπολογιστικό κόστος και η καλή προσαρμογή στην διαδικασία ταξινόμησης.

### 3.2.2 Ανίχνευση Χαρακτηριστικών

Η ανίχνευση χαρακτηριστικών γνωρισμάτων και η αντιστοίχιση (matching) είναι ένα ουσιαστικό συστατικό πολλών εφαρμογών όρασης των υπολογιστών. Χαρακτηριστικά εικόνας μπορεί να είναι ακμές, γωνίες, μικροί κύκλοι, γραμμές,

γεωμετρικά σχήματα κλπ. Τα χαρακτηριστικά, αφού αναγνωριστούν, μπορούν να χρησιμοποιηθούν για την καθοδήγηση αλγορίθμων, όπως η πλοήγηση, η συνένωση πανοραμικών εικόνων (stitching), η αντιστοίχιση ανάμεσα σε στερεοσκοπικά ζεύγη εικόνων, η αναγνώριση αντικειμένων κλπ. (Brown and Lowe 2007), (Schaffalitzky and Zisserman 2002), (Fergus, Perona, και Zisserman 2007).

Υπάρχουν δύο είδη χαρακτηριστικών α. Τα τοπικά χαρακτηριστικά (local features) και τα γενικά χαρακτηριστικά (global features). Τα πρώτα έχουν σημειακή υφή, τα δεύτερα έχουν πιο καθολική έκταση πάνω στην εικόνα. Στην εργασία αυτή, περιοριζόμαστε σε τοπικά χαρακτηριστικά. Η βασική προσέγγιση για την εξεύρεση αντιστοιχιών ανάμεσα σε χαρακτηριστικά σημεία που ανιχνεύτηκαν σε διαφορετικές εικόνες είναι η συσχέτιση περιοχών (correlation) ή η εύρεση των ελαχίστων απόλυτων διαφορών (SAD).

### **3.2.3 Αναλλοίωτα (invariant) χαρακτηριστικά**

Οι ανιχνευτές αναλλοίωτων χαρακτηριστικών έχουν ως ιδιότητα να ανιχνεύουν χαρακτηριστικά, που δεν μεταβάλλονται όταν η εικόνα υφίσταται διάφορους μετασχηματισμούς. Αμετάβλητα χαρακτηριστικά ονομάζονται όταν τα χαρακτηριστικά σημεία μιας εικόνας παραμένουν σταθερά, αμετάβλητα σε περιστροφή, μεταφορά, αλλαγή κλίμακας, σε αφινικό μετασχηματισμό και σε μικρές αλλαγές της φωτεινότητας. Η ανίχνευση αμετάβλητων χαρακτηριστικών μπορεί να γίνει με τη δημιουργία μιας κλιμάκωσης της εικόνας, έτσι ώστε να εξάγονται διακριτά χαρακτηριστικά σε διάφορες κλίμακες των αντικειμένων που θέλουμε να ταυτοποιήσουμε.

### **3.3 Η έννοια της χωροκλίμακας (Scale Space)**

Ο Witkin (1983) έκανε την πρώτη αναφορά στην χώρο-κλίμακα. Θεώρησε ότι η κλίμακα μιας εικόνας είναι μια συνεχής παράμετρος και έδωσε τις βάσεις για μελέτη και εξέλιξη των αλγορίθμων αναγνώρισης αντικειμένων πάνω στην εικόνα.

Η θεωρία της χωροκλίμακας είναι ένα πλαίσιο για πολλαπλή αναπαράσταση του σήματος για την υπολογιστική όραση, με κίνητρο την φυσική και βιολογική όραση.

Από μια αρχική εικόνα δίνονται οι τιμές φωτεινότητας και με συνέλιξη φίλτρου δημιουργείται η πρώτη εικόνα της κλίμακας.

Η χωροκλίμακα βασίζεται στη συνάρτηση τριων μεταβλητών. Οι δύο μεταβλητές είναι οι μεταβλητές της αρχική εικόνας και τρίτη είναι η παράμετρος της κλίμακας. Η παράγωγος της συνάρτησης των τριων αυτών μεταβλητών πρέπει να εξαρτάται μόνο από την κλίμακα. Όταν ανιχνευτούν τα στοιχεία της εικόνας πρέπει να ισχύει ότι η κάθε παράγωγος ως προς την κάθε μεταβλητή είναι ίση με 0. Μέσα από υπολογισμούς για το τι πρέπει να ισχύει στο μαθηματικό επίπεδο κατέληξαν σε σημαντικά συμπεράσματα οι ερευνητές Koendrick (1984) και Linderberg (1994). Το πιο αποδεκτό φίλτρο για η συνέλιξη της εικόνας στην έννοια της χωροκλίμακα είναι η συνάρτηση Gauss.

### 3.4.1 Η δημιουργία της χωροκλίμακας με φίλτρο Gauss

Έστω η παρακάτω συνάρτηση L, όπου x,y οι συντεταγμένες της εικόνας I και όπου σ η παράμετρος κλίμακας

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

Όπου η L παράγεται από τη συνέλιξη της συνάρτησης Gauss με την αρχική εικόνα.

Όπου G η συνάρτηση του Gauss:

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



Εικόνα 3.4.1 Αριστερά η αρχική εικόνα και δεξιά φιλτραρισμένη με Gauss

Ξεκινώντας από την αρχική εικόνα, εφαρμόζονται μάσκες φίλτρου Gauss με διαφορετικές παραμέτρους κάθε φορά, με αυξανόμενη την τυπική αποκλιση  $\sigma$ . Ο λόγος που συμβαίνει αυτή η συνέλιξη της εικόνας είναι διότι μπορεί να προσφέρει αναλλοίωτα τα χαρακτηριστικά της εικόνας. Τα χαρακτηριστικά αυτά λαμβάνονται έπειτα από την επεξεργασία αυτή σε μία ή περισσότερες οκτάβες. Στα όρια κάθε οκτάβας, η τιμή της τυπικής απόκλισης  $\sigma$  διπλασιάζεται.

### 3.4.2 Πυραμίδα Gauss

Η πυραμίδα Gauss είναι η εφαρμογή της συνάρτησης Gauss για τη δημιουργία της χωροκλίμακας. Στην εικόνα γίνεται η εφαρμογή των φίλτρων και σε κάθε νέα οκτάβα γίνεται υποδειματοληψία της εικόνας. Συνήθως, ο παράγοντας της υποδειματοληψίας είναι το 2. Αν υπάρχουν τρία επίπεδα μιας πυραμίδας Gauss σε μία εικόνα, δηλαδή τρεις οκτάβες, τότε το πρώτο επίπεδο θα είναι η αρχική εικόνα με την εφαρμογή της συνάρτησης. Το δεύτερο επίπεδο θα είναι η μισή ανάλυση της εικόνας πάλι με την εφαρμογή της συνάρτησης και το τρίτο επίπεδο επίσης θα έχει μειωθεί σε σχέση με την δεύτερη κατά το μισό, δηλαδή το  $\frac{1}{4}$  από τη αρχική εικόνα με την εφαρμογή της συνάρτησης.

### 3.5 Σχετική έρευνα

Η ανάπτυξη των αντιστοιχίσεων χαρακτηριστικών της εικόνας, με βάση το σύνολο των τοπικών σημείων ξεκινάει από την δουλεία του Moravec (1981) πάνω στην τεχνική της στερεοσκοπικής αντιστοίχισης, χρησιμοποιώντας έναν ανιχνευτή γωνιών. Ο ανιχνευτής του Moravec βελτιώθηκε μετέπειτα από τον Harris και τον Stephens (1988) οι οποίοι εισηγήθηκαν μια τεχνική για την εύρεση γωνιών και ακμών. Ο ανιχνευτής του Harris επίσης φάνηκε πιο αποτελεσματικός στον εντοπισμό κίνησης και στην εύρεση της τρισδιάστατης δομής (Harris, 1992). Από τότε, ο ανιχνευτής Harris έχει διαδοθεί σημαντικά για κάποιες εφαρμογές. Αν και οι συγκεκριμένοι ανιχνευτές ονομάζονται ανιχνευτές γωνιών, δεν λειτουργούν μόνο πάνω στις γωνίες

αλλά και σε άλλα γνωρίσματα της εικόνας η οποία έχει πολλές κλίσεις σε όλες τις κατευθύνσεις, σε μια προκαθορισμένη κλίμακα.

Οι αρχικές εφαρμογές ήταν για στερεοσκοπία και για εντοπισμό κίνησης μικρού εύρους σε διαχικές εικόνες, αλλά ο σκοπός των αλγορίθμων εξελίχθηκε αργότερα πάνω σε πιο σύνθετα προβλήματα. Ο Zhan et al (1995) έδειξε ότι είναι δυνατό να γίνει αντιστοίχιση γωνιών Harris πάνω σε ένα μεγαλύτερο εύρος σε διαδοχικές εικόνες, χρησιμοποιώντας παράθυρα συσχετίσεων γύρω από την κάθε γωνία για την καλύτερη επιλογή αντιστοιχήσεων (matching). Στη συνέχεια, απομακρύνθηκαν από την επίλυση του θεμελιώδη πίνακα οι ακραίες τιμές και αφαιρέθηκαν αντιστοιχήσεις που δεν συμφωνούσαν με την πλειοψηφία. Περίπου στο ίδιο χρονικό διάστημα, παρόμοια έρευνα είχε αναπτυχθεί από τον Torr (1995) για την αντιστοίχιση (matching) σε κίνηση μεγάλου εύρους, όπου χρησιμοποιήθηκε γεωμετρικός προσδιορισμός για να αφαιρεθούν οι ακραίες τιμές για άκαμπτα κινούμενα αντικείμενα μέσα σε μια εικόνα.

Η πρωτοποριακή δουλειά του Schmid και Mohr (1997) έδειξε ότι το αμετάβλητο τοπικό χαρακτηριστικό ταίριασμα μπορεί να επεκταθεί σε γενικά προβλήματα αναγνώρισης εικόνας, όταν το χαρακτηριστικό συγκρίνεται με μια μεγάλη βάση εικόνων. Επίσης έγινε η χρήση του ανιχνευτή γωνιών Harris, για να γίνει η επιλογή των σημείων ενδιαφέροντος, αλλά αντί να γίνει το ταίριασμα με τα παράθυρα συσχέτισης, χρησιμοποίησαν περιγραφέα αμετάβλητο σε περιστροφές, στην τοπική περιοχή της εικόνας. Αυτό επέτρεψε το ταίριασμα των χαρακτηριστικών κάτω από αυθαίρετο διαφορετικό προσανατολισμό μεταξύ των 2 εικόνων.

Ο αλγόριθμος Harris είναι πολύ ευάλωτος στις διαφορετικές κλίμακες εικόνας, Έτσι δεν προσφέρει αξιοπιστία κάτω από αυτές τις συνθήκες. Η δουλειά του Lowe (1999) επεκτάθηκε στην ανίχνευση τοπικών χαρακτηριστικών που παραμένουν αναλλοίωτα στις διαφορετικές κλίμακες. Αυτή η δουλειά είχε ως αποτέλεσμα τη δημιουργία του τοπικού περιγραφέα, ο οποίος τελικά παρέχει περισσότερα διακριτικά χαρακτηριστικά, ενώ είναι λιγότερο ευάλωτος στις παραμορφώσεις της εικόνας.

Αρκετές έρευνες έχουν βασιστεί στο χώρο της κλιμάκωσης. Κάποιες από τις πρώτες δουλειές είναι του Crowley & Parker (1984), που ανέπτυξαν μια αναπαράσταση που βρίσκει κορυφές σε κλίμακα και συνδέεται με την δομή δέντρου. Η δομή δέντρου θα μπορούσε να ταιριάζει ανάμεσα στις εικόνες με αυθαίρετη αλλαγή κλιμάκωσης. Πιο πρόσφατη εξέλιξη μετά από αυτό ήταν το ταίριασμα γραφήματος από τον Shokoufandeh Marsic και Dickinson (1999). Αυτοί παρείχαν περιγραφείς πιο

διακριτών χαρακτηριστικών χρησιμοποιώντας κυματικούς συντελεστές. Το πρόβλημα της ταυτοποίησης του κατάλληλου και συνεπούς τρόπου κλιμάκωσης για εύρεση χαρακτηριστικών μελετήθηκε σε βάθος από τον Linderberg (1993).

Πιο πρόσφατα (2000-2002) υπήρξε ένα εύρος εργασιών για την επέκταση των τοπικών χαρακτηριστικών με σκοπό να μένουν αναλλοίωτα σε πλήρεις μετασχηματισμούς από τους Baumberg, Tuytelaars & Van Gool, Mikolajczyk & Schmid, Schaffalitzky & Zisserman, Brown & Lowe). Καμία από όλες αυτές τις προσεγγίσεις δεν έχει φτάσει στο επίπεδο της πληρότητας.

Πολλοί αλγόριθμοι και μελέτες έχουν γίνει για την αναγνώριση αντικειμένων με εξαγωγή χαρακτηριστικών κάτω από διάφορες συνθήκες. Πάνω σε αυτές τις έρευνες η πιο βασική με τα περισσότερα θετικά αποτελέσματα είναι ο SIFT. Ο συγκεκριμένος αλγόριθμος είναι επιτυχής στην εξαγωγή αμετάβλητων χαρακτηριστικών.

### **3.6 Αλγόριθμοι εξαγωγής χαρακτηριστικών**

#### **RIFT (rotation-invariant Feature Transform)**

Ο αλγόριθμος RIFT είναι μια παραλλαγή του αλγόριθμου που εξετάζουμε, του SIFT. Είναι μια επέκταση με τη διαφορά ότι οι περιοχές εικόνας (patches) που εξάγονται είναι αναλλοίωτοι σε περιστροφές. Γίνεται διαίρεση της εικόνας σε ομόκεντρους δακτυλίους ίσου πλάτους υπολογίζοντας για το κάθε δακτύλιο ένα ιστόγραμμα προσανατολισμών.

#### **SURF (Speeded Up Robust Features)**

Η υλοποίηση του μοιάζει αρκετά με το SIFT. Είναι ένας ανιχνευτής που προσδιορίζει τα σημεία ενδιαφέροντος σε μια εικόνα. Γίνεται η εύρεση των σημείων ενδιαφέροντος, τα οποία παραμένουν αμετάβλητα σε διάφορες μεταβολές κλίμακας, και σε περιστροφές. Η κάθε γειτονιά του σημείου εμπεριέχει το διάνυσμα του περιγραφέα, έτσι ώστε να γίνει η αντιστοίχιση των χαρακτηριστικών. Με βάση την εξαγωγή διανυσμάτων των περιγραφέων, στο τέλος γίνεται η σύγκριση τους και προκύπτει αν αυτά είναι ίδια σε κάθε εικόνα. Γίνεται χρήση του γρήγορου εσσιανού πίνακα και εξαγωγή των σημείων από το χώρο της κλιμάκωσης. Αφού βρεθεί ο περιγραφέας και ο προσανατολισμός του αντικειμένου, παράγεται ένα διάνυσμα ενός περιγραφέα 64 στοιχείων.



### **SPIN (Sorting points into neighborhoods)**

Στον αλγόριθμο SPIN σημαντικό ρόλο παίζει η φωτεινότητα της εικόνας. Εξάγει ένα διδιάστατο ιστόγραμμα με βάση τα στοιχεία φωτεινότητας και της απόστασης από το κέντρο.

### **Harris**

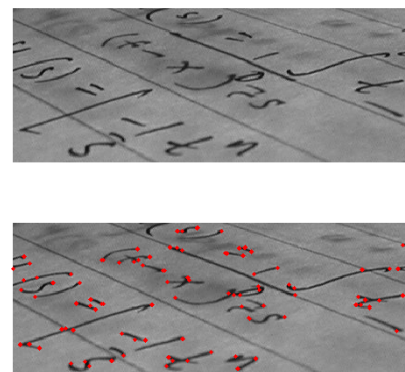
Βελτιωμένη έκδοση του Moravec (1971) είναι ο ανιχνευτής γωνιών του Harris and Stephens (1991), ο οποίος είναι γνωστός και σαν ανιχνευτής Plessey. Είναι αλγόριθμος βασισμένος στον πίνακα δεύτερης ροπής (second moment matrix) και ο πρώτος υπολογισμός που γίνεται είναι ο πίνακας Harris. Είναι γρήγορος, υπολογίζει ακμές, και γωνίες, αλλά δεν υπολογίζει προσανατολισμό και κλίμακα.

### **SUSAN (Smallest Univalued Segment Assimilating Nucleus)**

Ο Ανιχνευτής Ελάχιστων Τμημάτων Ενιαίας Τιμής Προσομοίωσης Πυρήνα είναι δημιουργία του Smith και Brady (1997) και βασίζεται στη σύγκριση εικονοστοιχείων με βάση τη φωτεινότητα. Έχει πολύ καλή ακρίβεια στα γωνιακά σημεία και η βασική του θεωρία είναι ότι όταν σε μια μικρή περιοχή εικονοστοιχείων υπάρχει ομοιομορφία φωτεινότητας υπάρχει η πιθανότητα να ανήκει σε ίδιο αντικείμενο. Βρίσκει γωνίες με βάση μια τιμή κατωφλίου.

### **Harris-Laplace**

Αποτελεί βελτιωμένη εκδοχή του Harris με βάση τον αλγόριθμο του πίνακα Harris. Επίσης, βασίζεται στον πίνακα συσχέτισης. Είναι βελτιωμένος ως προς μετασχηματισμούς μεταφοράς, περιστροφής και αλλαγής κλίμακας.



Εικόνα 3.6 Εξαγωγή χαρακτηριστικών από ανιχνευτές γωνιών

## **4. Ανιχνευτής SIFT (Scale Invariant Feature Transform)- Μετασχηματισμός αναλλοίωτων ως προς την κλίμακα χαρακτηριστικών σημείων**

### **4.1 Εισαγωγικά**

Ο ανιχνευτής SIFT είναι ένας αλγόριθμος της ρομποτικής όρασης για εύρεση και περιγραφή τοπικών αμετάβλητων χαρακτηριστικών πάνω στην εικόνα. Ο αλγόριθμος αυτός παρουσιάστηκε από τον David Lowe το 1999. Ο David G. Lowe είναι επιστήμονας του κλάδου της πληροφορικής στον Καναδά και υπεύθυνος του τμήματος πληροφορικής στο πανεπιστήμιο της Κολούμπια. Έχει ερευνήσει την ρομποτική όραση και έχει γράψει το πιο διαδεδομένο αλγόριθμο πάνω στην περιγραφή και εύρεση των χαρακτηριστικών της εικόνας.

Το ταίριασμα χαρακτηριστικών σε μια εικόνα αποτελεί θεμελιώδες ζήτημα πολλών προβλημάτων στην ρομποτική όραση. Αναφέρεται σε προβλήματα αναγνώρισης αντικειμένων, αναπράστασης της δομής τρισδιάστατων κατασκευών από πολλαπλές εικόνες. Επίσης, βρίσκει χρησιμότητα σε προβλήματα στεροσκοπικής αντιστοίχισης και εντοπισμού κίνησης. Τα χαρακτηριστικά SIFT μιας εικόνας έχουν πολλές ιδιότητες από τις οποίες μπορεί να γίνει εντοπισμός διαφόρων τμημάτων εικόνων για την αναγνώριση του περιεχόμενου μιας εικόνας. Τα χαρακτηριστικά SIFT είναι αναλλοίωτα στην κλίμακα της φωτογραφίας και εν μέρει αναλλοίωτα στις συνθήκες φωτισμού. Εντοπίζονται και στο πεδίο της συχνότητας. Μπορούν να εξαχθούν μεγάλοι αριθμοί χαρακτηριστικών SIFT από απλές φωτογραφίες.

Άλλο ένα επίσης σημαντικό χαρακτηριστικό είναι ότι οι σχετικές θέσεις των χαρακτηριστικών SIFT δεν αλλάζουν από τη μια εικόνα στην άλλη. Για παράδειγμα, έστω ότι αναγνωρίζονται τα keypoints σε μια πόρτα που είναι κλειστή. Αν υπάρχει μια εικόνα της πόρτας που είναι λίγο ανοιχτή, τότε με άλλους αλγόριθμους θα ήταν δύσκολο να βρεθεί το ταίριασμα (matching) της κλειστής πόρτας. Στο σημείο αυτό είναι πιο αποτελεσματικός ο SIFT. Μπορεί και εξάγει περισσότερα χαρακτηριστικά, που είναι αμετάβλητα σε αλλαγές της εικόνας, για να βοηθήσουν στην αποφυγή των λαθών στο ταίριασμα των εικόνων.

## 4.2 Ιδιότητες

Ο αλγόριθμος SIFT μπορεί να εντοπίσει δυναμικά τα αντικείμενα, ακόμη και αν υπάρχει ακαταστασία στην εικόνα, διότι η βάση για τη λειτουργία του αλγόριθμου είναι ότι η περιγραφή που αποδίδει είναι αναλλοίωτη σε ομοιόμορφη κλιμάκωση, προσανατολισμό και αμετάβλητη σε παραμόρφωση και αλλαγές φωτισμού.

Τα σημεία-κλειδιά (keypoints) που εξάγονται από τον αλγόριθμο SIFT αποθηκεύονται σε μια βάση δεδομένων. Όταν αυτές οι βάσεις συγκριθούν και βρεθούν τα χαρακτηριστικά για την ομοιότητα δύο εικόνων, τότε χρησιμοποιείται και η ευκλείδεια απόσταση των διανυσματικών χαρακτηριστικών τους. Πέρα από την ευκλείδεια απόσταση, μπορούν επίσης να χρησιμοποιηθούν παρόμοιες αποστάσεις από το γνωστικό επίπεδο της αναγνώρισης προτύπων.

Το κόστος της εξαγωγής των περιγραφέων μπορεί να μειωθεί σημαντικά, εκτελώντας διαδοχικά φίλτρα, στα οποία υπάρχει περισσότερη ακρίβεια και εφαρμόζονται μόνο στα σημεία όπου έχουν περάσει τα αρχικά τεστ. Ο αλγόριθμος ακολουθεί τα εξής στάδια:

### **1) Ανίχνευση ακρότατων στο χώρο της κλιμάκωσης (Scale-space extrema detection)**

Στο πρώτο στάδιο γίνεται αναζήτηση ακρότατων πάνω στις Λαπλασιανές εικόνες. Έχει εφαρμοσθεί αποτελεσματικά χρησιμοποιώντας την διαφορά των συναρτήσεων Gauss (Differences of Gaussians) για να αναγνωρισθούν υποψήφια σημεία-κλειδιά, τα οποία είναι αναλλοίωτα στην κλίμακα και στον προσανατολισμό.

### **2) Εντοπισμός σημείων-κλειδιών (Patch Localization)**

Κάθε υποψήφιο σημείο εντοπίζεται με βάση τη σταθερότητά του και καθορίζονται οι στυντεταγμένες του και η κλίμακα όπου ανιχνεύεται.

### **3) Ανάθεση προσανατολισμού (Orientation Assignment)**

Ένας ή περισσότεροι προσανατολισμοί ανατίθενται στο κάθε σημείο-κλειδί με βάση τις τοπικές κλίσεις της εικόνας.

#### 4) Περιγραφή των σημείων-κλειδιών (Patch Description)

Οι τοπικές κλίσεις της εικόνας μετρούνται στην επιλεγμένη κλίμακα στην περιοχή γύρω από το κάθε σημείο-κλειδί. Αφού έχουν μετασχηματιστεί σε ένα αντιπροσωπευτικό διάνυμα, είναι δυνατό να ταυτοποιηθούν ακόμη και μετά την παραμόρφωση σχημάτων ή αλλαγή φωτεινότητας.

Αυτή η προσέγγιση έχει ονομαστεί Ανίχνευση ακρότατων στο χώρο της κλιμάκωσης (Scale-space extrema detection), εφόσον μετασχηματίζει τα δεδομένα εικόνας σε συντεταγμένες αναλλοίωτης κλίμακας, σχετικές με τα τοπικά χαρακτηριστικά.

Αυτή η προσέγγιση δημιουργεί μεγάλους αριθμούς χαρακτηριστικών τα οποία μπορούν να καλύψουν πυκνά την εικόνα. Μια τυπική εικόνα μεγέθους 500x500 pixels θα δώσει περίπου κατά μέγιστο 2000 σταθερά χαρακτηριστικά (αν και το νούμερο εξαρτάται από το περιεχόμενο της και τις επιλογές των παραμέτρων). Ο αριθμός των χαρακτηριστικών είναι πολύ σημαντικός παράγοντας για την αναγνώριση του αντικειμένου, όπου χρειάζεται τουλάχιστον 3 χαρακτηριστικά για την αναγνώριση ενός μικρού αντικειμένου πάνω σε μια εικόνα.

Για το ταίριασμα και την αναγνώριση αντικειμένου πάνω σε εικόνα, τα χαρακτηριστικά του αλγόριθμου που έχουν εξαχθεί από την εικόνα, αποθηκεύονται σε μια βάση δεδομένων έτσι ώστε αν εισαχθεί μια δεύτερη εικόνα και εξαχθούν τα χαρακτηριστικά της, να μπορεί να συγκριθεί με κάποια που ανήκουν στη βάση. Αν βρεθούν υποψήφια κλειδιά κατά τη σύγκριση, τότε υπολογίζονται με ευκλείδεια απόσταση οι αποστάσεις των σημείων.

Οι περιγραφείς των σημείων-κλειδιών είναι διακριτικοί, κάτι που επιτρέπει σε κάθε χαρακτηριστικό να βρει το σωστό ταίριασμα με καλή πιθανότητα σε μια μεγάλη βάση χαρακτηριστικών. Ωστόσο, σε μια ακατάστατη εικόνα, πολλά χαρακτηριστικά από το βάθος μπορεί να μην έχουν το σωστό ταίριασμα από τη βάση δεδομένων, δίνοντας έτσι αφορμή σε πολλά λάθος ταιριάσματα. Το σωστά ταιριάσματα μπορούν να φιλτραριστούν μέσα από ένα πλήρες σύνολο ταιριασμάτων απλά από την ταυτοποίηση υποσυνόλων των σημείων-κλειδιών το οποίο θα συμφωνεί με τα δεδομένα της θέσης, της κλίμακας και του προσανατολισμού της εικόνας.

#### 4.2.1 Προβλήματα και πλεονεκτήματα του SIFT

Πρόβλημα	Τεχνική	Πλεονέκτημα
Κλειδί εντοπισμού / Κλίμακα / Περιστροφή	DoG / χώρο-κλίμακα πυραμίδα / Τοποθέτηση προσανατολισμού	Ακρίβεια / Σταθερότητα / Αναλλοίωτη κλίμακα & περιστροφή
Γεωμετρική παραμόρφωση	Θολότητα/ δειγματοληψία των προσανατολισμών της εικόνας	Αναλλοίωτο σε αφινικό μετασχηματισμό
Εύρεση και Αντιστοίχιση	Πλησιέστερο γειτονικό στοιχείο/ Πρώτο ταίριασμα	Αποτελεσματικότητα / Ταχύτητα
Μαζική ταυτοποίηση	Μετασχηματισμός Hough	Αξιοπιστία
Επαλήθευση Μοντέλου / Εντοπισμός ακραίων τιμών	Γραμμική ελάχιστων τετραγώνων	Καλύτερη ανοχή σφάλματος με λιγότερα ταυριάσματα
Αποδοχή υπόθεσης	Πιθανότητα Bayesian	Αξιοπιστία

#### Πλεονεκτήματα

**Ακρίβεια:** Ο εντοπισμός των χαρακτηριστικών γίνεται σε ικανοποιητικό βαθμό με ακρίβεια στον χώρο και στην κλίμακα

**Πλήθος:** Για να είναι δυνατή μια καλή ταυτοποίηση πρέπει να υπάρχει ένα σημαντικό πλήθος χαρακτηριστικών.

**Διακριτικότητα:** Η αλλαγή φωτεινότητας στα γειτονικά στοιχεία των χαρακτηριστικών είναι σημαντικός παράγοντας για να ξεχωρίσει ένα χαρακτηριστικό σημείο

**Ταχύτητα:** Ο αλγόριθμος μπορεί να εφαρμοσθεί και σε real-time εφαρμογές.

## 4.3 Στάδια του ανιχνευτή SIFT

### 4.3.1 Εύρεση της χωροκλίμακας

Ο εντοπισμός σημείων ενδιαφέροντος γίνεται χρησιμοποιώντας μια προσεγγιστική αλληλουχία φίλτρων, όπου χρησιμοποιούνται αποτελεσματικοί αλγόριθμοι για την ταυτοποίηση υποψήφιων περιοχών. Το πρώτο στάδιο του εντοπισμού χαρακτηριστικών σημείων είναι να γίνει η ταυτοποίηση θέσεων και κλιμάκων σε επαναλαμβανόμενες εικόνες. Εντοπίζονται τοποθεσίες δηλαδή οι οποίες είναι αναλλοίωτες πάνω στην επεξεργασία της κλιμάκωσης.

Έχει αποδειχθεί από τον Koenderik και Lindeberg με πολλές δοκιμές ότι ο καλύτερος τρόπος για τη δημιουργία της κλίμακας-χώρου είναι η εφαρμογή του τύπου του Gauss.

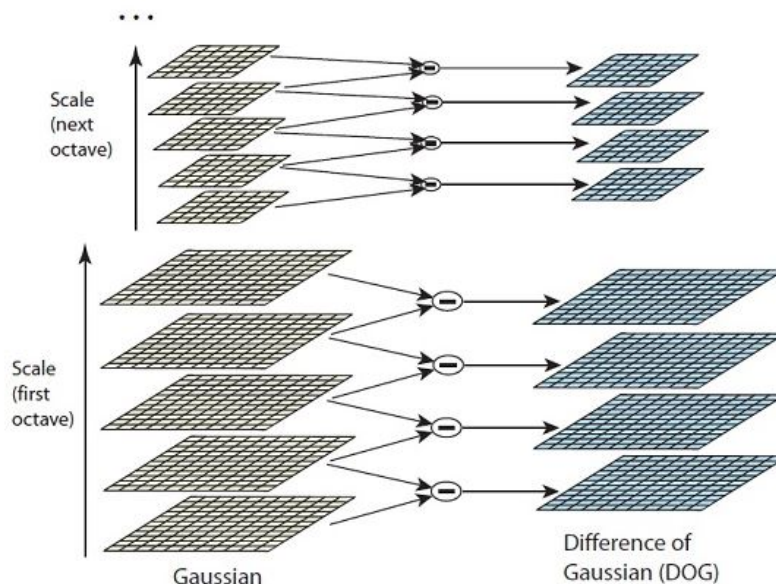
Το πρώτο στάδιο του ανιχνευτή SIFT βασίζεται στη δημιουργία κλιμακούμενων γκαουσιανών. Η οκτάβα είναι ένα σύνολο γκαουσιανών της ίδιας ανάλυσης αλλά με ένα αυξανόμενο σίγμα ( $\sigma$ ) που αυξάνει την θολότητα. Η παράμετρος σίγμα είναι η τυπική απόκλιση στον τύπο του φίλτρου Gauss. Αφού οριστεί ένα φίλτρο τότε εφαρμόζεται συνέλιξη αυτού του φίλτρου με την αρχική εικόνα. Στο συγκεκριμένο μέρος του αλγόριθμου απλά ακολουθείται μια συγκεκριμένη ροή. Ξεκινώντας την πρώτη οκτάβα με την αρχική ανάλυση της εικόνας, εφαρμόζεται συνέλιξη με φίλτρο Gauss με  $\sigma_0 = \sqrt{2}/2$ . Αυτή είναι πλέον η πρώτη γκαουσιανή της πρώτης οκτάβας.

Στη συνέχεια εφαρμόζεται συνέλιξη της νέας εικόνας με  $\sigma_0 = 2\sqrt{2}$ . Δημιουργούνται 5 εικόνες που έχουν υποστεί εφαρμογή φίλτρου με αυξανόμενο σίγμα για την πρώτη οκτάβα. Για τη δημιουργία της δεύτερης οκτάβας εφαρμόζεται υποδειγματολειψία στην αρχική εικόνα με τον παράγοντα 2. Η πρώτη εικόνα αυτή τη φορά όμως θα υποστεί συνέλιξη με αρχικό  $\sigma_0 = 2 * \sqrt{2}/2 = \sqrt{2}$ . Έτσι σε κάθε αλλαγή της επόμενης οκτάβας ξεκινάει με το επόμενο  $\sigma_0$ .

Κλίμακα →					
Οκτάβα ↓	$\sigma_0 = \frac{\sqrt{2}}{2}$	$\sigma_1 = \kappa \cdot \sigma_0$	$\sigma_2 = \kappa^2 \sigma_0$	$\sigma_3 = \kappa^3 \sigma_0$	$\sigma_4 = \kappa^4 \sigma_0$
	$2\sigma_0$	$2\sigma_1$	$2\sigma_2$	$2\sigma_3$	$2\sigma_4$
	$4\sigma_0$	$4\sigma_1$	$4\sigma_2$	$4\sigma_3$	$4\sigma_4$
	$8\sigma_0$	$8\sigma_1$	$8\sigma_2$	$8\sigma_3$	$8\sigma_4$

Πίνακας Οκτάβων. Όπου  $\kappa = \sqrt{2}$

Για κάθε οκτάβα στο χώρο της κλιμάκωσης, στην αρχική εικόνα γίνεται η συνέλιξη με το φίλτρο Gauss για να παράγει το σετ διαφόρων εικόνων με διαφορετικό σίγμα, όπως δείχνει η εικόνα 4.3 αριστερά. Σκοπός της κλιμάκωσης των εικόνων είναι να αφαιρεθούνε μεταξύ τους και να παράγουνε την διάφορα των γκαουσιανών (Difference of Gaussians) στα δεξιά της εικόνας. Μετά από κάθε οκτάβα, για την επόμενη κλίμακα η γκαουσιανή εικόνα υποδειγματίζεται κατά τον παράγοντα 2 και η διαδικασία επαναλαμβάνεται. Σύμφωνα με τον Lowe, για την σωστή και επιτυχή διαδικασία της μεθόδου χρειάζονται τουλάχιστον 4 οκτάβες με την αρχική εικόνα να έχει διπλασιασμένο μέγεθος. Η μέθοδος για το διπλασιασμό εικόνας είναι με διγραμμική παρεμβολή με βάση τον πλησιέστερο γείτονα.



Εικόνα 4.3 : Απεικόνιση των οκτάβων και των DoG

Αφού έχει δημιουργηθεί ο χώρος της κλιμάκωσης, ο αλγόριθμος πλέον είναι σε θέση να εξαγάγει κάποια χαρακτηριστικές θέσεις για τον εντοπισμό των μέγιστων και ελάχιστων ακρότατων. Η αφαίρεση 2 γκαουσιανών με διαφορά κλίμακας μας δίνει την διαφορά των γκαουσιανών ή όπως ονομάζεται αλλιώς σε συντόμευση DoG από την φράση Difference of Gaussians. Από μια οκτάβα μπορούν να εξαχθούν 4 DoG. Αφού έχουν εξαχθεί το αποτέλεσμα οπτικά δεν είναι παρά μόνο κάποιες ακμές όπου φαίνονται κάποια στοιχεία των αντικειμένων που υπάρχουν στην εικόνα.

Αν οι εικόνες πρώτης οκτάβας είναι  $L_{n,1}, n = 1, \dots, N$  τότε εφαρμόζεται η συνέλιξη

$$L_{n,1}(x, y, \sigma(n, 1)) = G(x, y, \sigma(n, 1)) * I(x, y)$$

G είναι το φίλτρο Gauss

I είναι η εικόνα

L η εικόνα της οκτάβας

Τότε για τις DoG θα έχουμε τον εξής τύπο

$$D_{n,1}(x, y, \sigma(n, 1)) = L_{n,1}(x, y, \sigma(n+1, 1)) - L_{n,1}(x, y, \sigma(n, 1))$$

Επίσης η διαφορά των γκαουσιανών παρέχει μια πολύ καλή προσέγγιση στην εκτίμηση της Λαπλασιανή της Γκαουσιανής. (Laplacian of Gaussian), όπως έχει μελετηθεί από τον Linderberg. Μέσα από λεπτομερή πειραματικές συγκρίσεις ο Mikołajczyk βρήκε ότι τα maxima (μέγιστες τιμές) και τα minima (ελάχιστες τιμές) από λαπλασιανές μπορούν να παράγουν πιο σταθερά χαρακτηριστικά εικόνες σε σύγκριση με πλήθος άλλων πιθανών ανιχνευτών, όπως για παράδειγμα ο Harris.

L1-G1-G2

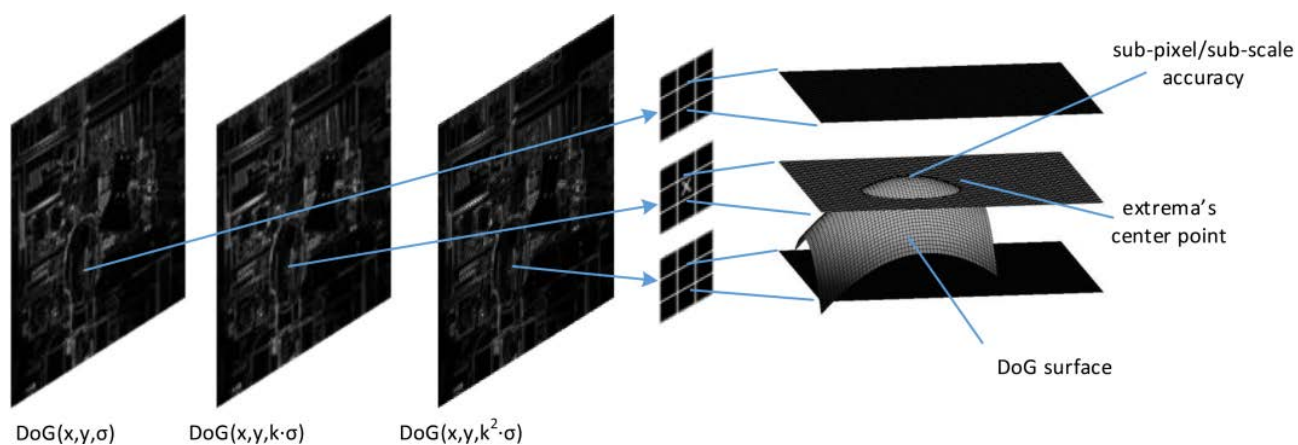


Εικόνα 4.3.1 Απεικόνιση DoG

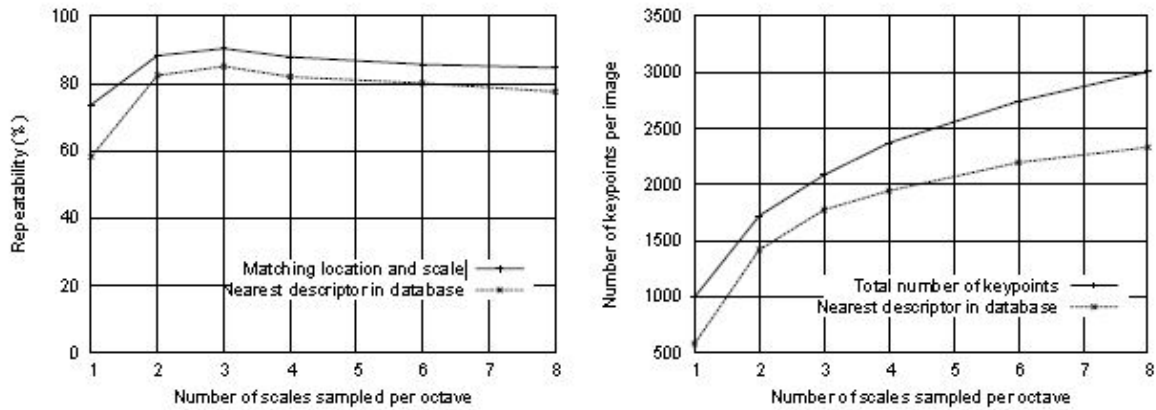


### 4.3.2 Εύρεση τοπικού μέγιστου (Keypoint Localization)

Αφού έχουν εξαχθεί οι DoG από τη πρώτη οκτάβα, το επόμενο στάδιο είναι η εύρεση πιθανών σημείων ενδιαφέροντος. Οι συγκρίσεις δεν γίνονται ξεχωριστά σε κάθε DoG αλλά αυτή τη φορά παίζουν ρόλο οι τρεις DoG όπου γίνονται οι βασικές συγκρίσεις. Το κάθε σημείο της μεσαίας DoG συγκρίνεται με τα 8 γειτονικά του μέσα στην ίδια DoG εικόνα και τα 9 γειτονικά σημεία στην πάνω και κάτω DoG. Επιλέγεται μόνο ότι είναι μεγαλύτερο ή μικρότερο από όλα τα υπόλοιπα σημεία. Το υπολογιστικό κόστος γι' αυτό τον έλεγχο είναι σχετικά χαμηλό.



Εικόνα 4.3 Εύρεση Μέγιστου και Ελάχιστου



Γράφημα Τρεις σχέσεις keypoint- χωροκλίμακας

Το πρώτο κομμάτι του γραφήματος 3 από τη μελέτη του D. Lowe, δείχνει το ποσοστό των κλειδιών όπου έχουν επαναλαμβανόμενα εντοπιστεί στην ίδια τοποθεσία και κλίμακα σε μια μετατροπή εικόνας. Όσο χαμηλώνει η γραμμή δείχνει ότι το ποσοστό των κλειδιών έχει ταιριάζει τους περιγραφείς σε μια μεγάλη βάση. Το δεύτερο γράφημα δείχνει ένα ολοκληρωτικό νούμερο από κλειδιά που εντοπίστηκαν από μια τυπική εικόνα σαν συνάρτηση των αριθμών των κλιμακωτών δειγμάτων.

Στο δεξί μέρος του γραφήματος δείχνει ότι το ποσοστό των κλειδιών τα οποία έχουν εντοπιστεί. Σε όλα αυτά τα παραδείγματα Χρησιμοποιούμε και τον παράγοντα που εφαρμόζουμε στον αλγόριθμο μας το  $\sigma = \sqrt{2}/2$ , που είναι η κλίμακα του σημείου-κλειδιού.

### 4.3.3 Ανάθεση προσανατολισμού (Orientation Assignment)

Ο προσανατολισμός είναι ένα από τα πιο σημαντικά κομμάτια του SIFT. Χάρη σε αυτό είναι εφικτό να βρεθεί σε τι προσανατολισμό είναι το κάθε σημείο ενδιαφέροντος, έτσι ώστε να μπορέσει να ταιριάζει μετά τους περιγραφείς χωρίς να υπάρχει πρόβλημα αναγνώρισης αντικειμένου με λήψη από διαφορετική οπτική γωνία, που προκαλεί στρόφη της εικόνας.

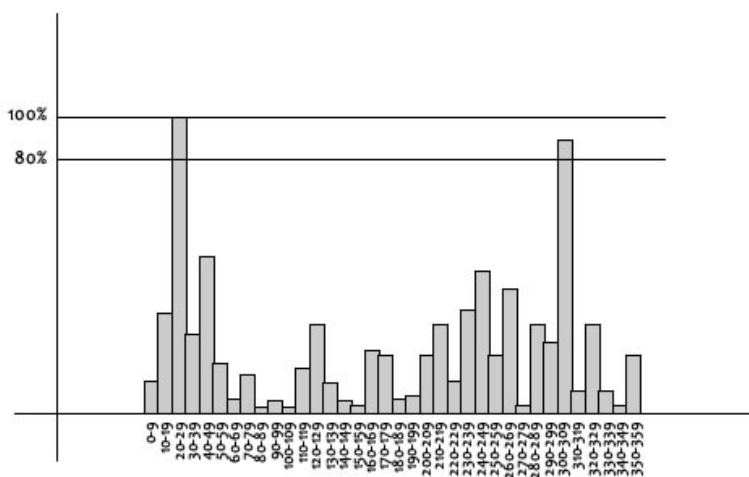
Για τον υπολογισμό του διανύσματος δίνονται οι τύποι του μέτρου και της κλίσης

$$m(x, y) = \sqrt{(L(x+1, y, \sigma) - L(x-1, y, \sigma))^2 + (L(x, y+1, \sigma) - L(x, y-1, \sigma))^2}$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1, \sigma) - L(x, y-1, \sigma)}{L(x+1, y, \sigma) - L(x-1, y, \sigma)} \right)$$

Για τον υπολογισμό του προσανατολισμού του κάθε σημείου ενδιαφέροντος λαμβάνονται οι τιμές της επιλεγμένης Γκαουσιανής εικόνας με  $\sigma=2$ . Για κάθε δείγμα  $L(x,y)$  υπολογίζεται το μέγεθος του στοιχείου  $m(x,y)$  (gradient magnitude) και ο προσανατολισμός  $\theta(x,y)$  (orientation) με τους παραπάνω τύπους. Ο υπολογισμός γίνεται για ολόκληρη την εικόνα και στη συνέχεια, όπου υπάρχει keypoint εξάγονται οι τιμές γύρω από αυτό, και φτιάχνεται το ιστόγραμμα του προσανατολισμού. Για την δημιουργία του ιστογράμματος εξάγεται με βάση τις γειτονικές κλίσεις του κάθε σημείου ενδιαφέροντος, η γωνία του σημείου-κλειδιού. Γίνεται η χρήση ενός πίνακα 7x7 γύρω από το keypoint, οπότε συνολικά υπάρχουν 49 τιμές με διάφορες γωνίες  $\theta$  που αντιστοιχούν σε αντίστοιχες κλίσεις  $m$ .

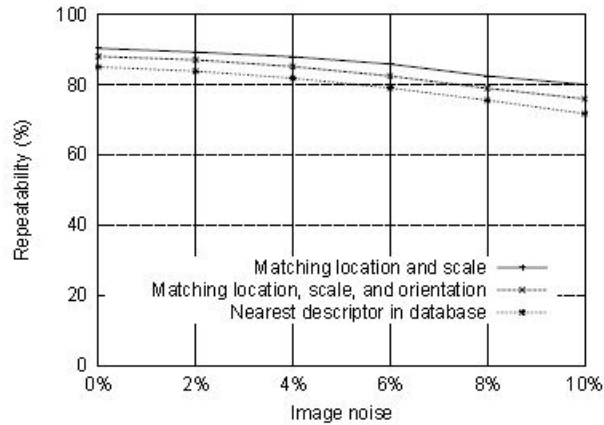
Ένα ιστόγραμμα προσανατολισμών γύρω από το σημείο-κλειδί είναι χρήσιμο για να δείξει ποιος προσανατολισμός υπερισχύει πάνω στο κλειδί. Αφού έχουν εξαχθεί οι



Γράφημα 4.3.1 Ιστόγραμμα ενός Keypoint με ένα επιπλέον από τα γειτονικά του στοιχεία

τιμές κλίσεων και βαρύτητας, δημιουργείται ένα ιστόγραμμα με 36 κελιά όπου το κάθε κελί αντιπροσωπεύει μια περιοχή γωνιών 10 μοιρών. Σε κάθε περιοχή γωνιών

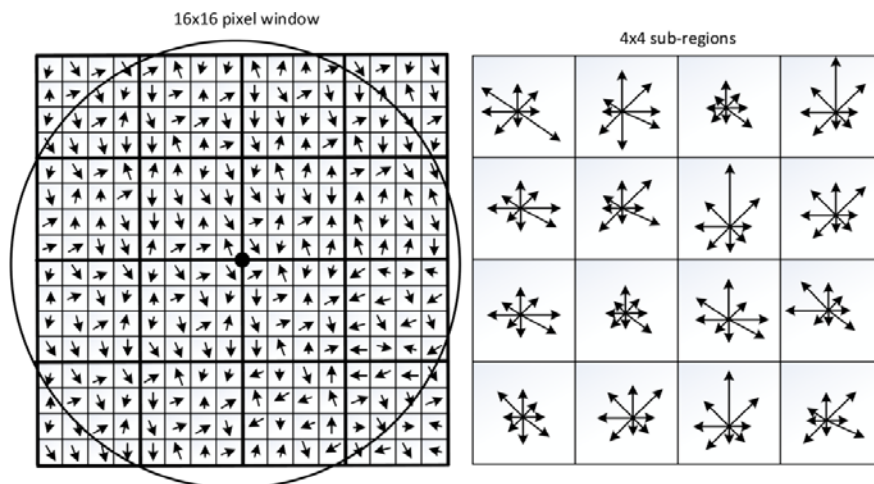
τοποθετείται η τιμή βαρύτητας  $m$  του κάθε ρικελ, έτσι υπολογίζεται αθροιστικά το συνολικό magnitude σε κάθε κελί (bin). Όποιο κελί έχει το μεγαλύτερο άθροισμα της κλίσης ( $\Sigma m$ ), αντιπροσωπεύει την κυρίαρχη γωνία.



Γράφημα 4.3.2

Το Γράφημα 4.3.2 δείχνει πειραματικά την σταθερότητα της θέσης, της κλίμακας και του προσανατολισμού, κάτω από διαφορετικές ποσότητες θορύβου. Η πρώτη γραμμή μας δείχνει τη σταθερότητα της τοποθεσίας του κλειδιού και την κλίμακα. Η δεύτερη γραμμή μας δείχνει τη σταθερότητα του ταιριάσματος όταν έχει ληφθεί υπόψη ότι ο προσανατολισμός μέσα σε σφάλμα 15 μοιρών.

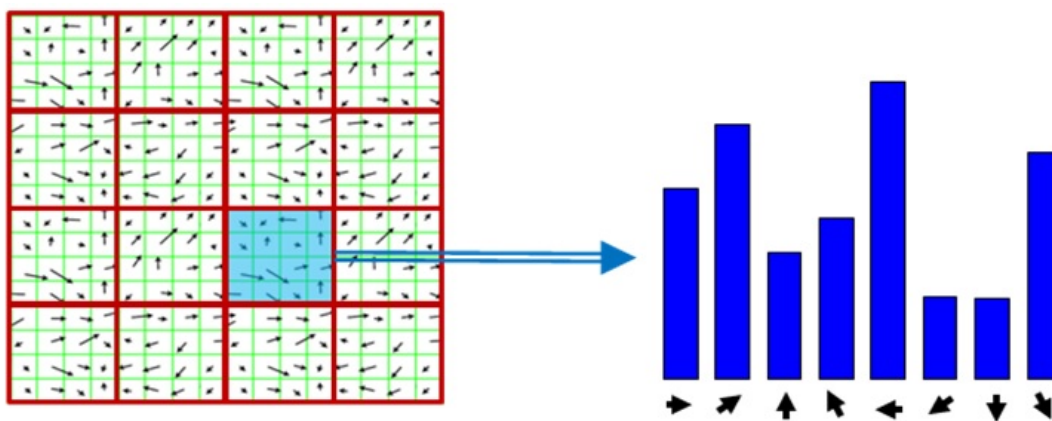
#### 4.3.4 Περιγραφέας



Εικόνα 4.3.4 Η μορφή του περιγραφέα

Στις προηγούμενες εφαρμογές είχε γίνει η εύρεση της θέσης, της κλίμακας και της κατεύθυνσης (orientation) σε κάθε σημείο ενδιαφέροντος. Το επόμενο βήμα είναι να υπολογιστεί ο περιγραφέας σε κάθε χαρακτηριστική τοπική περιοχή της εικόνας. Αυτό είναι μια πολύ διαφορετική διαδικασία. Χρεάζεται ένα τοπικό δισδιάστατο σύστημα συντεταγμένων το οποίο περιγράφει τη τοπική περιοχή της εικόνας και προσφέρει αμεταβλητότητα στην κλιμάκωση και στη στροφή. Μια καλή προσέγγιση θα μπορούσε να είναι η δειγματοποίηση των εντάσεων γύρω από το σημείο-κλειδί στην κατάλληλη κλίμακα.

Στην εικόνα 4.3.5 φαίνεται ο περιγραφέας που έχει βρεθεί γύρω από το κλειδί και έχει υπολογιστεί ο προσανατολισμός και τα μεγέθη γύρω από το δείγμα εικόνας. Τα βελάκια είναι διανύσματα βασισμένα στον προσανατολισμό της κλίσης γύρω από το χαρακτηριστικό σημείο. Στην εικόνα 4.3.5, στο πρώτο σκέλος, βλέπουμε 4 πίνακες 4x4 γύρω από το κλειδί, με το μήκος των βελών να παριστάνει το αντίστοιχο magnitude και στο δεύτερο σκέλος βλέπουμε έναν πίνακα 4x4, με τα αθροίσματα magnitudes κατανεμημένα στις αντίστοιχες κατευθύνσεις τους, σε ένα ιστόγραμμα προσανατολισμού με 8 κελιά (bins) γωνιών.



Εικόνα 4.3.5 Εξαγωγή ιστογράμματος 8 κάδων από κελί του περιγραφέα

### 4.3.5 Αναπαράσταση Περιγραφέα

Γύρω από το keypoint του περιγραφέα υπάρχουν οι κλίσεις και οι προσανατολισμοί από την κλίμακα του keypoint που έχει ληφθεί με το συγκεκριμένο επίπεδο θολότητας με βάση το  $\sigma$  του φίλτρου Gauss. Για να μην υπάρχει πρόβλημα προσανατολισμού του αντικείμενου στην αναγνώριση του, παίζει ρόλο και ο

προσανατολισμός του χαρακτηριστικού σημείου, που βρέθηκε προηγουμένως, κατά το προηγούμενο βήμα (4.3.3). Οι τιμές φωτεινότητας στρέφονται κατά γωνία ίση με τον προσανατολισμό του χαρακτηριστικού σημείου και στη συνέχεια εξάγεται ο περιγραφέας. Με τον τρόπο αυτό αν η εικόνα ενός αντικειμένου στραφεί κατά μία ορισμένη γωνία, τότε ο περιγραφέας που θα προκύψει θα είναι αρκετά όμοιος με αυτόν που προκύπτει από την εικόνα που δεν έχει περιστραφεί.

Για την εξαγωγή του περιγραφέα λαμβάνεται ένα ιστόγραμμα προσανατολισμού με οκτώ κελιά, σε κάθε περιοχή 4x4 από αυτές που φαίνονται στο σχήμα 4.3.5. Κάθε ιστόγραμμα προσανατολισμού θα περιέχει σε κάθε κελί το συσσωρευμένο άθροισμα των τιμών των κλίσεων (gradient magnitudes). Άρα, θα λάβουμε σε κάθε περιοχή 4x4 οκτώ τιμές. Επειδή υπάρχουν 16 τέτοιες περιοχές γύρω από το keypoint, ο συνολικός περιγραφέας αποτελείται από  $16 \times 8 = 128$  τιμές.

Στη τελική του μορφή ο περιγραφέας τροποποιείται για να μειώσει τις επιπτώσεις του φωτισμού. Πρώτον ο περιγραφέας απλοποιείται στην μονάδα μήκους. Μια αλλαγή στην αντίθεση της εικόνας, στην οποία η κάθε τιμή pixel μπορεί να πολλαπλασιαστεί από μια σταθερά, θα πολλαπλασιάσει τις κλίσεις προκαλώντας αλλαγή της αντίθεσης θα ακυρώσει όλη την απλοποίηση του φορέα. Μια αλλαγή φωτεινότητας όμως δεν μπορεί να επηρεάσει τις κλίσεις εφόσον επηρεάζονται όλα τα pixel.

Για αυτό το λόγο ο περιγραφέας αποδεικνύει ότι είναι αναλλοίωτος πάνω σε αλλαγές στο φωτισμό. Είναι δυνατόν όμως να προκύψουν διάφοροι κορεσμοί πάνω από τη λήψη της φωτογραφίας και να μην υπάρχει σωστό αποτέλεσμα από τις αλλαγές φωτισμού τρισδιάστατων αντικειμένων. Οι επιδράσεις αυτές ωστόσο είναι μικρές και μπορεί να φανεί σε τιμές μικρών κλίσεων. Για καλύτερη απόδοση και αποφυγή σφαλμάτων πάνω στις κλίσεις μπορεί να μειωθεί η επιρροή των μεγάλων τιμών κλίσεων και να οριστεί κατωφλίωση.

## 5. Δημιουργία Εφαρμογής Αλγόριθμου SIFT

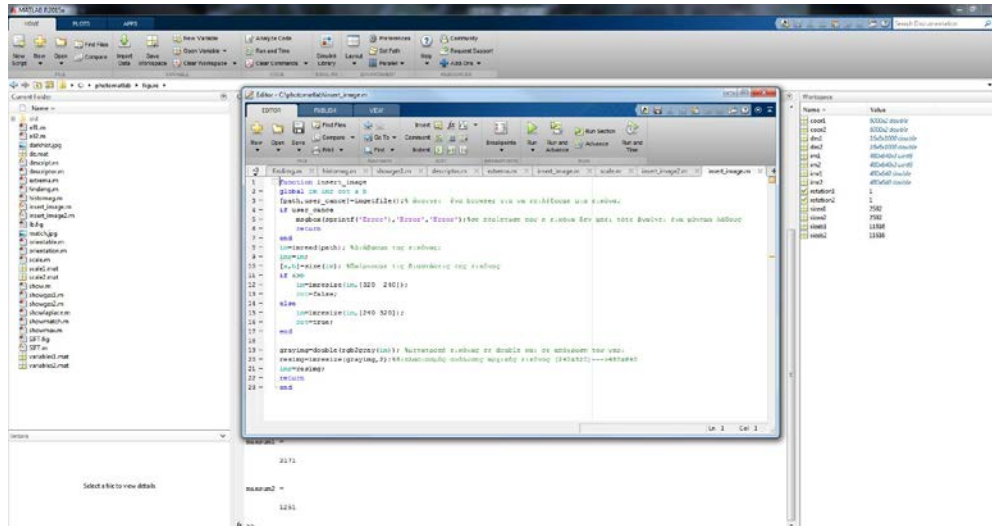
### 5.1 Μαθηματικά προγράμματα

Για επεξεργασία εικόνας κατάλληλα προγράμματα είναι τα: Mathematica, Mathcad και Maple και Matlab.

Το Matlab παρέχει πολλές μαθηματικές δυνατότητες και ευελιξία στην επεξεργασία των μαθηματικών πινάκων. Οι ανανεωμένες του εκδόσεις (συνήθως βγαίνουν δύο κάθε χρόνο) παρέχουν πλήθος εντολών και εργαλείων σε όλους τους τομείς, όπως επίσης και στο αντικείμενο της ψηφιακής εικόνας. Το Image Toolbox από την έκδοση του 2013 και μετά είναι ένα πολύ καλό εργαλείο για την μελέτη και τον πειραματισμό διαφόρων αλγορίθμων πάνω στην εικόνα. Είναι ένα επαρκές προγραμματιστικό μαθηματικό εργαλείο και στην ψηφιακή επεξεργασία εικόνας. Κάποιες λειτουργίες, όπως εφαρμογή μιας μάσκας πάνω σε μια εικόνα, παρέχεται από έτοιμες εντολές με τη δυνατότητα της ακρίβειας δεκαδικών ψηφίων. Η γλώσσα προγραμματισμού της είναι κοινή με τις βασικές που είναι ευέλικτες για έναν προγραμματιστή. Στο παράρτημα κώδικα υπάρχουν σχόλια για την πλήρη κατανόηση της λειτουργίας του ανιχνευτή SIFT.

### 5.2 Matlab

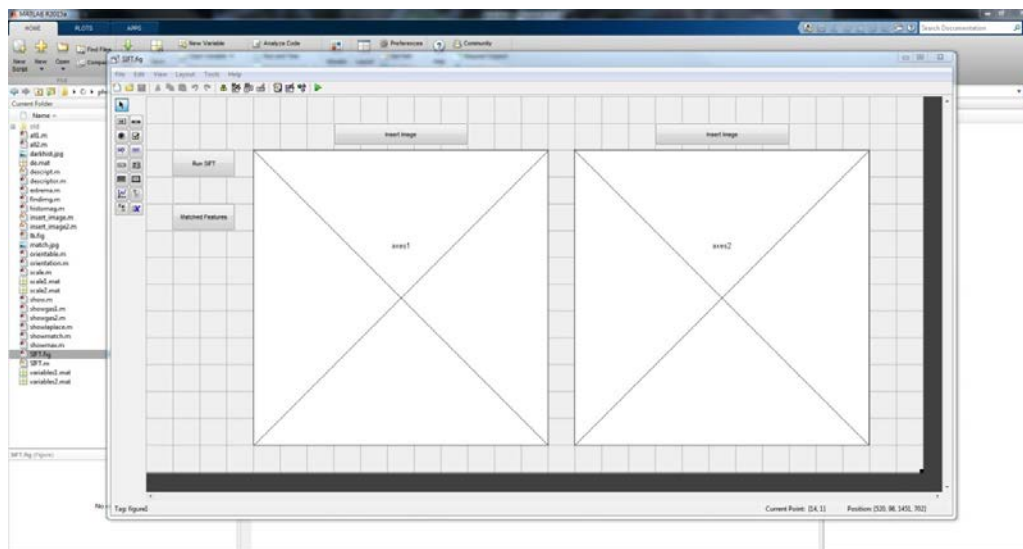
Το Workspace είναι το βασικό κομμάτι που μπορεί κανείς να τρέξει τις απλές εντολές και τις συναρτήσεις. Στο περιβάλλον αυτό υπάρχει ο χώρος με τη δυνατότητα χρήσης εντολών, το κομμάτι των αποθηκευμένων μεταβλητών και το κομμάτι του ιστορικού. Οι μεταβλητές παραμένουν στο πρόγραμμα όσο αυτό είναι σε λειτουργία. Ο κώδικας είναι χωρισμένος σε διάφορα m files τα οποία αρχεία τρέχουν με την ονομασία τους όταν πληκτρολογήσουμε στο περιβάλλον των εντολών. Επίσης ένα m file μπορεί να λειτουργήσει σαν συνάρτηση, ένα πολύ σημαντικό εργαλείο προγραμματισμού για να εισάγουμε και να εξάγουμε δεδομένα, για αναδρομικές λειτουργίες κτλ. Βασισμένη πάνω στην έρευνα του Lowe χωρίσαμε την κάθε λειτουργία σε m file με ονομασίες πάνω στα βήματα αυτά. Στη συνέχεια αφού ολοκληρώθηκε η λειτουργία του κώδικα και ο έλεγχος του ανιχνευτή, δημιουργήσαμε ένα γραφικό περιβάλλον για την απεικόνιση και την ευκολία της λειτουργίας.



Εικόνα 5.2.1 Το Περιβάλλον του Matlab

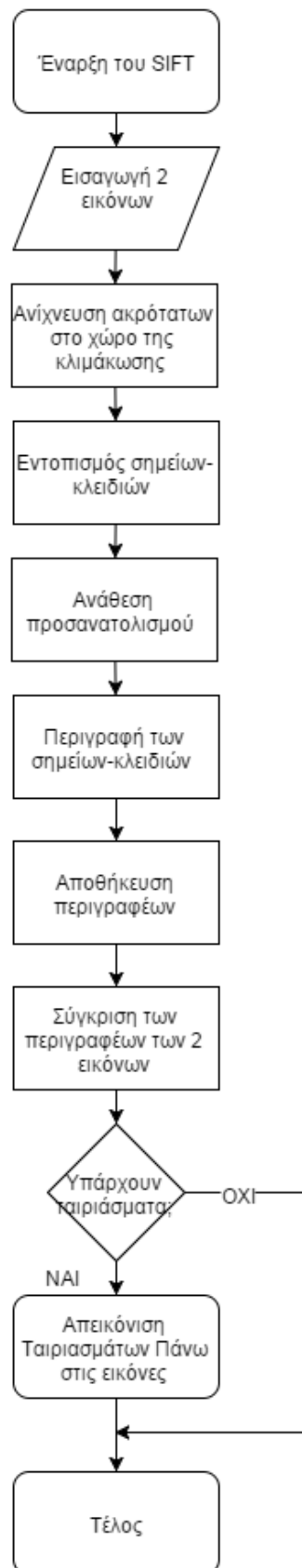
Το GUI (γραφικό περιβάλλον του Matlab) δεν χρησιμοποιεί τις μεταβλητές του workspace, εξαιτίας διαφορετικών επιλογών έπρεπε να κατανεμηθεί διαφορετικά ο κώδικας για την ορθή λειτουργία του. Παρουσιάζεται το αρχικό κομμάτι με επίδειξη εικόνων βήμα-βήμα, δηλαδή ποιο είναι το αποτέλεσμα εικονικά μετά από την επεξεργασία δεδομένων, και μετά τη λειτουργία του GUI εικονικού περιβάλλοντος που δέχεται δυο εικόνες και εξάγει τα συμπεράσματα του.

Εικόνα 5.2.2 Σχεδιασμός του γραφικού περιβάλλοντος



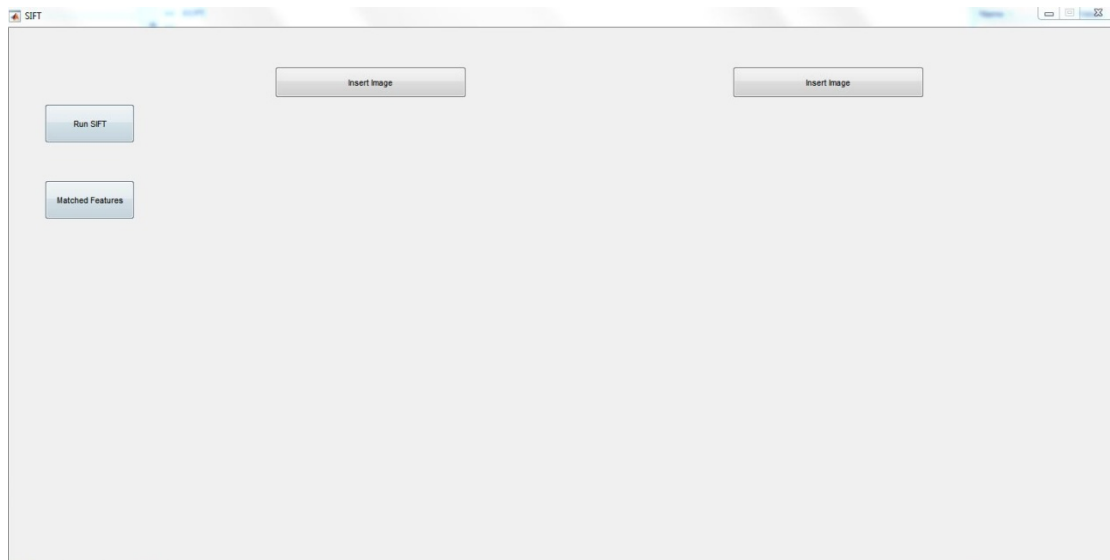


### 5.3 Διάγραμμα Ροής



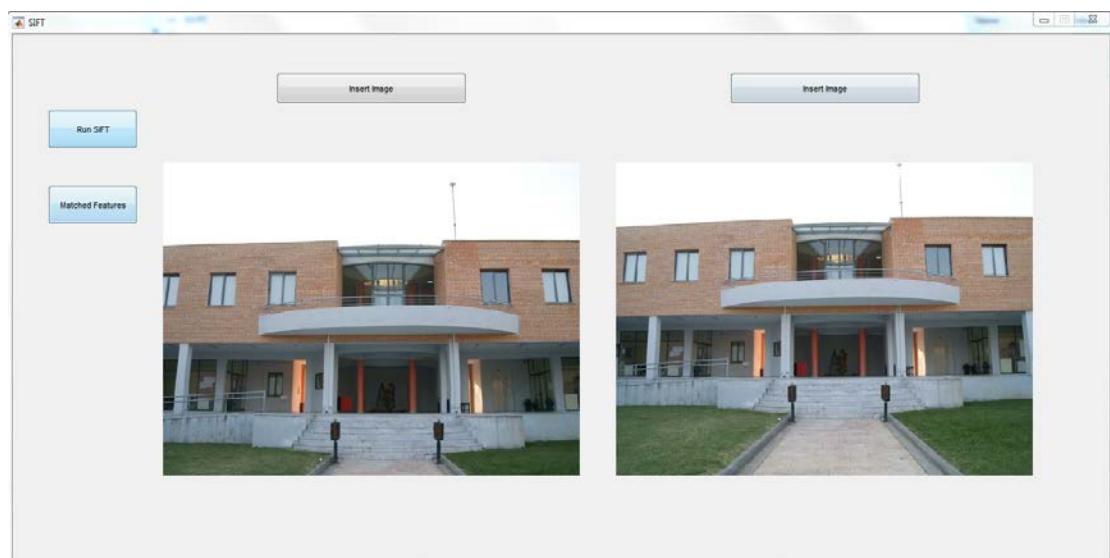
## 5.4 Λειτουργία του προγράμματος

Πρώτο βήμα είναι να τρέξουμε το πρόγραμμα μας. Μέσα από το workspace του Matlab βρισκόμαστε στο φάκελο της εργασίας και απλά πληκτρολογούμε SIFT. Ανοίγει το γραφικό περιβάλλον, όπως το βλέπουμε στην παρακάτω εικόνα :



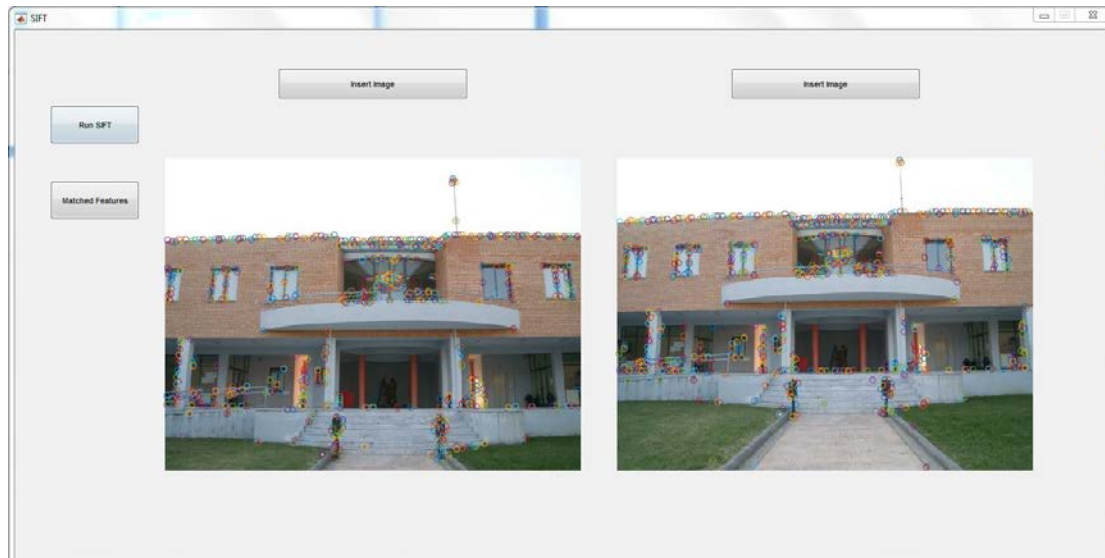
Εικόνα 5.4.1 Γραφικό Περιβάλλον

Στη συνέχεια πατάμε τα κουμπιά Insert Image για να εισάγουμε 2 εικόνες. Για παράδειγμα εδώ έχω 2 φωτογραφίες με ένα τοπίο το οποίο έχει τραβηχτεί με λίγο διαφορετική γωνία λήψης όπως βλέπουμε στην κάτω εικόνα.



Εικόνα 5.4.2 Εισαγωγή εικόνων

Όσο έχει γίνει η εισαγωγή των εικόνων πίσω από το πρόγραμμα οι εικόνες έχουν αποθηκευτεί και είναι έτοιμες για να περάσουν από την επεξεργασία των σταδίων του αλγόριθμου SIFT. Πατώντας το κουμπί Run SIFT ξεκινάει η επεξεργασία και των 2 εικόνων και στη συνέχεια όταν ολοκληρωθεί έχουμε ένα οπτικό αποτέλεσμα στην παρακάτω εικόνα. Έχουν βρεθεί 671 στην πρώτη και 698 στην δεύτερη εικόνα Keypoints.



**Εικόνα 5.4.3** Εύρεση των Keypoints

Ο αλγόριθμος έχει εκτελεστεί και εμφανίζονται τα Keypoints. Πατώντας το δεύτερο κουμπί Matched Features τρέχει από πίσω ο κώδικας για τη σύγκριση των περιγραφών των 2 εικόνων που έχουν αποθηκευτεί σε βάση και όταν βρεθούν τα κοινά σημεία μας ανοίγει ένα παράθυρο με την απεικόνιση των ταιριασμάτων. Το αποτέλεσμα φαίνεται στην εικόνα 5.4.4.



**Εικόνα 5.4.4** Ταίριασμα χαρακτηριστικών των 2 εικόνων

Όταν εισαχθούν οι εικόνες αποθηκεύονται σαν global Μεταβλητές και μεταφέρονται στο workspace του Matlab σε περίπτωση που χρειαστούν για εκτενέστερη μελέτη στο περιβάλλον από το χρήστη. Οι εικόνες αποθηκεύονται αρχικά σαν έγχρωμες αλλά και σαν δισδιάστατοι πίνακες ανάλυσης 480x640 όπου είναι σε grayscale μορφή. Όταν πατηθεί το Run SIFT τότε γίνεται η κλήση των δυο m files που περιέχουν τα στάδια του ανιχνευτή SIFT, το all1.m για την πρώτη εικόνα, και το all2.m που έχει τα στάδια για την δεύτερη εικόνα. Στην ουσία τρέχουν δυο φορές τα στάδια με διαφορετικές εικόνες. Απλά, μέσα στο κάλεσμά τους γίνεται ξεχωριστή αποθήκευση δεδομένων, για να υπάρχουν κάποια στοιχεία για μελέτη αν χρειαστεί και οι περιγραφείς που θα χρειαστούν μετά για σύγκριση. Το all1 και το all2 περιέχουν τα εξής m files : scale.m, extrema.m, orientation.m, descriptor.m.

Το scale.m όπου υπάρχει στο παράρτημα του κώδικα ξεκινάει πρώτα με τη δημιουργία των Γκαουσιανών. Βήμα-βήμα δημιουργείται το  $\sigma$ , ο πίνακας-φίλτρο με την βοήθεια της εντολής fspecial όπου ορίζεται το μέγεθος του πίνακα και το  $\sigma$ , και μετά, με την εντολή imfilter, πραγματοποιείται η συνέλιξη όπως έχει αναφερθεί στο κεφάλαιο 4.3.1. Δημιουργείται μια οκτάβα 5 Γκαουσιανών και στη συνέχεια μια δεύτερη οκτάβα με μειωμένη ανάλυση κατά το μισό της εικόνας.

Η δημιουργία της δεύτερης οκτάβας γίνεται για λόγους μελέτης του αλγόριθμου και όχι για τη λειτουργία της βασικής εφαρμογής. Στο workspace του Matlab μπορεί να μελετηθεί και να απεικονιστεί το αντικείμενο της μελέτης με τα m files showgas1.m για την πρώτη οκτάβα και το showgas2.m (όπου υπάρχουν στο παράρτημα του κώδικα) για την δεύτερη οκτάβα. Το αποτέλεσμα είναι στις εικόνες 5.4.5 και 5.4.6.

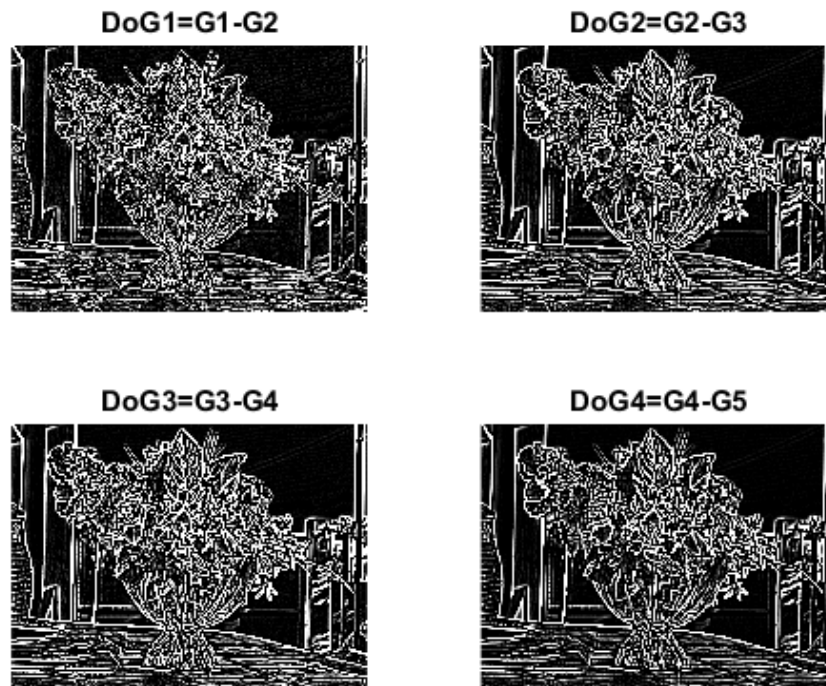


Εικόνα 5.4.5 Πρώτη οκτάβα



Εικόνα 5.4.6 Δεύτερη οκτάβα

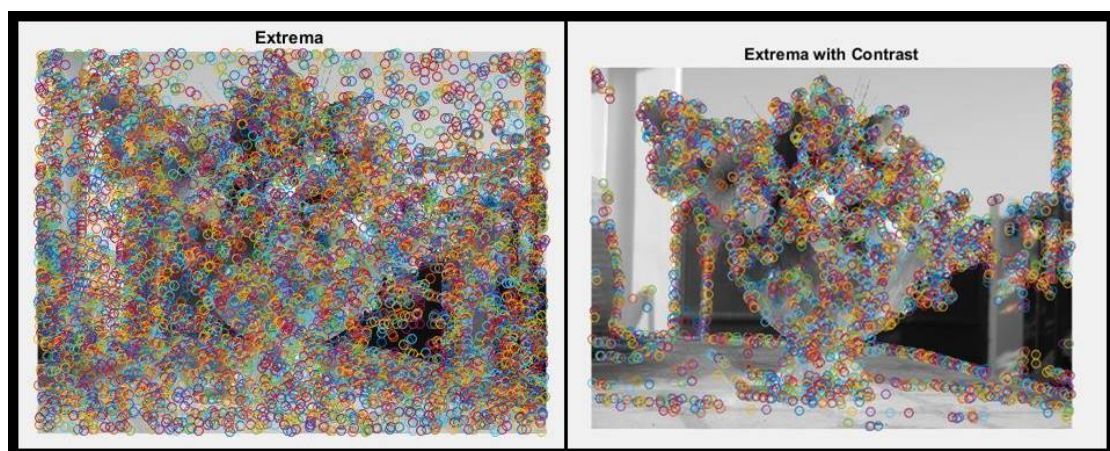
Επόμενο βήμα του `scale.m` είναι η δημιουργία των DoG, αυτό είναι εφικτό με την αφαίρεση των γκαουσιανών. Από τις 5 Gaussians εξάγουμε 4 DoG όπου και πάλι μπορούν να απεικονιστούν με `showdog.m` του παραρτήματος κώδικα, το αποτέλεσμα είναι στην εικόνα 5.4.7.



Εικόνα 5.4.7 Διαφορές των Γκαουσιανών

Αφού ολοκληρωθεί η `scale.m` ξεκινάει το στάδιο της εύρεσης των ακρότατων με το αρχείο `extrema.m`, που εμφανίζεται στο παράρτημα του κώδικα. Όπως φαίνεται από την εικόνα 3.4.2 γίνεται μια αναζήτηση μέγιστου ή ελάχιστου σημείου μεταξύ των 3 DoG. Προγραμματιστικά δημιουργούνται κάποιες μεταβλητές για το μέτρημα των σημείων ενδιαφέροντος, δηλώνονται πίνακες όπου ο ένας έχει τα μέγιστα και ο άλλος έχει τα ελάχιστα. Επίσης δημιουργούνται βοηθητικοί πίνακες που παίρνουν μέσα στη δομή επανάληψης το χαρακτηριστικό και τα γειτονικά του. Μέσα στη δομή επανάληψης αρχικοποιούνται οι προσωρινοί πίνακες 3x3 των δύο Dog με κεντρικό σημείο, το σημείο που βρίσκεται η συνθήκη. Αφού αποθηκευτούν οι τιμές μένει να περάσει από ένα κριτήριο αυτό το σημείο για να θεωρηθεί μέγιστο ή ελάχιστο. Πρώτα μπαίνουν 3 συνθήκες που συγκρίνουν αν αυτό το σημείο θεωρείται μέγιστο.

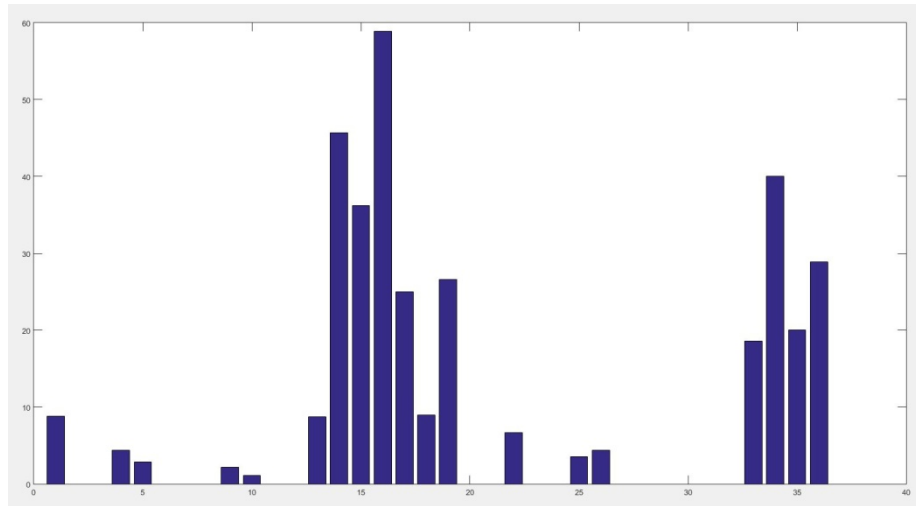
Συγκρίνεται με τα δικά του γειτονικά στοιχεία και μετά με το εύρος των άλλων δύο. Η βασική DoG που εξάγει τα σημεία ενδιαφέροντος είναι η μεσαία από τις τρεις DoG, η δεύτερη που έχει το  $\sigma$  ίσο με 1. Αν ισχύει η συνθήκη αυτή τότε σε ένα προσωρινό πίνακα αποθηκεύεται η τιμή 255 σαν σημείο αναφοράς ότι υπάρχει εκεί, επίσης βοηθάει για την απεικόνιση και τη μελέτη αν χρειαστεί να δει κάποιος τα μέγιστα μόνο. Μετά από την εύρεση των μέγιστων, ξεκινάει η συνθήκη για την εύρεση ελάχιστων, αυτή τη φορά συγκρίνει αν το στοιχείο είναι μικρότερο πρώτα από τη δικιά του γειτονιά και μετέπειτα με τα ελάχιστα των υπόλοιπων 2. Αποθηκεύεται ένας πίνακας αναφοράς ελάχιστων με την τιμή 255 και μετά όταν κλείσει αυτή η συνθήκη ξεκινάει μια συνθήκη που βλέπει που είναι τα σημεία τα ελάχιστα, τα μέγιστα και αποθηκεύει τις πραγματικές τιμές σε έναν νέο προσωρινό πίνακα. Αφού έχουν βρεθεί τα υποψήφια κλειδιά maxima & minima το επόμενο βήμα είναι η απαλοιφή μη έγκυρων με το κριτήριο της αντίθεσης της εικόνας (κεφάλαιο 2.1). Το κριτήριο της αντίθεσης δεν θα είναι για τις τιμές φωτεινότητας της εικόνας αλλά για τα πλέον επεξεργασμένα στοιχεία που έχουν εξαχθεί από την διαδικασία της εύρεσης μέγιστων και ελάχιστων ακρότατων της εικόνας. Αν τα νούμερα είναι μικρότερα ή μεγαλύτερα από μια τιμή τότε καθορίζεται πια θα μείνουν και ποια θα απαλείφουν. Η τιμή της αντίθεσης είναι το πέντε, μετά από δοκιμές έχει φανεί ότι δίνει το πιο ξεκάθαρο αποτέλεσμα. Ο contrastable είναι πλέον το τελικό αποτέλεσμα των σημείων ενδιαφέροντος. Στο περιβάλλον του workspace υπάρχει η δυνατότητα να δει ο χρήστης με το showcontrast.m πως είναι η εύρεση ακρότατων χωρίς το κριτήριο της αντίθεσης και με το κριτήριο αυτό στην εικόνα 5.4.8



Εικόνα 5.4.8 Ανίχνευση σημείων ενδιαφέροντος και απαλοιφή με αντίθεση

Επόμενο στάδιο είναι η ανάθεση του προσανατολισμού από το κεφάλαιο 4.3.3. Το αρχείο `orientation` από το παράρτημα κώδικα υπολογίζει με βάση τον τύπο του κεφαλαίου τα `magnitude` και τις κλίσεις από την γκαουσιανή με  $\sigma = 1$ . Εξάγονται 2 πίνακες, ο ένας με τα μεγέθη και ο άλλος με τις γωνίες. Για την εύρεση του προσανατολισμού χτίζεται ένα ιστόγραμμα με bins γωνιών έτσι ώστε να εισαχθούν τα `magnitude` στις αντίστοιχες γωνίες. Αυτό είναι εφικτό με τον εξής τρόπο, ξεκινάει μια δομή επανάληψης που ελέγχει που είναι τα σημεία ενδιαφέροντος με βάση τον `contrastable` και σε κάθε στοιχείο αποθηκεύει τα γειτονικά του, ένα εύρος  $7 \times 7$  γύρω του. Έχοντας τώρα τις τιμές των βαρών και των κλίσεων σε δύο διαφορετικούς πίνακες με κεντρικό το σημείο ενδιαφέροντος είναι πλέον να γίνει ένα ιστόγραμμα με τις 49 τιμές των βαρυτήτων πάνω στο εύρος των αντιστοιχών γωνιών. Η κάθε γωνία μετά την εφαρμογή του τύπου στο κεφάλαιο 4.3.3, έχει ως συνήθως ένα νούμερο με κάποια δεκαδικά ψηφία ακριβείας, για το λόγο αυτό γίνεται μια στρογγυλοποίηση και απλοποίηση έτσι ώστε να είναι ένα απλό νούμερο που θα απεικονίζει το `index` του ιστογράμματος. Μέσα στο κώδικα του `orientation.m` η μορφή γωνιών από 0-360 εφαρμόζεται ως 1-36 bins και ανάλογα σε αυτούς τους κάδους τοποθετούνται οι αντίστοιχες τιμές `magnitude`. Μέσα στη δομή επανάληψης εύρεσης πιθανού σημείου ενδιαφέροντος, εφαρμόζεται μια δομή επανάληψης γύρω από το χαρακτηριστικό σημείο για τη δημιουργία του ιστογράμματος. Στη συνέχεια αφού τελειώσει το χτίσιμο του ιστογράμματος και η δομή επανάληψης, ξεκινάει η επόμενη δομή επανάληψης που αθροίζει τα μεγέθη των bins και τα τοποθετεί σε ένα μονοδιάστατο πίνακα 36 θέσεων, μετά το τέλος αυτής της δομής ξεκινάει η τελική δομή επανάληψης όπου βρίσκει το μέγιστο της μονοδιάστατης αθροιστικής μορφής ιστόγραμμα για το μέγιστο `peak` που είναι η απεικόνιση της ισχύουσας κλίσης, ο βασικός προσανατολισμός όπου ανήκει το αντίστοιχο `keypoint`. Μετά από αυτό το βήμα γίνεται αποθήκευση συνταγμένων εικόνας, με το βάρος και την κλίση του διανύσματος του `keypoint` στον πίνακα `logtable`. Για Μελέτη ενός ιστογράμματος γύρω από ένα σημείο ενδιαφέροντος έχει δημιουργηθεί το αρχείο `histomag.m` όπου υπάρχει στο παράρτημα του κώδικα. Παίρνει τις τιμές από τους πίνακες προσωρινής αποθήκευσης τα δεδομένα και απεικονίζει σε ένα `figure` του Matlab το ιστόγραμμα του με τη βοήθεια της εντολής `bar()` όπως φαίνεται στην εικόνα 5.4.9.





Εικόνα 5.4.9. Ιστόγραμμα προσανατολισμού ενός keypoint

Επόμενο βήμα από τη θεωρία του Lowe είναι το τελικό στάδιο, η δημιουργία του περιγραφέα όπου βρίσκεται στο παράρτημα του κώδικα το αρχείο `descriptor.m`. Όπως αναφέρεται στο κεφάλαιο 4.3.4, χρησιμοποιείται ο πίνακας σημείων ενδιαφέροντος από το πίνακα μεταβλητών `contrastable`. Αφού γίνει η εύρεση σε μια δομή επανάληψης του keypoint αποθηκεύεται γύρω από το σημείο αυτό το εύρος των γειτονικών σημείων όπου είναι ένας  $16 \times 16$  πίνακας γωνιών, και ένας αντίστοιχος πίνακας κλίσεων. Για την εξαγωγή του περιγραφέα είναι απαραίτητη η δημιουργία ιστογραμμάτων μέσα στα πλαίσια των πινάκων αυτών. Είναι εφικτό μέσα στη δομή επανάληψης να εισαχθούν προσωρινές τιμές πινάκων σε μια συνάρτηση όπου έχει χτιστεί για να δέχεται 2 πίνακες κλίσεων γωνιών και να εξάγει τον περιγραφέα του σημείου ενδιαφέροντος. Γίνεται η κλίση του `descriptor.m` (παράρτημα κώδικα) και αφού δεχθεί τους πίνακες τους χωρίζει σε 4 πίνακες (64 τιμές ο κάθε ένας)  $4 \times 4$  (16 τιμές. Σε κάθε πίνακα  $4 \times 4$  (από τους 16 συνολικά πίνακες) παίρνονται οι τιμές γωνιών και γίνεται μια αντιστοίχιση των ευρών τους σε 8 ομάδες γωνιών, έτσι ώστε να χτιστεί ένα ιστόγραμμα για τον πίνακα αυτό, με γέμισμα των αντίστοιχων `magnitudes`. Μέσα σε μια δομή επανάληψης που στοχεύει κάθε φορά σε σειρά κομμάτι ενός πίνακα  $4 \times 4$  για τη δημιουργία ιστογραμμάτων έτσι ώστε να εξαχθούν 16 ιστογράμματα. Μέσα στην δομή επανάληψης για την επεξεργασία του ιστογράμματος, οι γωνίες απλοποιούνται και συγκρίνονται έτσι ώστε να ανήκουν σε κάδους γωνιών, και αφού έχουν συγκριθεί και ανήκουν σε αυτό το εύρος, σε ένα νέο πίνακα `descr2` όπου εισάγεται και η ανάλογη τιμή του `magnitude`. Αφού εξαχθεί ένας

μονοδιάστατος πίνακας με 8 θέσεις γωνιών και αθροίσματα magnitude τοποθετείται σε σειρά σε ένα νέο πίνακα 16x8 όπου είναι η μορφή του περιγραφέα. Ο περιγραφέας είναι η έξοδος συνάρτησης από το `descriptor.m` όπου στο `descriptor.m` στη συνέχεια αποθηκεύει σε μια μορφή βάσης δεδομένων τον πίνακα του περιγραφέα 16x8 σε μορφή 3σδιατατου πίνακα `descriptors(16x8x1000)`. Αυτή είναι η μορφή βάσης περιγραφέων της εικόνας και αποθηκεύεται επίσης και σε μεταβλητή σε αρχείο Matlab για το επόμενο βήμα του λογισμικού όπου είναι η σύγκριση και ταυτοποίηση περιγραφέων των εικόνων.

Αφού αυτή η διαδικασία έχει τρέξει και για τις δύο εικόνες που έχουμε εισάγει στο GUI της εφαρμογής SIFT και έχει αποθηκεύσει τους περιγραφείς σε μορφές τρισδιάστατων πινάκων με επιπλέον πληροφορίες όπως είναι και οι συντεταγμένες, στο περιβάλλον του προγράμματος έχουν εμφανιστεί πάνω στις εικόνες τα keypoints από τα στοιχεία των αντίστοιχων πινάκων τους. Στο κουμπί του λογισμικού Matched Features βρίσκεται από πίσω ένα m file, το `finding.m` από το παράρτημα κώδικα, το οποίο παίρνει τους 3διαστατους πίνακες από τις αποθηκευμένες δομές του Matlab, και συγκρίνει αν οι περιγραφείς μοιάζουν. Ο πρώτος έλεγχος είναι αν δομικά μοιάζουν, αν έχουν σε κάποια σημεία μηδενικές τιμές. Αν μοιάζουν τότε γίνεται η αφαίρεση των περιγραφέων μέσα σε απόλυτη τιμή και αν το αποτέλεσμα της αφαίρεσης είναι σχετικά μικρό τότε υπάρχει ταίριασμα. Μετά ελέγχεται και με απόσταση manhattan η απόσταση μεταξύ των χαρακτηριστικών, αν είναι μικρή και αυτή τότε υπάρχει ταυτοποίηση των ταιριασμάτων. Δημιουργούνται 2 πίνακες με τις συντεταγμένες των ταιριασμάτων με τη σειρά έτσι ώστε μετά να μπορέσουν να απεικονίσουν τα ταιριάσματα των εικόνων με ένα figure. Αφού εξαχθούν Οι συντεταγμένες των δύο εικόνων, με το `showmatch.m` του παραρτήματος κώδικα είναι εφικτό το οπτικό αποτέλεσμα όπως φαίνεται στην εικόνα 5.4.10



Εικόνα 5.4.10 Ταίριασμα σημείων σε 2 εικόνες

## Συμπεράσματα

Μέσα από την υλοποίηση του αλγόριθμου στο Matlab η λειτουργία του ανιχνευτή γίνεται πιο κατανοητή σε έναν προγραμματιστή δίνοντας του την δυνατότητα της έρευνας σε βάθος. Αφού έχουν εκτελεστεί οι εντολές βήμα-βήμα του αλγόριθμου, δίνεται η δυνατότητα της επίδειξης της λειτουργίας του ανιχνευτή SIFT. Η όλη θεωρία του ανιχνευτή αποδεικνύει το πώς αυτά τα στάδια αποδίδουν με αυτόν τον τρόπο, που θεωρείται αρκετά ευέλικτος. Έτσι, για παράδειγμα η χρήση των DoG αντί της χρήσης των Λαπλασιανών αποδίδει πιο γρήγορα με ικανοποιητική ακρίβεια χωρίς σημαντικές απώλειες, με συντομότερους υπολογισμούς.

Τα σημεία ενδιαφέροντος του ανιχνευτή SIFT είναι χρήσιμα εξαιτίας της διακριτικότητάς τους, η οποία επιτρέπει το σωστό ταίριασμα (matching) με άλλα σημεία ενδιαφέροντος που μπορούν να βρίσκονται σε μεγαλύτερες βάσεις δεδομένων ή να εξάγονται από παρόμοιες εικόνες που έχουν υποατσει μετασχηματισμούς. Η διακριτικότητα επιτυγχάνεται και από το σχεδιασμό του περιγραφέα, ο οποίος εξάγεται από τα μεγέθη των βαρυτήτων και των κλίσεων των τοπικών περιοχών της εικόνας. Τα σημεία ενδιαφέροντος φαίνονται να είναι αναλλοίωτα στην περιστροφή της εικόνας και στην αλλαγή κλίμακας, δείχνοντας έτσι πόσο καλύτερος είναι ο ανιχνευτής σε σχέση με άλλους. Επίσης, εμφανίζεται ισχυρός σε ένα μεγάλο εύρος αλλαγής της γωνίας θέασης, στην προσθήκη θορύβου και σε αλλαγές φωτεινότητας. Μεγάλο πλήθος χαρακτηριστικών σημείων μπορούν να εξαχθούν από τυπικά μεγέθη εικόνων. Ο υπολογισμός είναι αποτελεσματικός, οπότε αρκετές χιλιάδες περιγραφείς μπορούν να εξαχθούν από μια τυπική εικόνα στα τρέχοντα υπολογιστικά συστήματα και να χρησιμοποιηθούν για αναγνώριση.

## Βιβλιογραφία

1. Lindeberg T. Scale Selection. *Comput Vis A Ref Guid.* 2014:701-713. doi:10.1007/978-0-387-31439-6.Scale.
2. Group BE. Οπτική αναγνώριση για υποστηριξη αυτονομισ πτησης ελικοπτερου. 2010:1-6.
3. Gonzales R, Woods R, Eddins S. Digital Image Processing with Matlab. 2004.
4. Lin H. Computer Vision Image Features Image Features ? *October.* 2011.
5. Crandall D, Backstrom L, Huttenlocher D, Kleinberg J. Mapping the World ' s Photos. *Www.* 2009.
6. Alonso-Fernandez F, Tome-Gonzalez P, Ruiz-Albacete V, Ortega-Garcia J. Iris recognition based on SIFT features. *Biometrics Identity Secur Bids 2009 Int Conf.* 2009;387(1-3):175-184. doi:10.1109/BIDS.2009.5507529.
7. Andersen V, Pellarin L. Scale Invariant Feature Transform ( SIFT ): Performance and Application. 2006:1-14.
8. Azad P, Asfour T, Dillmann R. Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. *2009 IEEE/RSJ Int Conf Intell Robot Syst IROS 2009.* 2009:4275-4280. doi:10.1109/IROS.2009.5354611.
9. Berretti S, Ben Amor B, Daoudi M, Bimbo A. 3D Facial Expression Recognition Using SIFT Descriptors of Automatically Detected Keypoints. *Vis Comput.* 2011;27(11):1021-1036. doi:10.1007/s00371-011-0611-x.
10. Gerig G. Sift features.
11. Guis B. Building GUIs with MATLAB. 1996:88.
12. Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. 2004:1-28. doi:10.1023/B:VISI.0000029664.99615.94.
13. Nixon M, Aguado A. *Feature Extraction & Image Processing;* 2008.

14. Schmid C. Coloring local feature extraction. *Comput Vision--ECCV 2006*. 2006;3952:1-14. doi:10.1007/11744047\_26.
15. Vedaldi A. An implementation of SIFT detector and descriptor. *Int J*. 2008.
16. Szeliski R. Computer Vision : Algorithms and Applications. *Computer (Long Beach Calif)*. 2010;5:832. doi:10.1007/978-1-84882-935-0.
17. Νικόλαος Παπαμάρκος, Ψηφιακή επεξεργασία και ανάλυση της εικόνας
18. <http://docs.gimp.org/el/plug-in-gauss.html>
19. <https://en.wikipedia.org/wiki/SIFT>
20. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>
21. <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction>
22. [www.matlab.com](http://www.matlab.com)

## **Παράρτημα Κώδικα**

## SIFT.m

```
function varargout = SIFT(varargin)
% SIFT MATLAB code for SIFT.fig
%     SIFT, by itself, creates a new SIFT or raises the
existing
%     singleton*.
%
%     H = SIFT returns the handle to a new SIFT or the
handle to
%     the existing singleton*.
%
%     SIFT('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in SIFT.M with the given
input arguments.
%
%     SIFT('Property','Value',...) creates a new SIFT or
raises the
%     existing singleton*. Starting from the left,
property value pairs are
%     applied to the GUI before SIFT_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes
property application
%     stop. All inputs are passed to SIFT_OpeningFcn
via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose
"GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help SIFT

% Last Modified by GUIDE v2.5 23-Sep-2015 17:53:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @SIFT_OpeningFcn,
                  ...
                  'gui_OutputFcn',  @SIFT_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before SIFT is made visible.
function SIFT_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to SIFT (see
VARARGIN)

% Choose default command line output for SIFT
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes SIFT wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
command line.
function varargout = SIFT_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```



```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata,
handles)
insert_image; %καλέσµα της συνάρτησης για εισαγωγή
εικόνας
global im1 %αποθήκευση global µεταβλητής µε την
αποθηκευµένη εικόνα µε ανάλυση 480x360
set(handles.axes1,'visible','on') %Ορισµός για το που θα
εμφανιστεί η εικόνα
imshow(im1,'Parent',handles.axes1); %Εμφάνιση εικόνας που
θα επεξεργαστεί

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata,
handles)
insert_image2; %Κάλεσµα της συνάρτησης για εισαγωγή της
δεύτερης εικόνας
global im2
set(handles.axes2,'visible','on')
imshow(im2,'Parent',handles.axes2); %Εμφάνιση της
δεύτερης εικόνας που θα επεξεργαστεί

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
global im1 im2

all1; %Κάλεσµα όλων των συναρτήσεων για την εξαγωγή
χαρακτηριστικών για την πρώτη εικόνα
all2; %Κάλεσµα όλων των συναρτήσεων για την εξαγωγή
χαρακτηριστικών της δεύτερης εικόνας
img1=load('variables1.mat');
contrastable1=img1.contrastable; %απεικόνιση της πρώτης
εικόνας και των Keypoints
image1=im1;
axes(handles.axes1);
imshow(image1)
hold on
for i=2:479
    for j=2:639
        if contrastable1(i,j)~=0
            plot(j,i, 'o')
            hold on
        end
    end
end
end

img2=load('variables2.mat');

```

```

contrastable2=img2.contrastable; %Απεικόνιση της
δεύτερης εικόνας και των Keypoints
image2=im2;
axes(handles.axes2);
imshow(image2);
hold on
for i=2:479
    for j=2:639
        if contrastable2(i,j)~=0
            plot(j,i, 'o')
            hold on
        end
    end
end
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata,
handles)
finding %Κάλεσμα του κώδικα για την σύγκριση των
περιγραφέων
showmatch %Απεικόνιση ταιριασμάτων
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)

```

---

## All1.m

```

global im1 imr1
img=im1;
resimg=imr1;
scale
extrema
orientation
descriptor
save scale1.mat sigma1 sigma2 sigma3 sigma4 sigma5
gausmask1 gausmask2 gausmask3 gausmask4 gausmask5
gausimg1 gausimg2 gausimg3 gausimg4 gausimg5 halfimg
halfgausimg1 halfgausimg2 halfgausimg3 halfgausimg4
halfgausimg5 L1 L2 L3 L4 L21 L22 L23 L24
save variables1.mat contrastable Mxy thxy otable logtable
logtable2 mag contrastable descriptors positions

```

---

## All2.m

```

global im2 imr2
img=im2;
resimg=imr2;

```

```

scale
extrema
orientation
descriptor
save scale2.mat sigma1 sigma2 sigma3 sigma4 sigma5
gausmask1 gausmask2 gausmask3 gausmask4 gausmask5
gausimg1 gausimg2 gausimg3 gausimg4 gausimg5 halfimg
halfgausimg1 halfgausimg2 halfgausimg3 halfgausimg4
halfgausimg5 L1 L2 L3 L4 L21 L22 L23 L24
save variables2.mat contrastable Mxy thxy otable logtable
logtable2 mag contrastable descriptors positions

```

---

## insert\_image.m

```

function insert_image
global im1 imr1
[path,user_cancel]=imgetfile();% άνοιγει ένα browser για
να επιλέξουμε μια εικόνα.
if user_cancel
    msgbox(sprintf('Error'),'Error','Error');%σε
περίπτωση που η εικόνα δεν μπει τότε βγαίνει ένα μήνυμα
λάθους
    return
end
im1=imread(path); %Διάβασμα της εικόνας;
[a,b]=size(im1); %Παίρνουμε τις διαστάσεις της εικόνας
if (a>=b)
    im1=imresize(im1,[640 480]);
    rot=false;
else
    im1=imresize(im1,[480 640]);
    rot=true;
end

grayimg=double(rgb2gray(im1)); %μετατροπή εικόνας σε
double και σε απόχρωση του γκρι
resimg=imresize(grayimg,2);%διπλασιασμός ανάλυσης αρχικής
εικόνας (240x320)--->480x640
imr1=grayimg;
assignin('base','im1',im1); % Μεταφορά των μεταβλητών
στο workspace
assignin('base','imr1',imr1);
assignin('base','rotation1',rot);
assignin('base','sizea1',a);
assignin('base','sizeb1',b);
return
end

```

---

## insert\_image2.m

```
function insert_image2
global im2 imr2
[path,user_cancel]=imgetfile();% άνοιγει ένα browser για
να επιλέξουμε μια εικόνα.
if user_cancel
    msgbox(sprintf('Error'),'Error','Error');%σε
περίπτωση που η εικόνα δεν μπει τότε βγαίνει ένα μήνυμα
λάθους
    return
end
im2=imread(path); %Διάβασμα της εικόνας;
[a,b]=size(im2); %Παίρνουμε τις διαστάσεις της εικόνας
if (a>=b)
    im2=imresize(im2,[640 480]);
    rot=false;
else
    im2=imresize(im2,[480 640]);
    rot=true;
end
grayimg=double(rgb2gray(im2)); %μετατροπή εικόνας σε
double και σε απόχρωση του γκρι
resimg=imresize(grayimg,2);%διπλασιασμός ανάλυσης αρχικής
εικόνας (240x320)--->480x640
imr2=grayimg;
assignin('base','im2',im2);
assignin('base','imr2',imr2);
assignin('base','rotation2',rot);
assignin('base','sizea2',a);
assignin('base','sizeb2',b);
return
end
```

---

## scale.m

```
sigma1=sqrt(2)/2;%ορίζουμε το σίγμα, ρίζα δύο προς 2
gausmask1=fspecial('gaussian',7,sigma1); % A. δημιουργία
μάσκας Gauss 7x7 με σίγμα r1.
gausimg1=imfilter(resimg,gausmask1,'replicate'); %B.
εφαρμογή μάσκας πάνω στην αρχική μας εικόνα
```

```

sigma2=sqrt(2)*sigma1;% το σίγμα πολλαπλασιάζεται με το
προηγούμενο =1
gausmask2=fspecial('gaussian',7,sigma2);%A
gausimg2=imfilter(resimg,gausmask2,'replicate');%B

sigma3=sqrt(2)*sigma2;%σίγμα 1* ρίζα δύο=ρίζα 2
gausmask3=fspecial('gaussian',7,sigma3);%A
gausimg3=imfilter(resimg,gausmask3,'replicate');%B

sigma4=sqrt(2)*sigma3;% σίγμα ρίζα 2 επί ρίζα 2 = 2
gausmask4=fspecial('gaussian',7,sigma4);%A
gausimg4=imfilter(resimg,gausmask4,'replicate');%B

sigma5=sqrt(2)*sigma4;%σίγμα 2ρίζα2
gausmask5=fspecial('gaussian',7,sigma5);
gausimg5=imfilter(resimg,gausmask5,'replicate');
viewimg5=uint8(gausimg5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Δεύτερη εφαρμογή στην F3
halfimg=imresize(gausimg3,1/2); %Μείωση της ανάλυσης
halfgausimg1=imfilter(halfimg,gausmask1,'replicate');%εφα
ρμογή μάσκας με σίγμα ρίζα2 προς 2
halfgausimg2=imfilter(halfimg,gausmask2,'replicate');
halfgausimg3=imfilter(halfimg,gausmask3,'replicate');
halfgausimg4=imfilter(halfimg,gausmask4,'replicate');
halfgausimg5=imfilter(halfimg,gausmask5,'replicate');

L1=gausimg1-gausimg2;%Αφαίρεση των gaussian εικόνων για
την δημιουργία των
L2=gausimg2-gausimg3;
L3=gausimg3-gausimg4;
L4=gausimg4-gausimg5;

%Αλλάζω τις τιμές των DoG με τις τιμές των νέων εικόνων
της δεύτερηςοκτάβας%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L21=halfgausimg1-halfgausimg2;
L22=halfgausimg2-halfgausimg3;
L23=halfgausimg3-halfgausimg4;
L24=halfgausimg4-halfgausimg5;

```

---

## Extrema.m

```
a=480;
```

```

b=640;
maxpoint1=zeros(a,b); %Δημιουργία πινάκων Με τα μέγιστα,
MAX1 για την L2
maxpoint2=zeros(a,b);
minpoint1=zeros(a,b);
minpoint2=zeros(a,b);%Για την L3
maxsum1=0;%Μετρητής για όσες φορές βρεθούν τα μέγιστα και
ελάχιστα στην εικόνα για L2
maxsum2=0;%Μετρητής για L3
maxsum3=0;
maxsum4=0;
minsum1=0;
minsum2=0;
minsum3=0;
minsum4=0;
temptable=zeros(480,640); %Δήλωση προσωρινού πίνακα
extable=zeros(480,640);

for i=3:a-3
    n=i-1;
    m=i+1;
    for j=3:b-3
        s=j-1;
        x=j+1;
        L1reg=L1(n:m,s:x); %Δημιουργία πινάκων με
γειτονικά στοιχεία

        L2reg=L2(n:m,s:x);

        L3reg=L3(n:m,s:x);

        L4reg=L4(n:m,s:x);

        if L2(i,j)>=max(max(L2reg)) %Αν το μεσαίο
στοιχείο της L2 είναι το μεγαλύτερο ή μικρότερο από τα
γειτονικά του

            if L2(i,j)>=max(max(L1reg)) %Αν το
στοιχείο αυτό είναι μεγαλύτερο ή μικρότερο από το σύνολο
των στοιχείων του L1

                if L2(i,j)>=max(max(L3reg)) %Αν το
στοιχείο αυτό είναι μεγαλύτερο ή μικρότερο από το σύνολο
των στοιχείων του L3

                    maxpoint1(i,j)=255; %Τότε βάλε τιμή
255 στον πίνακα point1 όπου βρέθηκε το μέγιστο ή ελάχιστο
                    maxsum1=maxsum1 +1;
                end
            end
        end
    end
end

```

```

        if L3(i,j)>=max(max(L3reg)) %Αν το μεσαίο
στοιχείο της L3 είναι το μεγαλύτερο ή μικρότερο από τα
γειτονικά του

        if L3(i,j)>=max(max(L2reg))

            if L3(i,j)>=max(max(L4reg))

                maxpoint2(i,j)=255;
                maxsum2=maxsum2 +1;

            end

        end

    end

end

end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear L1reg L2reg L3reg
for i=2:479
    n=i-1;
    m=i+1;
    for j=2:639
        s=j-1;
        x=j+1;

            L1reg=L1(n:m,s:x); %Δημιουργία πινάκων με
γειτονικά στοιχεία

            L2reg=L2(n:m,s:x);

            L3reg=L3(n:m,s:x);

            L4reg=L4(n:m,s:x);

            if L2(i,j)<=min(min(L2reg))%Αν το μεσαίο
στοιχείο της L2 είναι το μεγαλύτερο ή μικρότερο από τα
γειτονικά του

                if L2(i,j)<=min(min(L1reg)) %Αν το
στοιχείο αυτό είναι μεγαλύτερο ή μικρότερο από το σύνολο
των στοιχείων του L1

                    if L2(i,j)<=min(min(L3reg))%Αν το
στοιχείο αυτό είναι μεγαλύτερο ή μικρότερο από το σύνολο
των στοιχείων του L3

                        minpoint1(i,j)=255; %Τότε βάλε τιμή
255 στον πίνακα point1 όπου βρέθηκε το μέγιστο ή ελάχιστο

```

```

        minsum1=minsum1 +1;
    end
end
end

    if L3(i,j)<=min(min(L3reg)) %Αν το μεσαίο
στοιχείο της L3 είναι το μεγαλύτερο ή μικρότερο από τα
γειτονικά του

        if L3(i,j)<=min(min(L2reg))

            if L3(i,j)<=min(min(L4reg))

                minpoint2(i,j)=255;
                minsum2=minsum2 +1;

            end
        end
    end
end
end

%Αφού έχουμε βρει ελάχιστα και μέγιστα τα τοποθετούμε όλα
με κριτήριο της
%αντίθεσης εικόνας σε ένα νέο πίνακα όλα μαζί

for i=1:480

    for j=1:640

        if maxpoint1(i,j)==255 %&& L2(i,j)>=9

            temptable(i,j)=L2(i,j);

        elseif minpoint1(i,j)==255; %&& L2(i,j)<=-9

            temptable(i,j)=L2(i,j);

        else

            temptable(i,j)=0;

        end
    end
end

maxsum1
maxsum2
clear L1reg L2reg L3reg

```



```

%%Εφόσον έχουν βρεθεί τα μέγιστα και τα ελάχιστα, θα
γίνει μια απαλοιφή
%κάποιων σημείων για να έχουμε ποιο ξεκάθαρο
αποτέλεσμα. Θα
%μετατρέψουμε την εικόνα σε θετικά νούμερα και θα
κρίνουμε ποια θα
%κρατήσουμε. Θα Πούμε ότι όσα είναι μικρότερα από ένα
νούμερο θα
%εξαλειφθούν. Το αποτέλεσμα θα είναι ποιο ξεκάθαρο.
contrastable=zeros(480,640);

%%Τώρα θα μπει ένα κριτήριο για να κάνουμε την απαλοιφή
των μικρότερων
%νουμέρων που δεν χρειαζόμαστε.
counter=0;

for i=1:480
    for j=1:640
        if abs(temptable(i,j))>=3
            contrastable(i,j)=temptable(i,j); % Βάζουμε
τις αρχικές τιμές που είναι και οι αρνητικές αυτή τη φορά
για να γίνει η χρήση του πίνακα μας στις παρακάτω
μετατροπές.
            counter=counter+1;
        end
    end
end

% Εμφάνιση των αποτελεσμάτων
%εμφάνιση της αρχικής μας εικόνας

```

---

## Showgas1.m

```

viewimg1=uint8(gausimg1);
viewimg2=uint8(gausimg2);
viewimg3=uint8(gausimg3);
viewimg4=uint8(gausimg4);
viewimg5=uint8(gausimg5);
fg1=figure(1); %Ανοιγμα παραθύρου
set(fg1,'name',' Gaussian Images First Octave')
subplot(3,2,1) %ορισμός της ποσότητας εικόνων που θα
εμφανήσουμε
imshow(viewimg1)
title(['σ=',num2str(sigma1)])
subplot(3,2,2)
imshow(viewimg2)
title(['σ=',num2str(sigma2)])
subplot(3,2,3)
imshow(viewimg3)

```

```

title([' $\sigma$ ', num2str(sigma3)])
subplot(3,2,4)
imshow(viewimg4)
title([' $\sigma$ ', num2str(sigma4)])
subplot(3,2,5)
imshow(viewimg5)
title([' $\sigma$ ', num2str(sigma5)])

```

---

## scale2.m

```

%%Εφαρμογή σε εικόνα μειωμένη κατά 50 για τη δημιουργία
της οκτάβας
resimg2=imresize(imr1,0.5);
sigma21=2*sigma1;
sigma22=2*sigma2;
sigma23=2*sigma3;
sigma24=2*sigma4;
sigma25=2*sigma5;
gausmask21=fspecial('gaussian',7,sigma21)
gausmask22=fspecial('gaussian',7,sigma22)
gausmask23=fspecial('gaussian',7,sigma23)
gausmask24=fspecial('gaussian',7,sigma24)
gausmask25=fspecial('gaussian',7,sigma25)
gausimg21=imfilter(resimg2,gausmask21,'replicate');
gausimg22=imfilter(resimg2,gausmask22,'replicate');
gausimg23=imfilter(resimg2,gausmask23,'replicate');
gausimg24=imfilter(resimg2,gausmask24,'replicate');
gausimg25=imfilter(resimg2,gausmask25,'replicate');
L21=gausimg21-gausimg22;
L22=gausimg22-gausimg23;
L23=gausimg23-gausimg24;
L24=gausimg24-gausimg25;
viewimg21=uint8(gausimg21);
viewimg22=uint8(gausimg22);
viewimg23=uint8(gausimg23);
viewimg24=uint8(gausimg24);
viewimg25=uint8(gausimg25);
fg1=figure(1); %Άνοιγμα παραθύρου
set(fg1,'name',' Gaussian Images Second Octave')
subplot(3,2,1) %ορισμός της ποσότητας εικόνων που θα
εμφανήσουμε
imshow(viewimg21)
title([' $\sigma$ ', num2str(sigma21)])
subplot(3,2,2)
imshow(viewimg22)
title([' $\sigma$ ', num2str(sigma22)])
subplot(3,2,3)
imshow(viewimg23)
title([' $\sigma$ ', num2str(sigma23)])
subplot(3,2,4)

```

```
imshow(viewimg24)
title(['σ=', num2str(sigma24)])
subplot(3,2,5)
imshow(viewimg25)
title(['σ=', num2str(sigma25)])
```

---

## showdog.m

```
fg3=figure(3);
set(fg3, 'name', ' Laplace Images')
subplot(2,2,1)
imshow(L1)
title('DoG1-DoG2')
subplot(2,2,2)
imshow(L3)
title('DoG2-DoG3')
subplot(2,2,3)
imshow(L3)
title('DoG3-DoG4')
subplot(2,2,4)
imshow(L4)
title('DoG4-DoG5')

fg4=figure(4);
set(fg4, 'name', 'Half Laplace Images')
subplot(2,2,1)
imshow(L21)
title('DoG1-DoG2')
subplot(2,2,2)
imshow(L22)
title('DoG2-DoG3')
subplot(2,2,3)
imshow(L23)
title('DoG3-DoG4')
subplot(2,2,4)
imshow(L24)
title('DoG4-DoG5')
```

---

## orientation.m

```
%Εφαρμογή του Orientation
Lx=zeros(480,640);
Ly=zeros(480,640);
peaktable=zeros(480,640);
testvar=zeros(7,7); %Δημιουργία ενός πίνακα 7x7 για να
μπορέσουμε να αποθηκεύσουμε τα διανύσματα της εικόνας
L=gausimg2; % Βασιζόμαστε πάνω στην εικόνα μας με σ2=1
logtable=zeros(100,2);
logtable2=zeros(100,2);
```

```

q=0;
for i=2:479
    n=i+1;
    m=i-1;
    for j=2:639
        s=j-1;
        x=j+1;
        %Για την ευρεση του Magnitude και της γωνίας
        βρίσκουμε τα
        %επιμέρους στοιχεία της Gauss εικόνας με το σίγμα 2.
        Lx(i,j)=L(n,j)-L(m,j);
        Ly(i,j)=L(i,x)- L(i,s);
    end
end
%% Εφαρμογή του τύπου για το μέγεθος των διανυσμάτων
k=1;
for i=2:479
    for j=2:639;
        Mxy(i,j)=sqrt(Lx(i,j)^2+Ly(i,j)^2);
        Mxy2(k)=Mxy(i,j);
        Gx2(k)=Lx(i,j);
        Gy2(k)=Ly(i,j);
    end
end
k=1;
% εφαρμογή του τύπου για την κατεύθυνση των διανυσμάτων
for i=2:479
    for j=2:639;
        thxy(i,j)= mod(atan2d(Ly(i,j),Lx(i,j)),360);
        thxy2(k)=thxy(i,j);
        k=k+1;
    end
end
%Με Βάση τα μέγιστα σημεία βρίσκουμε τις γειτονικές
περιοχές τους, 7x7 δηλ
%49 στοιχεία για το κάθε μέγιστο ή ελάχιστο.
testv=0;
for i=4:475
    n=3;
    m=3;
    for j=4:635;
        s=i-3;
        x=i+3;
        if contrastable(i,j)~=0
            testvar= Mxy((i-3):(i+3),(j-3):(j+3));
            testvar2=thxy((i-3):(i+3),(j-3):(j+3));
            %Δηλώνω για το Keypoint τις μεταβλητές για τη
            δημιουργία ενός
            %πίνακα που τοποθετεί το magnitude στην αντίστοιχη
            θέση γωνίας
            r=1;

```

```

otable=zeros(36,49);
testvar2=round(testvar2);
testvar2=int16(testvar2);
%Στον πίνακα που έχουμε πάρει τώρα τοποθετούμε σε ένα
καινούριο πίνακα
%36(γωνίες)χ49(συνολικά τα στοιχεία 7χ7)
for k=1:7
    for l=1:7
        if testvar2(k,l)<=10 && testvar2(k,l)>=0
            thita=1;
        else
            thita=testvar2(k,l)/10;
        end
        otable(thita,r)=testvar(k,l);
        r=r+1;
    end
end
%Δημιουργία του ιστογράμματος
counttable=zeros(1,36); %%Δημιουργία ενός
μονοδιάστατου πίνακα για να τοποθετήσουμε τα πλήθη των
χαρακτηριστικών για την κάθε γωνία
sumtable=zeros(1,36);
for k=1:36
    counting=0;
    for l=1:49
        if otable(k,l)~=0
            counting=counting+1;
            counttable(k)=counting;
            sumtable(k)=sumtable(k)+otable(k,l);
        end
    end
end
%Ευρεση του μέγιστου σημείου
maxsumag=0;
for k=1:36
    if sumtable(k)>maxsumag
        maxsumag=sumtable(k);
        maxcorner=k; %Το max sum Magnitude παίρνει την τιμή
του μέγιστου σημείου και το maxcorner παίρνει την τιμή
της γωνίας
    end
end
peaktable(i,j)=maxcorner*10;
q=q+1;
logtable(q,1)=i;
logtable(q,2)=j;
logtable2(q,1)=maxsumag;
logtable2(q,2)= maxcorner*10;
end
end
end

```

## histomag.m

```
counttable=zeros(1,36); %%Δημιουργία ενός μονοδιάστατου
πίνακα για να το ποθετήσουμε τα πλήθη των χαρακτηριστικών
για την κάθε γωνία
sumtable=zeros(1,36);
for i=1:36
    for j=1:49
        if otable(i,j)~=0
            sumtable(i)=sumtable(i)+otable(i,j);
        end
    end
end
bar(sumtable) %επίδειξη του ιστογράμματος με βάση τον
πίνακα sumtable
```

---

## Descriptor.m

```
mag=zeros(16,16); %Δημιουργούμε ένα πίνακα 16x16 με το
magnitude
angle=zeros(16,16); % και ένα πίνακα με τις γωνίες
descriptors=zeros(16,8,1000);
positions=zeros(1000,2);
count=1;
for i=10:470
    k=i-8; %δηλώνουμε γύρω από το σημείο του μέγιστου
σημείου να πάρει τα 16 γειτονικά του στοιχεία
    l=i+7;
    for j=10:630
        m=j-8;
        n=j+7;
        if abs(contrastable(i,j))>=6 % Όταν βρεθεί η
μέγιστη τιμή από τον πίνακα των μέγιστων τότε σε εκείνη
τη περιοχή παίρνουμε τα στοιχεία του πίνακα
            mag=Mxy((k:l),(m:n)); %από τον πίνακα magnitude
παίρνουμε γύρω από την περιοχή του keypoint τα 16x16
γειτονικά στοιχεία του
            angle=thxy((k:l),(m:n)); %το ίδιο παίρνουμε και
για τις γωνίες
            descriptors(:,:,count)=descript(mag,angle);
%Καλούμε την συνάρτηση όπου θα μας φτιάξει γύρω από την
περιοχή τα ιστογράμματα
            positions(count,1)=i; %αποθήκευση των θέσεων
            positions(count,2)=j;
            count=count+1;
        end
    end
end
```

```
end
end
end
```

---

## descript.m

```
function descr =descript(mag,angle) %δήλωση της
συνάρτησης εισάγεται ένας 16χ16 πίνακας
m1=0; %μεταβλητές που θα βοηθήσουν στο να χωρίσουμε 4χ4
πίνακες γύρω από
n1=0;
deschist=zeros(8,8);
histnum=1;
descr2=zeros(8,16);
g=1;
for i1=1:4 %δηλώνουμε ότι ξεκινάμε από το πρώτο κομμάτι
4χ4 του 16χ16 και στη σειρά
m2=m1+1; %όταν πάρουμε το ιστόγραμμα του πρώτου 4χ4 πάμε
στο επόμενο μέχρι να
m3=m2+3; %διαβαστεί ολόκληρο
m4=m2:m3;
for j1=1:4
n2=n1+1;
n3=n2+3;
n4=n2:n3;
binangle=angle(m4,n4);
binmag=mag(m4,n4);
r=1;
binangle2=round(binangle); %μετατρέπουμε τις γωνίες
σε απλά νούμερα για να μπορέσουμε να τις ομαδοποιήσουμε
binangle2=int16(binangle2);
for i2=1:4;
for j2=1:4;
thita2=binangle2(i2,j2); %φτιάχνουμε ένα
ιστόγραμμα με 8 bins και τοποθετούμε το εύρος των γωνιών
if ((thita2>=0) && (thita2 <= 44))
descr2(1,r)=binmag(i2,j2);
elseif ((thita2>=45) && (thita2 <=89))
descr2(2,r)=binmag(i2,j2);
elseif ((thita2>=90) && (thita2<=134))
descr2(3,r)=binmag(i2,j2);
elseif ((thita2>=135) && (thita2<=179))
descr2(4,r)=binmag(i2,j2);
elseif ((thita2>=180) && (thita2<=224))
descr2(5,r)=binmag(i2,j2);
elseif ((thita2>=225) && (thita2<=269))
descr2(6,r)=binmag(i2,j2);
elseif ((thita2>=270) && (thita2<=314))
descr2(7,r)=binmag(i2,j2);
elseif ((thita2>=315) && (thita2<=360))
descr2(8,r)=binmag(i2,j2);
```

```

        end
        r=r+1;
    end
end
sumbintable=zeros(1,8);
for i3=1:8
    counting=0;
    for j3=1:16
        if descr2(i3,j3)~=0
            counting=counting+1;
            deschist(histnum,i3)=counting;
        end
    end
    sumbintable(1,i3)=sumbintable(i3)+descr2(i3,j3);
end
end
end
histnum=histnum+1;
n1=n3;
descr(g,:)=sumbintable;
g=g+1;
end
n1=0;
m1=m3;
end
end

```

---

## finding.m

```

counter=0; %Μετρητής για το αν μοιάζει το ιστόγραμμα του
περιγραφέα ανάμεσα στις 2 εικόνες
testing=0; %Μετρητής για το πόσα κοινά σημεία υπάρχουν
newpoints1=zeros(480,640); %πίνακας για εισαγωγή τιμών
για την τελική απεικόνιση των κοινών σημείων της πρώτης
εικόνας
newpoints2=zeros(480,640); %πίνακας για εισαγωγή τιμών
για την τελική απεικόνιση των κοινών σημείων της δεύτερης
εικόνας
input1=load('variables1.mat'); %Εισάγουμε τις τιμές από
την πρώτη εικόνα
descriptors1=input1.descriptors;%αποθηκεύουμε τους
συνολικούς περιγραφείς της πρώτης εικόνας
input2=load('variables2.mat'); %Εισάγουμε τις τιμές από
την δεύτερη εικόνα
descriptors2=input2.descriptors;%αποθηκεύουμε τους
συνολικούς περιγραφείς της δεύτερης εικόνας
pos1=input1.positions; %αποθηκεύουμε και τις
συντεταγμένες των περιγραφέων
pos2=input2.positions;
coor1=zeros(6000,2); %Δημιουργία μεταβλητών για τις νέες
συντεταγμένες μετά τη σύγκριση

```



```

coor2=zeros(6000,2);
checkdescr=false;
[a,d]=size(pos1); %για τη δομή επανάληψης πέρνουμε τα
μεγέθη των διαστάσεων των πινάκων
[b,d]=size(pos2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:(a-1)
    for j=1:(b-1)

        for k=1:16
            for l=1:8

                if (descriptors1(k,l,i)~=0) &&
(descriptors2(k,l,j)~=0) %Αν οι περιγραφείς δεν είναι
μηδενικοί
                    mion(k,l)=abs(descriptors1(k,l,i)-
(descriptors2(k,l,j)));
                    checkdescr=true; %τότε μόλις τελειώσει η
συνθήκη η τιμή ας είναι αληθής
                else
                    checkdescr=false; %αλλιώς δεν ισχύει

                end
            end
        end
    end
    if checkdescr==true %εφόσον είναι αδιάφοροι του 0

        susp=abs(descriptors1(:, :, i)-
descriptors2(:, :, j)); %Τους αφαιρούμε

        sumtable=sum(sum(susp)); %Αποθηκεύουμε σύνολο
όλων των τιμών του αφαιρεμένου πίνακα
        sumtable2=sum(sum(mion));
        if sumtable<=1000 % Αν είναι πολύ χαμηλό τότε
είναι όμοιοι
            counter=counter+1;
            x1=pos1(i,1); %το x της πρώτης εικόνας
            y1=pos1(i,2); %το y της πρώτη εικόνας
            x2=pos2(j,1); %το x της δεύτερης εικόνας
            y2=pos2(j,2); %το y της δεύτερης εικόνας
            distancepix=abs(x1-x2)+abs(y1-y2); % η
απόσταση μανχάταν
            if ((distancepix<=80) && (x1~=0) &&
(y1~=0)) %Αν η τιμή της απόστασης μανχάταν είναι
μικρότερη από 5 pixel
                testimg=testimg+1;

                if (testimg==1)

```



```
ax = axes;  
showMatchedFeatures(im1,im2,coor1,coor2,'montage','Parent'  
,ax)
```

---