Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής Και Πολυμέσων

Πτυχιακή Εργασία

Τίτλος: Γρήγορος εντοπισμός αντικειμένων, υπολογισμός του πραγματικού τους μεγέθους και αναγνώριση με τη χρήση του αισθητήρα Microsoft Kinect

Μιχάλης Δημητρίου (ΑΜ:1554)

Επιβλέπων Καθηγητής: Γεώργιος Τριανταφυλλίδης

Επιτροπή Αξιολόγησης: Γεώργιος Τριανταφυλλίδης
Νικόλαος Βιδάκης,
Τσαμπίκος Κουναλάκης

Ημερομηνία Παρουσίασης: Απρίλιος 2013

ABSTRACT

This work presents an efficient and complete system for multiple object detection and classification in a 3D scene using the Microsoft Kinect sensor. It employs a new and fast detection method based on the depth map generated by the Kinect sensor and then applies the Linear Spatial Pyramid Matching [1] classification algorithm proposed by Jianchao Yang et al for the classification task. Successful 3D scene's object detection and classification are crucial features in computer vision. The main goal is making machines that see and understand objects like humans do. To this goal, the Kinect sensor can be utilized since it provides real-time depth map generation which can be used along with the RGB images for our tasks. In our system we employ effective depth map processing techniques, along with edge detection, connected components detection and filtering approaches, in order to design a complete algorithm for efficient object detection of multiple individual objects in a single scene, even in complex scenes with many objects. Besides, we use the LSPM algorithm for the efficient classification of the detected objects. This method provides many benefits over traditional object detection and classification methods; among others is the high detection rate, the accurate detection of boundaries, the real size estimation of objects and fast detection speed. Object classification when preceded by detection can provide better recognition rates, computational efficiency and multiple object classification from a single scene image.

ΣΥΝΟΨΗ

Αυτή η εργασία παρουσιάζει ένα αποδοτικό και ολοκληρωμένο σύστημα για τον εντοπισμό πολλαπλών αντικειμένων από μια τρισδιάστατη σκηνή και την κατηγοριοποίηση τους κάνοντας χρήση του αισθητήρα Microsoft Kinect. Χρησιμοποιεί μια νέα και γρήγορη μέθοδο για τον εντοπισμό των αντικειμένων που βασίζεται στους χάρτες βάθους που παράγει το ο αισθητήρας Kinect και στη συνέχεια εφαρμόζει τον αλγόριθμο ταξινόμησης Γραμμικής Χωρικής Ταύτισης Πυραμίδας (Linear Spatial Pyramid Matching[1]) που προτάθηκε από τον Jianchao Yang και τους συνεργάτες του (CVPR09)για να κάνει κατηγοριοποίηση των αντικειμένων. Η επιτυχής ανίχνευση και κατηγοριοποίηση των αντικειμένων μιας τρισδιάστατης σκηνής είναι κρίσιμος παράγοντας της Υπολογιστικής Όρασης. Ο κύριος στόχος της Υπολογιστικής Όρασης είναι να κατασκευαστούν μηχανές οι οποίες θα βλέπουν αλλά και θα κατανοούν τα αντικείμενα όπως και ο άνθρωπος. Προς αυτή την κατεύθυνση, το Kinect μπορεί να χρησιμοποιηθεί αφού έχει την δυνατότητα να παράγει σε πραγματικό χρόνο χάρτες βάθους που περιέχουν τη τρισδιάστατη πληροφορία και μαζί με τις αντίστοιχες RGBεικόνες που επιστρέφει μπορούν να χρησιμοποιηθούν για τον στόχο μας. Στο σύστημα μας χρησιμοποιούμε αποτελεσματικές μεθόδους για την επεξεργασία του χάρτη βάθους σε συνδυασμό με ανίχνευση ακμών, εντοπισμό συνδεδεμένων στοιχείων και τεχνικές φιλτραρίσματος με σκοπό την υλοποίηση ενός αλγορίθμου που μπορεί να ανιχνεύει πολλαπλά αντικείμενα από μία μόνο σκηνή, ακόμα και σε πολύπλοκες σκηνές με πολλά και αλληλεπικαλυπτόμενα αντικείμενα. Επιπλέον χρησιμοποιούμε τον αλγόριθμο LSPM για την αποδοτική κατηγοριοποίηση των αντικειμένων που εντοπίζονται. Η ανίχνευση αντικειμένων με την προτεινόμενη μέθοδο παρουσιάζει πολλά πλεονεκτήματα σε σχέση με τις παραδοσιακές μεθόδους ανάμεσα στα οποία είναι η αποτελεσματικότητα, η ακρίβεια στον εντοπισμό των ορίων, η εκτίμηση του πραγματικού μεγέθους των αντικειμένων και η μεγάλη ταχύτητα ανίχνευσης. Η κατηγοριοποίηση αντικειμένων όταν έχει προηγηθεί ανίχνευση βοηθάει στην καλύτερη αναγνώριση των αντικειμένων, καλύτερη αξιοποίηση της επεξεργαστικής ισχύος του συστήματος και κάνει δυνατή την αναγνώριση πολλαπλών αντικειμένων που προέρχονται από την ίδια εικόνα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

ΕΙΣΑΓΩΓΗ

Αυτή η εργασία έχει υλοποιηθεί στα πλαίσια της πτυχιακής εργασίας του τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων και αναλύει την εφαρμογή που έχει δημιουργηθεί για τον εντοπισμό των αντικειμένων σε μια τρισδιάστατη σκηνή, τον υπολογισμό του πραγματικού μεγέθους του κάθε αντικειμένου που εντοπίζεται και την κατηγοριοποίηση αντικειμένων που εντοπίζονται με αυτή την μέθοδο σε κλάσεις αντικειμένων με τη χρήση του αισθητήρα Kinect.

Η εφαρμογή αυτή έχει δημιουργηθεί από εμένα στο κομμάτια του εντοπισμού των αντικειμένων και του υπολογισμού μεγέθους ενώ για την αναγνώριση έχω χρησιμοποιήσει την μέθοδο Γραμμικής Χωρικής Ταύτισης Πυραμίδας[1] (LSPM) που προτείνει ο Jianchao Yang και οι συνεργάτες του στην εργασία του για το 22^{ο} διεθνές συνέδριο Υπολογιστικής Όρασης και Αναγνώρισης Προτύπων (CVPR09).

Η ιδέα για την εφαρμογή έχει βασιστεί εν μέρη σε προηγούμενη εργασία από τον Τ. Κουναλάκη και προσπαθεί να ενσωματώσει τις τελευταίες εξελίξεις στην Τεχνητή Όραση και την νέα τεχνολογία για να δημιουργήσει μια εφαρμογή ικανή να χρησιμοποιηθεί σε πραγματικές συνθήκες.

ΠΕΡΙΛΗΨΗ

Ένας από τους κυριότερους σκοπούς της τεχνητής όρασης είναι να δημιουργήσει μια μηχανή, η οποία θα μπορεί να βλέπει τα πράγματα όπως τα βλέπουν οι άνθρωποι. Η αντίληψη του ανθρώπου για την όραση όμως είναι μια πολύπλοκη διαδικασία. Ο τρόπος με τον οποίο το ανθρώπινο μυαλό αποκωδικοποιεί και φιλτράρει την πληροφορία που καταφθάνει σε αυτό και οι εσωτερικές του λειτουργίες παραμένουν άγνωστα στον άνθρωπο, παρόλο που τα τελευταία χρόνια στον τομέα αυτό σημειώνεται σημαντική πρόοδος.

Υλοποιώντας στην τεχνητή όραση όσα μάθαμε για τη βιολογική όραση μπορούμε να πετύχουμε σπουδαία πράγματα, Για παράδειγμα, γνωρίζουμε ότι η αντίληψη του βάθους στη βιολογική όραση οφείλεται στο γεγονός ότι ο άνθρωπος έχει δύο μάτια, και το φώς που φτάνει στο κάθε μάτι από μια πηγή φωτός σχηματίζει διαφορετική γωνία. Σε αντίστοιχη εργασία από τον Τ. Κουναλάκη [2] και άλλους, πάνω στην οποία βασίστηκε και η παρούσα, προσομοιώθηκε η παραπάνω διεργασία με τη χρήση μιας στερεοσκοπικής κάμερας για τη δημιουργία του χάρτη βάθους, δίνοντας με αυτό τον τρόπο σε μια δυσδιάστατη εικόνα και την πληροφορία για την τρίτη διάσταση στο χώρο. Ωστόσο ενώ το ανθρώπινο μυαλό μπορεί να επεξεργάζεται αυτού του είδους τα δεδομένα με μεγάλη ακρίβεια και φαινομενικά ακαριαία, οι αλγόριθμοι εξαγωγής χάρτη βάθους με τη χρήση στερεοσκοπικών δεδομένων απαιτούν αρκετό χρόνο σε κάθε εικόνα και καθιστούν την εφαρμογή της σε συνθήκες πραγματικού χρόνου αδύνατη.

Καθώς η τεχνολογία προοδεύει, μια νέα προσέγγιση είναι πλέον εφικτή. Αντί να χρησιμοποιούμε στερεοσκοπικές κάμερες για να πάρουμε δύο εικόνες και να υπολογίσουμε από αυτές το χάρτη βάθους, τώρα έχουμε στη διάθεση μας αισθητήρες βάθους που δημιουργούν απευθείας τον χάρτη βάθους κάνοντας ένα συνδυασμό ενός προβολέα υπέρυθρης ακτινοβολίας και μιας κάμερας σε ρυθμούς των 30fps (π.χ. Microsoft Kinect [3]). Αυτή η προσέγγιση μας δίνει επιπλέον την δυνατότητα να μετράμε την πραγματική απόσταση ανάμεσα στον αισθητήρα και το εμπόδιο που αντανακλά την υπέρυθρη ακτινοβολία. Με αυτά τα μέσα και σε συνδυασμό με τις τελευταίες τεχνικές στον τομέα της επεξεργασίας εικόνας και της τεχνητής όρασης, μπορούμε να παράγουμε πολύ καλύτερα αποτελέσματα μόνο σε ένα μικρό κλάσμα του χρόνου που χρειαζόταν παλαιότερα. Το κλειδί

στο σύστημα μας είναι ότι χρησιμοποιούμε αλγόριθμους ανίχνευσης ακμών κατευθείαν στο χάρτη βάθους για να εντοπίσουμε απότομες αλλαγές στο βάθος αντί για τη φωτεινότητα.

ΚΙΝΗΤΡΟ ΓΙΑ ΤΗ ΔΙΕΞΑΓΩΓΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Το κίνητρο για τη διεξαγωγή της εργασίας αυτής είναι η απόκτηση γνώσης και εμπειρίας από τον συγγραφέα στους τομείς της τεχνητής όρασης, της αναγνώρισης προτύπων, της επεξεργασίας εικόνας και της τεχνητής νοημοσύνης. Επίσης η ενασχόληση με μια τεχνολογία αιχμής όπως αυτή του Kinect και η ενσωμάτωση της στους παραπάνω τομείς είναι κάτι που μπορεί να συνεισφέρει στην περεταίρω εξέλιξη τους, να εμφανίσει νέες δυνατότητες εφαρμογών και να κινήσει το ενδιαφέρον και σε άλλους για έρευνα.

ΣΚΟΠΟΣ ΚΑΙ ΣΤΟΧΟΙ ΤΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της παρούσας εργασίας είναι η συνεισφορά στους τομείς της τεχνητής όρασης και της επεξεργασίας εικόνας με έμμεσο αλλά και με άμεσο τρόπο. Η χρήση του Kinect και η ενσωμάτωση του στα ερευνητικά πεδία των παραπάνω επιστημών είναι κάτι που δεν έχει ξαναγίνει και για αυτό η εφαρμογή φιλοδοξεί να συμβάλει στους παραπάνω τομείς.

Οι κύριοι στόχοι της εργασίας είναι η συνένωση διαφόρων τεχνικών και τεχνολογιών αιχμής σε ένα ολοκληρωμένο σύστημα, η διερεύνηση και η εφαρμογή πιθανών βελτιώσεων, η αναγνώριση των αδυναμιών του συστήματος και η αντικειμενική αξιολόγηση των αποτελεσμάτων με τρόπο ώστε να δίνει την δυνατότητα και σε άλλους να συνεχίσουν την παρούσα εργασία βελτιώνοντας την ή να βοηθηθούν από αυτή στο ξεκίνημα κάποιας άλλης εφαρμογής.

ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Για να είναι ποιο εύκολη η κατανόηση από τον αναγνώστη και η αξιολόγηση της εργασίας, έχει χωριστεί σε κεφάλαια. Το πρώτο κεφάλαιο ξεκινάει με μια σύντομη εισαγωγή και τελειώνοντάς θα κλείσει με μια γενική επισκόπηση στα ερευνητικά πεδία και στις τεχνολογίες που συμμετέχουν στην εργασία.

Στο Δεύτερο κεφάλαιο θα επεξηγηθούν έννοιες κλειδιά στην εργασία όπως η ψηφιακή εικόνα, οι χάρτες βάθους, η ψηφιακή επεξεργασία εικόνας, η τεχνητή όραση, κάνοντας έτσι μια εκτενή ανάλυση των μεθόδων που χρησιμοποιούνται στην εφαρμογή και των τεχνολογιών αιχμής που είναι διαθέσιμες για να υλοποιήσουν το στόχο μας. Θα αναλύσουμε την λειτουργία της συσκευής Microsoft Kinect [3]. Στο τέλος του κεφαλαίου αυτού ο αναγνώστης πρέπει να είναι σε θέση να κατανοεί τις βασικές έννοιες που χρησιμοποιούνται στην εργασία.

Στο τρίτο κεφάλαιο θα αναπτύξουμε τους αλγόριθμους ψηφιακής επεξεργασίας εικόνας, τους αλγόριθμους διαβάθμισης εικόνας, την ανίχνευση ακμών, την βελτιστοποίηση ακμών, αλγόριθμούς ανίχνευσης αντικειμένων, τεχνικές φιλτραρίσματος, τεχνικές εξαγωγής χαρακτηριστικών και τέλος τους αλγόριθμους αναγνώρισης αντικειμένων. Θα δείξουμε γιατί επιλέξαμε τις συγκεκριμένες μεθόδους και τεχνολογίες για την υλοποίηση και πώς μπορούμε να τις συνδυάσουμε για να πετύχουμε το επιθυμητό αποτέλεσμα.

Στο τέταρτο κεφάλαιο θα αναλύσουμε την εφαρμογή. Το κεφάλαιο θα χωρίζεται σε δύο βασικά μέρη, το πρώτο θα ασχολείται με την ανίχνευση των αντικειμένων, τον διαχωρισμό τους από τη σκηνή και τον υπολογισμό του πραγματικού τους μεγέθους. Το δεύτερο μέρος θα ασχολείται με την εξαγωγή χαρακτηριστικών, την αναγνώριση και την κατηγοριοποίηση των αντικειμένων.

Τέλος θα δείξουμε στο πέμπτο κεφάλαιο τα πειραματικά αποτελέσματα της ανίχνευσης και της κατηγοριοποίησης και θα αναφέρουμε κάποια συμπεράσματα και σκέψεις για μελλοντική εργασία.

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

ΨΗΦΙΑΚΗ ΕΙΚΟΝΑ

Εικόνα θεωρείται ως η κατανομή της πληροφορίας (βαθμός αμαύρωσης ή χρώμα) στο επίπεδο (x,y). Η εικόνα είναι ουσιαστικά μια πηγή πληροφορίας. Κάθε εικόνα για να υποστεί ψηφιακή επεξεργασία θα πρέπει κατ' αρχή να μετατραπεί σε ψηφιακή. Η ψηφιακή εικόνα αποτελείται από εικονοστοιχεία, τα pixels. Στην ασπρόμαυρη εικόνα κάθε ένα εικονοστοιχείο είναι ένα δείγμα από τη συνάρτηση f(x,y) που αντιστοιχεί στην αναλογική εικόνα. Τα λευκά εικονοστοιχεία αντιστοιχούν στο 255 ενώ τα μαύρα στο 0. Η ψηφιακή εικόνα παριστάνεται μαθηματικά ως η συνάρτηση $f \rightarrow f_q$ (n_1, n_2), όπου τα n1 και n2 αντιστοιχούν στις διακριτές χωρικές μεταβολές x και y. Στην περίπτωση που η εικόνα είναι έγχρωμη τότε σε κάθε θέση (n1,n2) η f έχει τρεις τιμές των χρωμάτων κόκκινο, πράσινο και μπλε.

Ο προσανατολισμός των συντεταγμένων σε μία ψηφιακή εικόνα μπορεί να είναι αυθαίρετος. Στις περισσότερες όμως περιπτώσεις, η αρχή των συντεταγμένων (n1,n2)=(1,1) λαμβάνεται στην επάνω αριστερή γωνία της εικόνας. Κάθε ένα εικονοστοιχείο έχει τις δικές του συντεταγμένες. Η ψηφιακή εικόνα [4], μπορεί να ληφθεί είτε από αναλογικές εικόνες ή απευθείας από συστήματα λήψης ψηφιακών εικόνων. Τα συστήματα λήψης ψηφιακής εικόνας παρέχουν τη δυνατότητα στο χρήστη να λάβει τη ψηφιακή εικόνα με διαφορετικές αναλύσεις. Έτσι του δίνεται η δυνατότητα να δοκιμάσει το ανεκτό όριο τόσο στην ανάλυση όσο και στο πλήθος των διαβαθμίσεων της αμαύρωσης. Η ποιότητα της εικόνας υποβαθμίζεται με την ελάττωση της ανάλυσης αλλά ταυτόχρονα μειώνεται και η απαιτούμενη μνήμη για αποθήκευσή της.

RGB ΕΙΚΟΝΑ

Το πρότυπο χρώματος RGB [4] είναι ένα προσθετικό πρότυπο στο οποίο τα χρώματα κόκκινο, πράσινο και μπλε (χρώματα που χρησιμοποιούνται συχνά σε προσθετικά χρωματικά πρότυπα) συνδυάζονται με διάφορους τρόπους για να αναπαραχθούν άλλα χρώματα. Το όνομα του προτύπου και η σύντμηση RGB προέρχονται από τα τρία βασικά χρώματα, το κόκκινο (Red), πράσινο (Green), και το μπλε (Blue). Αυτά τα τρία χρώματα δεν πρέπει να συγχύζονται με τα τρία ανακλαστικά χρώματα κόκκινο, μπλε, και κίτρινο, τα οποία αναφέρονται στον χώρο των τεχνών ως βασικά χρώματα. Η επιλογή των βασικών χρωμάτων προήλθε πιθανώς από την ανθρώπινη βιολογία επειδή είναι ερεθίσματα τα οποία διεγείρουν συγκεκριμένους δέκτες του ανθρώπινου αμφιβληστροειδούς.

Το ανθρώπινο μάτι έχει 3 τέτοιους δέκτες (κωνία), και ο κάθε ένας είναι ευαίσθητος σε συγκεκριμένη περιοχή μήκους κύματος, αλλά είναι γενικά πιο ευαίσθητα στο πράσινο φως.

Ένα χρώμα στο πρότυπο χρώματος RGB μπορεί να περιγραφεί με το προσδιορισμό του πόσο κάθε ένα από το κόκκινο, πράσινο και μπλε χρώματα συμπεριλαμβάνεται. Κάθε ένα μπορεί να ποικίλει μεταξύ του ελάχιστου (καθόλου χρώμα) και του μεγίστου (πλήρης ένταση). Εάν όλα τα χρώματα είναι στο ελάχιστο το αποτέλεσμα είναι μαύρο. Εάν όλα τα χρώματα είναι στο μέγιστο, το αποτέλεσμα είναι το λευκό. Τα χρώματα μπορούν να περιγραφούν ποσοτικά με διάφορους τρόπους:

- Οι επιστήμονες του χρώματος συχνά τοποθετούν τα χρώματα στην κλίμακα 0 (ελάχιστο) έως 1 (μέγιστο). Πολλοί μαθηματικοί τύποι που σχετίζονται με το χρώμα χρησιμοποιούν αυτές τις τιμές. Π.χ. το μέγιστο κόκκινο είναι 1,0,0 για Κόκκινο, Πράσινο, Μπλε.

- Οι τιμές χρώματος μπορούν να γραφτούν επίσης ως ποσοστά, από 0% (ελάχιστο) ως 100% (μέγιστο). Το μέγιστο κόκκινο είναι 100%, 0%, 0%.

- Οι τιμές χρώματος μπορούν να γραφτούν ως αριθμοί στην κλίμακα 0 έως 255, απλά με τον πολλαπλασιασμό της κλίμακας 0.0 έως 1.0 με 255. Αυτό το μοντέλο απαντάται συνήθως στην πληροφορική, όπου οι προγραμματιστές προτιμούν να αποθηκεύουν κάθε αξία χρώματος σε ένα byte (8 bit). Αυτή η σύμβαση έχει γίνει τόσο διαδεδομένη ώστε πολλοί συγγραφείς την θεωρούν αυτονόητη και δεν παρέχουν το σωστό υπόβαθρο αναφοράς. Το μέγιστο κόκκινο είναι το 255,0,0.

- Η ίδια σειρά, 0 έως 255, γράφεται μερικές φορές σε δεκαεξαδικό, και ίσως με ένα πρόθεμα (π.χ. #). Επειδή οι δεκαεξαδικοί αριθμοί σε αυτήν την κλίμακα μπορούν να γραφτούν με ένα σταθερό σχήμα δύο ψηφίων, το μέγιστο κόκκινο #FF, #00, #00 μπορεί να γραφτεί και σαν #ff0000. Αυτή η σύμβαση χρησιμοποιείται στα χρώματα στο διαδίκτυο και θεωρείται επίσης από μερικούς συγγραφείς αυτονόητη.

ΧΑΡΤΗΣ ΒΑΘΟΥΣ

Ο απλούστερος τρόπος αναπαράστασης και αποθήκευσης τρισδιάστατων συντεταγμένων της επιφάνειας ενός αντικειμένου είναι με τη χρήση χαρτών βάθους [5] (depth maps).Ο χάρτης βάθους είναι μια εικόνα όπου κάθε εικονοστοιχείο, εκφράζει αντί για φωτεινότητα την απόσταση του αντίστοιχου σημείου από το επίπεδο της κάμερας (focal plane).

Εικόνα 1: Παράδειγμα εικόνας βάθους.

Ο χάρτης βάθους (Εικόνα 1) είναι μια δισδιάστατη εικόνα βεληνεκούς, όπου κάθε εικονοστοιχείο της παίρνει μια χρωματική τιμή από τις διαβαθμίσεις του γκρι. Η χρωματική τιμή υποδηλώνει την απόσταση του σημείου από τον οπτικό αισθητήρα στον τρισδιάστατο χώρο. Το πλεονέκτημα των χαρτών βάθους συνίσταται στο γεγονός ότι αναπαριστούν το τρισδιάστατο σχήμα ενός αντικειμένου που είναι αμετάβλητο στις αλλαγές των χρωματικών και ανακλαστικών ιδιοτήτων των αντικειμένων. Οι χάρτες βάθους ή πίνακες βάθους σχηματίζονται με την χρήση αισθητήρων βάθους (όπως του Kinect) που υπολογίζουν την

απόσταση ενός αντικειμένου από μια κάμερα δίνοντας έτσι διαστάσεις απόστασης από τον αισθητήρα και δημιουργώντας μια τρισδιάστατη εικόνα.

KINECT

Το Kinect [3] είναι μια συσκευή φυσικής διεπαφής χρήσης-μηχανής (NUI devise) από Microsoft για την παιχνιδομηχανή Xbox 360 και τα Windows PCs. Είναι στην ουσία μια περιφερειακή συσκευή εισόδου που διαθέτει φακούς-κάμερες και ηχητικούς αισθητήρες ώστε να επιτρέπει στους χρήστες να ελέγχουν και να αλληλεπιδρούν με τις συσκευές εξ αποστάσεως χωρίς να είναι απαραίτητη η επαφή τους με κάποιο άλλο χειριστήριο, ενσύρματο ή ασύρματο. Η αλληλεπίδραση πραγματοποιείται χρησιμοποιώντας χειρονομίες ή προφορικές εντολές μέσω του φυσικού περιβάλλοντος του χρήστη.

Εικόνα 2: Ο αισθητήρας Microsoft Kinect

Το Kinect βασίζεται σε τεχνολογίες λογισμικού που αναπτύσσονται από τη Rare, θυγατρική εταιρεία της Microsoft Game Studios, και την τεχνολογία της ισραηλινής Prime Sense[6] πάνω στις κάμερες και τις συσκευές λήψης βίντεο. Η Prime Sense ανέπτυξε ένα σύστημα που μπορεί να αντιληφθεί τις ανθρώπινες χειρονομίες. Καθιστά δυνατό τον έλεγχο της συσκευής χρησιμοποιώντας μια υπέρυθρη κάμερα, δύο φακούς και ένα ειδικό μικροτσίπ για την παρακολούθηση της κίνησης των αντικειμένων και των ατόμων σε τρεις διαστάσεις. Αυτό το σύστημα τρισδιάστατης σάρωσης, που ονομάζεται «Light Coding», χρησιμοποιεί μια παραλλαγή τρισδιάστατης εικόνας η οποία μπορεί να αναπαρασταθεί στον τρισδιάστατο χώρο.

Κατασκευαστικά, ο αισθητήρας Kinect είναι μια συσκευή που αποτελείται από μια οριζόντια διάταξη που συνδέεται με μια μικρή βάση μέσω ενός βραχύ μηχανοκίνητου άξονα και έχει σχεδιαστεί για να τοποθετείται κατά μήκος πάνω ή κάτω από την οθόνη που προβάλει το λογισμικό που το χρησιμοποιεί. Η συσκευή διαθέτει έναν RGB φακό, έναν αισθητήρα βάθους που είναι συνδυασμός δύο φακών και ένα πολλαπλό μικρόφωνο και λειτουργεί με το λογισμικό που περιλαμβάνει το Microsoft Kinect SDK [7] και παρέχει πλήρη τρισδιάστατη καταγραφή της κίνησης, αναγνώριση προσώπων και δυνατότητες αναγνώρισης φωνής. Ο αισθητήρας βάθους αποτελείται από έναν υπέρυθρο προβολέα υπέρυθρων σε συνδυασμό με ένα αισθητήρα CMOS, ο οποίος καταγράφει δεδομένα τρισδιάστατου βίντεο κάτω από οποιεσδήποτε συνθήκες φωτισμού.

Το προσωπικό της Microsoft αναφέρει ως κύρια καινοτομία του Kinect την τεχνολογία του λογισμικού του που επιτρέπει την προηγμένη αναγνώριση χειρονομιών, αναγνώριση προσώπων και αναγνώριση φωνής. Σύμφωνα με τις πληροφορίες που παρέχονται, το Kinect είναι σε θέση να εντοπίζει ταυτόχρονα έως έξι άτομα, μεταξύ των οποίων δύο ενεργούς παίκτες των οποίων αναλύει την κίνηση με δυνατότητα αντίληψης 20 αρθρώσεων ανά παίκτη.

Ο αισθητήρας Kinect εξάγει βίντεο με ρυθμό καρέ 30 Hz. Η ροή βίντεο χρησιμοποιεί το σύστημα χρωμάτων RGB 8-bit και ανάλυση 640 × 480 pixel με χρωματικό φίλτρο Bayer, ενώ η μονόχρωμη αισθητήρια ροή βίντεο χρησιμοποιεί ανάλυση 640 × 480 pixel με βάθος 11-bit, το οποίο παρέχει 2.048 επίπεδα ευαισθησίας. Ο αισθητήρας Kinect έχει ένα πρακτικό εύρος απόστασης ανίχνευσης που κυμαίνεται από 1,2 έως 3,5 μέτρα. Η περιοχή που καλύπτει το οπτικό πεδίο του Kinect είναι περίπου 6 τ.μ., αν και ο αισθητήρας μπορεί να διατηρήσει την εστίαση παρακολούθησης σε ένα διευρυμένο φάσμα περίπου 0,7 έως 6 μέτρων. Ο αισθητήρας έχει οπτικό πεδίο 57 ° οριζόντια και 43 ° κατακόρυφα, ενώ ο κινητήριος κεντρικός άξονας είναι σε θέση να κινηθεί σε εύρος γωνίας έως και 27 ° προς τα πάνω ή προς τα κάτω. Το οριζόντιο πεδίο του αισθητήρα Kinect στην ελάχιστη απόσταση θέασης του (περίπου 0,8 μέτρα) είναι περίπου 87 εκατοστά, και το κατακόρυφο περίπου 63 εκατοστά, δηλαδή αντιστοιχούν περίπου 1,3 χιλιοστά ανά ψηφίδα (pixel). Το πολλαπλό μικρόφωνο διαθέτει τέσσερις μικροφωνικές συσκευές και καθεμία λειτουργεί με κανάλι των 16-bit ήχου με ρυθμό δειγματοληψίας 16 kHz.

Επειδή ο αισθητήρας Kinect έχει μηχανοκίνητο μηχανισμό ανάκλησης απαιτεί περισσότερη ενέργεια από αυτή που μπορεί να του παρέχει μια USB θύρα. Για το λόγο αυτό η συσκευή κάνει χρήση ενός ειδικού καλωδίου τροφοδοσίας (περιλαμβάνεται με τον αισθητήρα), το οποίο χωρίζει τη σύνδεση σε ξεχωριστές συνδέσεις USB και ισχύος. Ισχύς παρέχεται από το ηλεκτρικό δίκτυο μέσω ενός μετασχηματιστή.

ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Η ψηφιακή επεξεργασία εικόνας[8][9] ασχολείται με την καταγραφή και επεξεργασία εικόνων με την βοήθεια υπολογιστή. Χρησιμοποιείται για την βελτίωση της ποιότητας μιας εικόνας, για την αποκατάσταση της ή και για την αφαίρεση θορύβου (πχ. Salt and pepper). Μπορεί επίσης να χρησιμοποιηθεί για τη συμπίεση και αποθήκευση μιας εικόνας. Η ανάλυση εικόνας παρέχει τη δυνατότητα περιγραφής του περιεχομένου μιας εικόνας αλλά και της αναγνώρισης του περιεχομένου όπου αυτό δεν είναι ξεκάθαρο. Η βασική διαφορά της επεξεργασίας εικόνας από την υπολογιστική όραση είναι ότι η επεξεργασία εικόνας έχει συγκεκριμένο πρόβλημα να λύσει σε γενικά καλά καθορισμένες συνθήκες ενώ η υπολογιστική όραση προσπαθεί να μιμηθεί την ανθρώπινη όραση/ανθρώπινο εγκέφαλο στην κατανόηση εικόνας, ένα σχετικά δύσκολο πρόβλημα με όχι καλά καθορισμένες συνθήκες.

Η ψηφιακή επεξεργασία εικόνας αντιμετωπίζει την ψηφιοποίηση και κωδικοποίηση της εικόνας για αποδοτικότερη μετάδοση και αποθήκευση της. Επίσης φροντίζει για την βελτιστοποίηση και την αποκατάσταση των εικόνων για καλύτερη απεικόνιση και κατανόηση τους. Μέσα από την ψηφιακή επεξεργασία εικόνας είναι δυνατή επίσης η τμηματοποίηση και περιγραφή μιας εικόνας καθώς επίσης και η δυνατότητα ανάλυσης και κατανόησης της. Γίνεται επίσης αξιοποίηση μέσων και μπορεί να ταυτιστεί με προσπάθειες για προσέγγιση της ανθρώπινης όρασης. Ταυτίζεται έτσι με πεδία όπως η τεχνητή όραση, η αναγνώριση προτύπων και η τεχνητή νοημοσύνη.

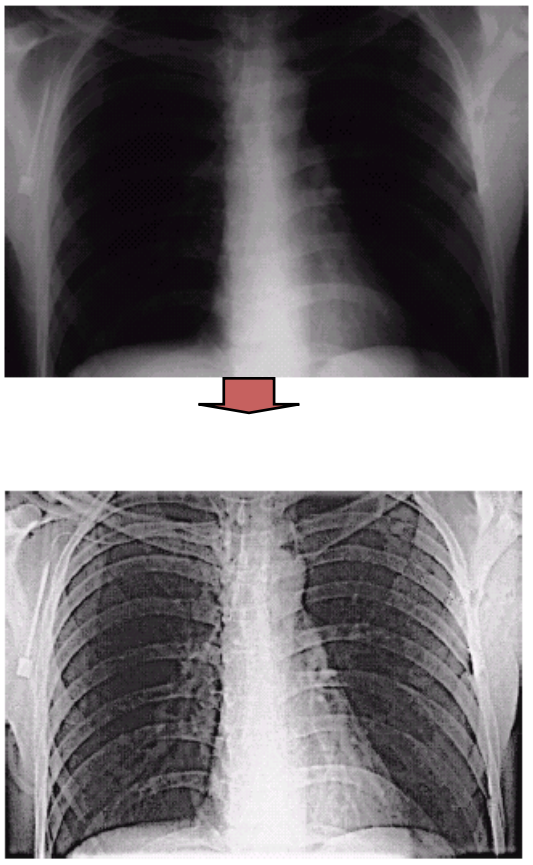
ΙΣΤΟΡΙΑ ΤΗΣ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ

Μια από τις πρώτες εφαρμογές χρήσης της ψηφιακής εικόνας ήταν στον χώρο της βιομηχανίας έκδοσης εφημερίδων. Το 1921 η χρήση ενός συστήματος μετάδοσης εικόνων μέσω ενός υποβρύχιου καλωδίου που δημιουργήθηκε από τον Bartlane [10] μείωνε δραματικά τον χρόνο που χρειαζόταν για να μεταφερθεί μια εικόνα από την μια στην άλλη άκρη του Ατλαντικού από μια εβδομάδα σε λιγότερο από τρεις ώρες. Ειδικός εκτυπωτικός

εξοπλισμός κωδικοποιούσε εικόνες για την καλωδιακή μετάδοση και μετά τα αποκωδικοποιούσε στην άλλη πλευρά. Στα μέσα της δεκαετίας του 1920 έγιναν σημαντικές βελτιώσεις στο σύστημα του Bartlane βασισμένες σε μεθόδους που χρησιμοποιούνταν στην εκτύπωση φωτογραφιών. Όμως η πραγματική πρόοδος στο χώρο της ψηφιακής επεξεργασίας εικόνων έγινε την δεκαετία του 1960 με τα άλματα που έγιναν στο χώρο της τεχνολογίας υπολογιστών αλλά και με την κούρσα για το ταξίδι στο διάστημα. Τα νέα υπολογιστικά συστήματα ήταν αρκετά εξελιγμένα για να έχουν την δυνατότητα να επεξεργαστούν τα δεδομένα που λαμβάνονταν. Η ψηφιακή επεξεργασία εικόνων αξιοποιήθηκε για την βελτίωση εικόνων που πάρθηκαν από την Σελήνη.

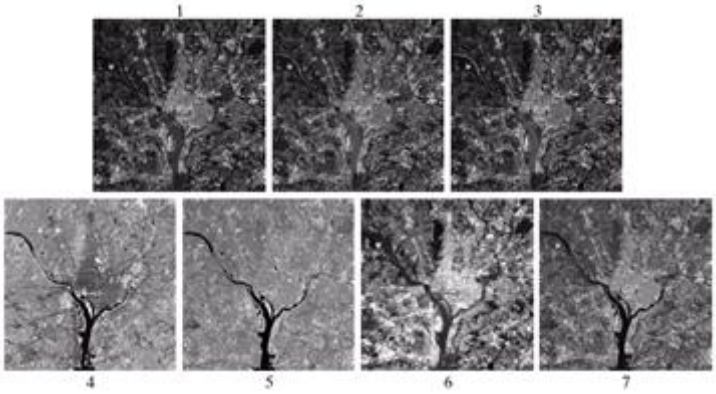
Παράλληλα με την χρήση της ψηφιακής επεξεργασίας εικόνων για την διαστημική τεχνολογία, έγινε και χρήση της τεχνολογίας αυτής στη δεκαετία του 1970 σε εφαρμογές της ιατρικής όπως η αξονική τομογραφία (CAT) που θεωρείται ένα από τα μεγαλύτερα ιατρικά επιτεύγματα στον διαγνωστικό τομέα της ιατρικής. Από το 1980 μέχρι και σήμερα η χρήση τεχνικών ψηφιακής επεξεργασίας εικόνων έχει κάνει τεράστια άλματα και χρησιμοποιούνται σήμερα σε πάρα πολλούς τομείς όπως η αποκατάσταση και βελτίωση εικόνων, για καλλιτεχνικά αποτελέσματα, στην ιατρική απεικόνιση, στην άσκηση ελέγχων στη βιομηχανία.

Μια από τις πιο συχνές εφαρμογές της ψηφιακής επεξεργασίας εικόνας είναι για να βελτιώσει την ποιότητα και να αφαιρείται ο θόρυβος από μια εικόνα κάνοντας την έτσι πιο καθαρή.

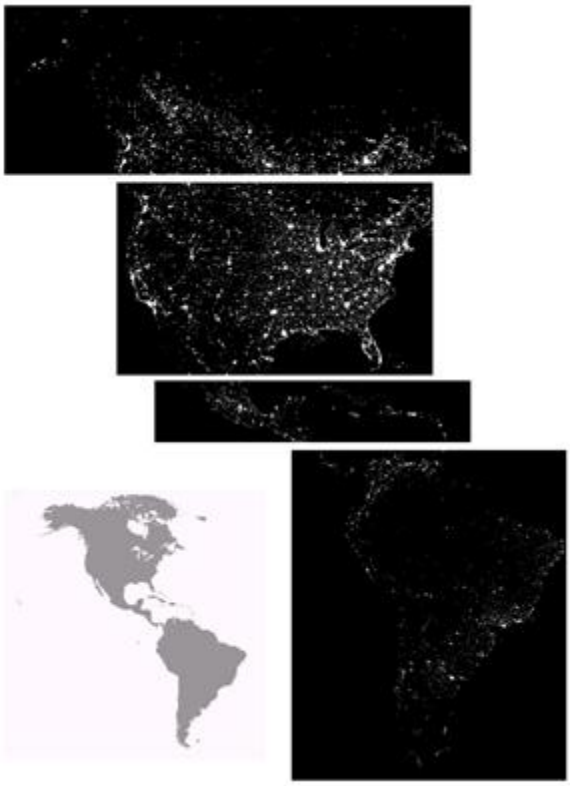


Εικόνα 3: βελτίωση της ποιότητας δίνει πιο καθαρό αποτέλεσμα για καλύτερη διάγνωση.

Μια άλλη χρήση της ψηφιακής επεξεργασίας εικόνας είναι στα συστήματα γεωγραφικών πληροφοριών όπου τεχνικές της ψηφιακής επεξεργασίας εικόνας χρησιμοποιούνται για να επεξεργαστούν εικόνες που λαμβάνονται από δορυφόρους για λήψη τοπογραφικών και μετεωρολογικών δεδομένων που δεν είναι καθαρά.



Εικόνα 4: Φωτογραφίες της γης από δορυφόρο για τοπογραφικά δεδομένα.

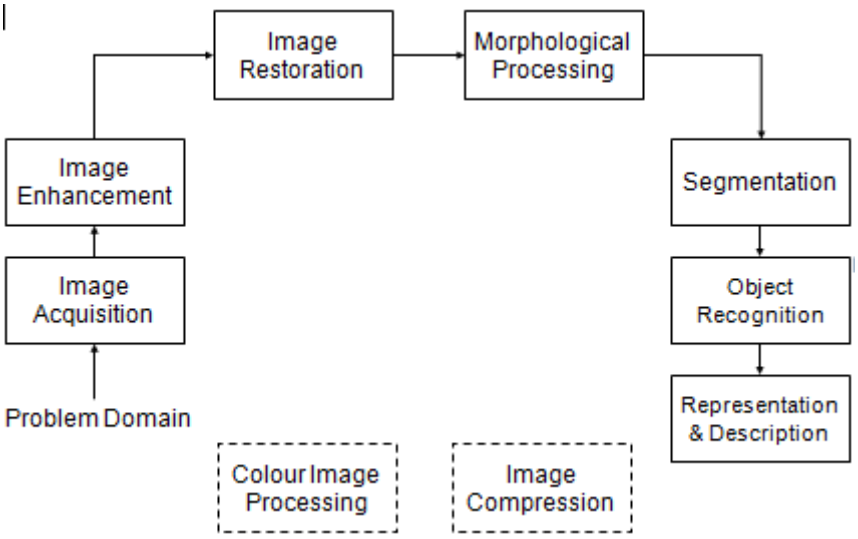


Εικόνα 5: Εικόνα από το world Data set [11] που δείχνει φωτογραφίες της γης την νύχτα.

Η πιο πάνω εικόνα δείχνει φωτογραφίες από το World Data set [11] (Night time lights) δείχνοντας έτσι τα όρια του τεχνολογικά επηρεασμένου ανθρώπινου κόσμου, και φανερώνοντας τις δυνατότητες ανάλυσης τέτοιων δεδομένων. Η ψηφιακή επεξεργασία εικόνας χρησιμοποιείται σε μεγάλο βαθμό και από τα όργανα εφαρμογής του νόμου για τη διαλεύκανση υποθέσεων μέσα από την επεξεργασία τεκμηρίων (ανάλυση δεδομένων από κάμερες, αναγνώριση δακτυλικών αποτυπωμάτων κ.α).

Εικόνα 6: Χρήση της ψηφιακής επεξεργασίας εικόνας για την μεγέθυνση πινακίδων αυτοκινήτου στη διαδικασία ανάλυσης τεκμηρίων από την αστυνομία.

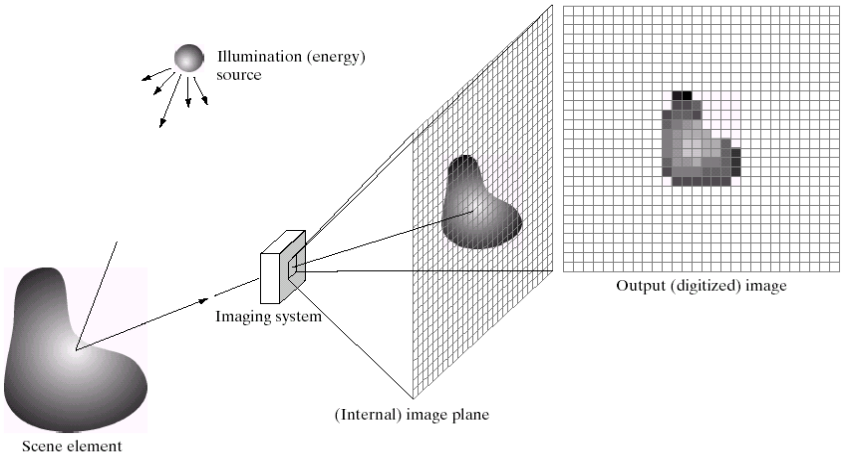
ΣΤΆΔΙΑ ΨΗΦΙΑΚΉΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΌΝΑΣ



Εικόνα 7: Τμήματα της ψηφιακής επεξεργασίας εικόνας [9]

ΚΤΗΣΗ ΕΙΚΟΝΑΣ (IMAGE ACQUISITION)

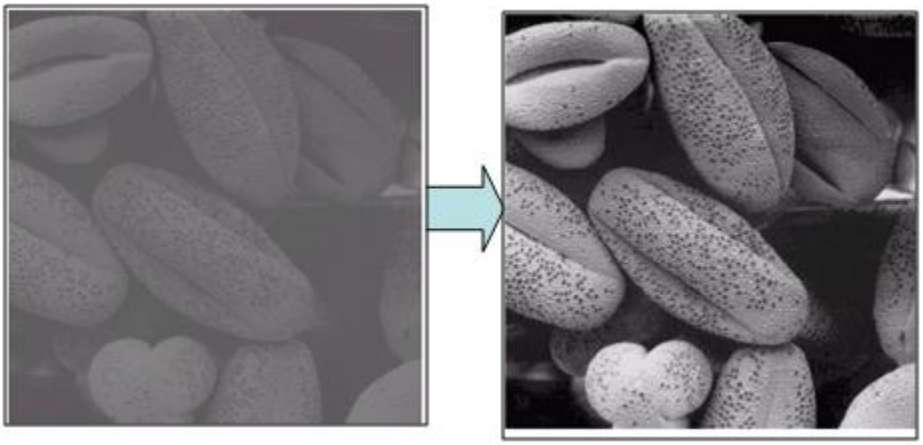
Η εικόνα που πρόκειται να τύχει ψηφιακής επεξεργασίας μπορεί να είναι δημιουργημένη σε υπολογιστή ή ψηφιακές εικόνες που προορίζονται να δημιουργήσουν μια ρεαλιστική αναπαράσταση των φυσικών στοιχείων. Μπορούν επίσης να δημιουργηθούν εικόνες από τον πραγματικό κόσμο, μετατρέποντας μια εικόνα σε ψηφιακή πληροφορία. Η μέθοδος ψηφιοποίησης μιας εικόνας καθορίζεται από την αρχική μορφή της εικόνας και τέτοιες εικόνες μπορούμε να πάρουμε από ψηφιακές φωτογραφικές μηχανές, από ένα σαρωτή (scanner) ή από δορυφορική μετάδοση κάποιων δεδομένων.



Εικόνα 8: Διαδικασία κτήσης εικόνας (image acquisition)

ΒΕΛΤΙΩΣΗ ΕΙΚΟΝΑΣ (IMAGE ENHANCEMENT)

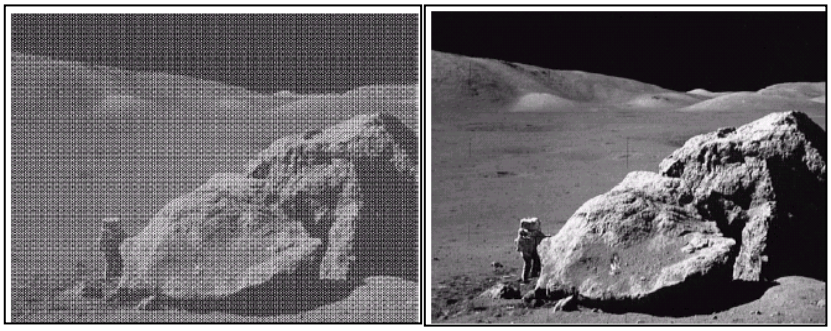
Βελτίωση εικόνων[9] είναι η διαδικασία της προσαρμογής ψηφιακών εικόνων, έτσι ώστε τα αποτελέσματα να είναι σε καλύτερη κατάσταση για επίδειξη ή για περαιτέρω ανάλυση. Μπορεί να αφαιρεθεί ο θόρυβος ή να ενισχυθεί ο φωτισμός, κάτι που καθιστά ευκολότερο να προσδιοριστούν τα βασικά χαρακτηριστικά της.



Εικόνα 9: Βελτίωση εικόνας

ΑΠΟΚΑΤΑΣΤΑΣΗ ΕΙΚΟΝΑΣ (IMAGE RESTORATION)

Ο σκοπός της αποκατάστασης εικόνας [9][4] είναι να αντισταθμίσει ή να σβήσει ελαττώματα τα οποία υποβαθμίζουν την εικόνα. Η υποβάθμιση έρχεται σε πολλές μορφές, όπως θαμπάδα λόγω κίνησης (motion blur), θόρυβος, και όχι καλή εστίαση της κάμερας. Σε περιπτώσεις όπως η θαμπάδα λόγω κίνησης, μπορούμε να φτάσουμε σε μια πολύ καλή εκτίμηση της πραγματικής λειτουργίας θολώματος και να αντιστρέψουμε τη θολούρα επαναφέροντας έτσι την αρχική εικόνα. Σε περιπτώσεις όπου η εικόνα είναι κατεστραμμένη από το θόρυβο, το καλύτερο που μπορούμε να κάνουμε είναι να αντισταθμίσουμε την υποβάθμιση που προκλήθηκε.

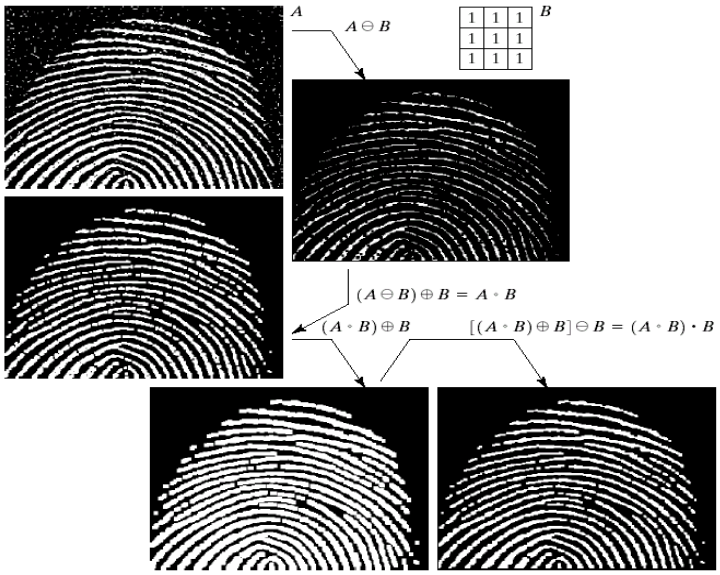


Εικόνα 10: παράδειγμα αποκατάστασης εικόνας

ΜΟΡΦΟΛΟΓΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Μορφολογική επεξεργασία εικόνας [12] είναι μια συλλογή από μη-γραμμικές λειτουργίες που σχετίζονται με το σχήμα ή τη μορφολογία των χαρακτηριστικών σε μια εικόνα. Οι μορφολογικές λειτουργίες βασίζονται μόνο στη σχετική θέση των τιμών pixel, όχι στις αριθμητικές τιμές τους, και ως εκ τούτου είναι ιδιαίτερα κατάλληλες για την επεξεργασία των δυαδικών εικόνων. Μορφολογικές λειτουργίες μπορούν επίσης να εφαρμοστούν σε εικόνες στην κλίμακα του γκρίζου όπου οι συναρτήσεις μεταφοράς του φωτός είναι άγνωστες και έτσι οι απόλυτες τιμές των pixel είναι ουσιαστικά αχρείαστες.

Οι μορφολογικές τεχνικές εξετάζουν μια εικόνα με ένα μικρό σχήμα ή πρότυπο που ονομάζεται δομικό στοιχείο. Το δομικό στοιχείο είναι τοποθετημένο σε όλες τις πιθανές θέσεις της εικόνας και έρχεται σε σύγκριση με την αντίστοιχη περιοχή των pixels. Μερικές λειτουργίες δοκιμάζουν εάν το δομικό στοιχείο εφαρμόζει στην περιοχή ενώ άλλες ελέγχουν αν χτυπά ή τέμνει την περιοχή. Μερικές από τις βασικότερες μορφολογικές λειτουργίες είναι η συστολή (erosion) και η διαστολή (dilation).



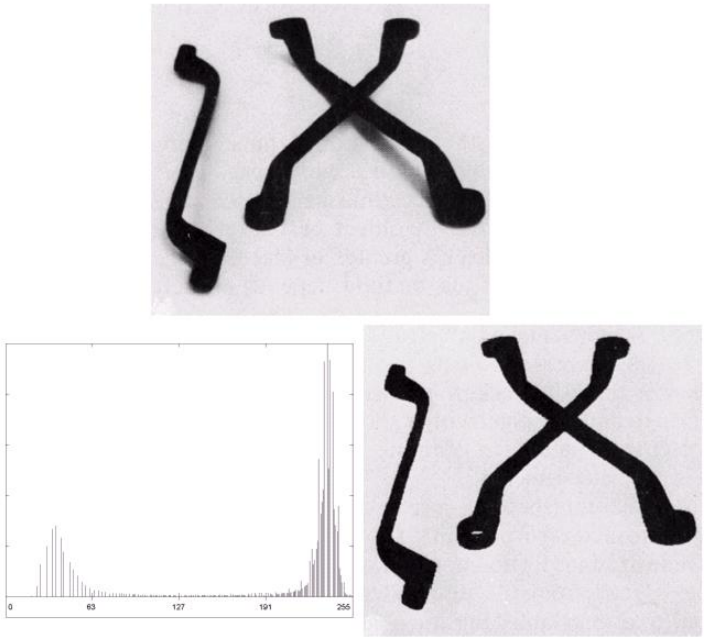
Εικόνα 11: Μορφολογική επεξεργασία εικόνας.

ΚΑΤΑΤΜΗΣΗ ΕΙΚΟΝΑΣ (IMAGE SEGMENTATION)

Στην υπολογιστική όραση κατάτμηση εικόνας [13] είναι η διαδικασία διαχωρισμού μιας ψηφιακής εικόνας σε πολλαπλά τμήματα (σετ από pixels, επίσης γνωστά ως superpixels). Ο στόχος της κατάτμησης είναι η απλούστευση ή αλλαγή της αναπαράστασης μιας εικόνας σε κάτι που είναι πιο ουσιαστικό και πιο εύκολο να αναλυθεί. Η κατάτμηση

χρησιμοποιείται συνήθως για να εντοπιστούν τα αντικείμενα και τα όρια (γραμμές, καμπύλες, κ.λπ.) σε εικόνες . Πιο συγκεκριμένα, κατάτμηση εικόνας είναι η διαδικασία ανάθεσης μιας ετικέτας σε κάθε pixel σε μια εικόνα έτσι ώστε pixels με τα ίδια ετικέτα να μοιράζονται ορισμένα οπτικά χαρακτηριστικά.

Το αποτέλεσμα της κατάτμησης εικόνας είναι ένα σύνολο από τμήματα που καλύπτουν συλλογικά ολόκληρη την εικόνα, ή μια σειρά από καμπύλες που προέρχονται από την εικόνα. Κάθε ένα από τα εικονοστοιχεία σε μία περιοχή είναι παρόμοια σε σχέση με κάποιο χαρακτηριστικό ή ιδιότητα, όπως το χρώμα, την ένταση ή την υφή. Γειτονικές περιοχές μπορεί να είναι σημαντικά διαφορετικές σε σχέση με το ίδιο χαρακτηριστικό. Η απλούστερη μέθοδος κατάτμησης εικόνων είναι η μέθοδος κατωφλίωσης. Η μέθοδος αυτή μετατρέπει μια εικόνα της κλίμακας του γκρίζου σε δυαδική.

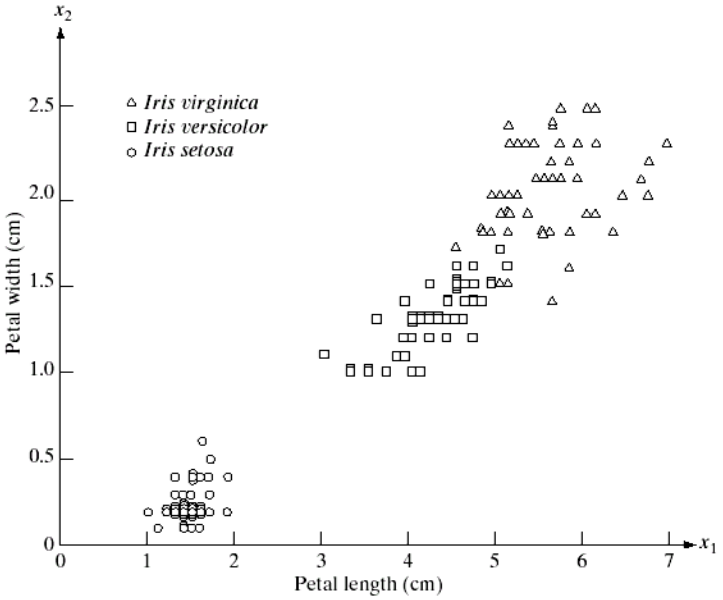


Εικόνα 12: κατάτμηση εικόνας (image segmentation)

ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Η αναγνώριση αντικειμένων [9][1] αναφέρεται στους αλγόριθμους που τα συστήματα τεχνητής όρασης και ψηφιακής επεξεργασίας εικόνας χρησιμοποιούν για να εξάγουν συμπεράσματα για το είδος του αντικειμένου και να το εντάξουν σε κάποια κατηγορία αντικειμένων, οπότε μιλάμε για κατηγοριοποίηση αντικειμένων (Classification), ή σε αλγόριθμους ταυτοποίησης για συγκεκριμένες υποστάσεις αντικειμένων (recognition, identification). Για τον άνθρωπο η κατηγοριοποίηση και αναγνώριση αντικειμένων είναι εύκολη υπόθεση για αντικείμενα τα οποία έχουμε μάθει να αναγνωρίζουμε. Αρκεί μόνο να ρίξουμε μια ματιά γύρο μας και αμέσως μπορούμε να αναγνωρίσουμε χιλιάδες αντικείμενα. Στην τεχνητή όραση δύστυχος δεν ισχύει κάτι τέτοιο. Η οπτική αναπαράσταση μιας εικόνας δεν δίνει πληροφορίες στον υπολογιστή για τα αντικείμενα που περιέχει , παρά μόνο για τις φωτεινότητες των εικονοστοιχείων της εικόνας. Απαραίτητη λοιπόν προϋπόθεση είναι η επεξεργασία της εικόνας για να εξαχθούν χαρακτηριστικά με μορφή διανυσμάτων τα οποία μπορεί εύκολα να χειριστεί ο υπολογιστής και να τα εισάγει στη συνέχεια σε ένα αλγόριθμο αναγνώρισης είτε ως παραδείγματα για την εκπαίδευση του συστήματος είτε ως δεδομένα

εισόδου για να πάρει από το σύστημα (εφόσον έχει εκπαιδευτεί) αποτελέσματα για το είδος ή την ταυτότητα του αντικειμένου που αναπαριστούν.



Εικόνα 13: Αναγνώριση αντικειμένων (object recognition)

ΤΕΧΝΗΤΗ ΌΡΑΣΗ

Η τεχνητή όραση [4][14][15] είναι το πεδίο της επιστήμης υπολογιστών που ασχολείται με την λήψη, την επεξεργασία και κατανόηση εικόνων και άλλων σχετικών δεδομένων από τον πραγματικό κόσμο με σκοπό την εξαγωγή συμβολικών χαρακτηριστικών και την λήψη αποφάσεων. Μια θεωρητική ιδέα για ένα σύστημα τεχνητής όρασης θα μπορούσε να είναι η προσομοίωση μέσω υπολογιστή του συστήματος της ανθρώπινης όρασης (βιολογική όραση) και της νοητικής διαδικασίας πίσω από αυτή.

Μπορεί η παραπάνω θεωρητική ιδέα να μην έχει ακόμα υλοποιηθεί, κυρίως λόγω της πολυπλοκότητας που περικλείει και της έλλειψης στοιχείων για τη λειτουργία της βιολογικής όρασης, ωστόσο η τεχνητή όραση έχει βρει εφαρμογές σε διάφορους τομείς, από τη βιομηχανία όπου ένα σύστημα τεχνητής όρασης μπορεί να υλοποιεί τον ποιοτικό έλεγχο της γραμμής παραγωγής μέχρι την έρευνα και την τεχνητή νοημοσύνη όπου ένας υπολογιστής ή ένα ρομπότ μπορεί να κατανοεί και να αλληλεπιδρά με το περιβάλλον του.

Εικόνα 14: Σύστημα τεχνητής όρασης που επιθεωρεί μπουκάλια

Εικόνα 15: Ρομπότ που χρησιμοποιεί τεχνητή όραση για να αλληλεπιδρά με αντικείμενα

ΤΟΜΕΙΣ ΕΦΑΡΜΟΓΗΣ ΤΕΧΝΗΤΗΣ ΟΡΑΣΗΣ

Σαν επιστημονικό πεδίο, η τεχνητή όραση ασχολείται με τη θεωρία εξαγωγής πληροφοριών από εικόνες με τη χρήση τεχνιτών μέσων. Οι εικόνες μπορεί να προέρχονται από διάφορες πηγές όπως απλές εικόνες, βίντεο, εικόνες από ιατρικά μηχανήματα, τρισδιάστατες εικόνες κλπ. Σε τεχνολογικό επίπεδο οι μέθοδοι και οι τεχνικές της τεχνητής όρασης συνδυάζονται σε συστήματα τεχνητής όρασης. Παραδείγματα τέτοιων συστημάτων είναι:

- Ο χειρισμός κάποιας διεργασίας π.χ. Ένα βιομηχανικό ρομπότ
- Η πλοήγηση π.χ. Σε ένα αυτόνομο όχημα
- Η ανίχνευση γεγονότων π.χ. Η καταμέτρηση ανθρώπων

- Η ταξινόμηση δεδομένων π.χ. Ταξινόμηση εικόνων με βάση το περιεχόμενο τους
- Η ανάλυση δεδομένων π.χ. Ανάλυση ιατρικών εικόνων και διάγνωση
- Η αλληλεπίδραση π.χ. Συσκευές για διεπαφή χρήστη-υπολογιστή

Στις περισσότερες πρακτικές εφαρμογές της τεχνητής όρασης τα συστήματα είναι προγραμματισμένα εκ των προτέρων για να εκτελούν κάποια πολύ συγκεκριμένη διαδικασία όταν πληρούνται οι προβλεπόμενες προϋποθέσεις, αλλά και τα συστήματα που στηρίζονται στην εκμάθηση (machine learning) όπως είναι και αυτό της εργασίας μας γίνονται όλο και ποιο διαδεδομένα.

Η τεχνητή όραση έχει άμεση σχέση με πολλά επιστημονικά πεδία. Τα κυριότερα είναι η ψηφιακή επεξεργασία εικόνας, η τεχνητή νοημοσύνη και η μηχανική όραση. Άλλα σχετικά πεδία είναι η φυσική, η νευροβιολογία, η επεξεργασία σήματος, και τα μαθηματικά.

Η φυσική συνδέεται στενά με την τεχνητή όραση, γιατί τα περισσότερα συστήματα τεχνητής όρασης χρησιμοποιούν αισθητήρες για την λήψη των εικόνων οι οποίοι βασίζονται στη φυσική για τη λειτουργία τους, οι οποίοι ανιχνεύουν την ηλεκτρομαγνητική ακτινοβολία όπως το ορατό φώς ή την υπέρυθρη ακτινοβολία (όπως το Kinect που θα δούμε αργότερα). Οι αισθητήρες αυτοί σχεδιάζονται χρησιμοποιώντας τη φυσική στερεάς κατάστασης. Ο τρόπος με τον οποίο το φώς αντανακλάται στις επιφάνειες εξηγείται με την χρήση της φυσικής. Η φυσική περιλαμβάνει την οπτική η οποία είναι σημαντικός παράγοντας στα περισσότερα συστήματα εικόνας. Οι εξελιγμένοι αισθητήρες χρησιμοποιούν ακόμα και την κβαντική φυσική. Από την άλλη η τεχνητή όραση μπορεί να βοηθήσει στη φυσική λύνοντας διάφορα προβλήματα μέτρησης όπως αυτό της κίνησης στα υγρά.

Στην τεχνητή νοημοσύνη θα συναντήσουμε πολλά κοινά με την τεχνητή όραση. Ένα σύστημα τεχνητής όρασης θα μπορούσε να αποτελεί μέρος ενός συστήματος τεχνητής νοημοσύνης, πχ σε ένα ρομπότ με τεχνητή νοημοσύνη η τεχνητή όραση θα μπορούσε να παρέχει τα μάτια του ρομπότ και να δίνει στο σύστημα πληροφορίες υψηλού επιπέδου για να μπορεί να κινείται ή να αλληλεπιδρά με το περιβάλλον του. Τομείς όπως η αναγνώριση προτύπων και ή εκπαίδευση μηχανών χρησιμοποιούνται ευρέως τόσο από την τεχνητή νοημοσύνη όσο και από την τεχνητή όραση.

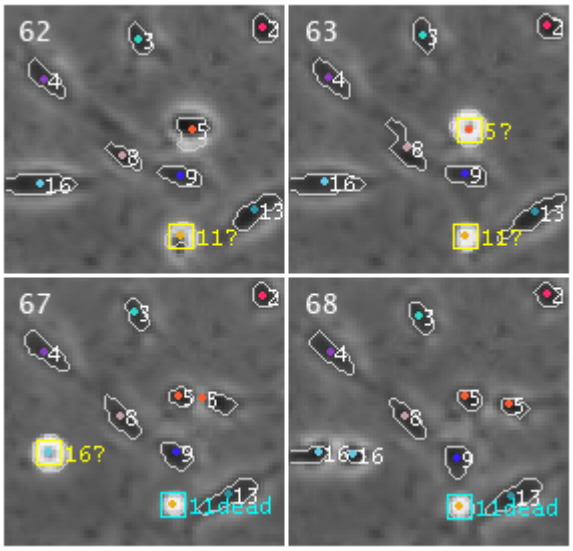
Η νευροβιολογία έχει σχέση με την τεχνητή όραση κυρίως στο κομμάτι που μελετά την βιολογική όραση. Η σημαντική πρόοδος στη μελέτη της βιολογικής όρασης που παρουσιάζεται τα τελευταία εκατό χρόνια και η μελέτη των νευρώνων και της λειτουργίας του εγκεφάλου όταν επεξεργάζεται οπτικές πληροφορίες είχε σαν αποτέλεσμα την καλύτερη κατανόηση για το πώς η βιολογική όραση λύνει ορισμένα προβλήματα. Πολλές από τις μεθόδους της τεχνητής όρασης προσπαθούν να μιμηθούν αυτές τις διεργασίες. Παραδείγματα είναι οι στερεοσκοπικές κάμερες για την λήψη τρισδιάστατης πληροφορίας και τα τεχνητά νευρωνικά δίκτυα για την ταξινόμηση εικόνων

Ένα άλλο σχετικό πεδίο είναι αυτό της ψηφιακής επεξεργασίας σήματος. Στην τεχνητή όραση πολλές από τις μεθόδους της ψηφιακής επεξεργασίας σήματος μπορούν να επεκταθούν για να εφαρμόζονται σε πολυδιάστατα σήματα όπως οι εικόνες. Κλασικό παράδειγμα είναι τα φίλτρα. Επίσης η ανάγκη για τέτοιες μεθόδους οδήγησε στην δημιουργία νέων μεθόδων που παρότι εντάσσονται στην ψηφιακή επεξεργασία σήματος έχουν εφαρμογές μόνο στην τεχνητή όραση.

Ακόμα, τα μαθηματικά βρίσκουν ευρεία εφαρμογή στην τεχνητή όραση. Ιδίως η γραμμική άλγεβρα, η στατιστική και η γεωμετρία. Η χρήση των μαθηματικών λύνει πρακτικά προβλήματα όπως τη μείωση της απαιτούμενης επεξεργαστικής ισχύος του

συστήματος, δίνοντας του την δυνατότητα να λειτουργεί σε πρακτικές εφαρμογές και μειώνοντας το κόστος υλοποίησης του.

Στην Ιατρική η τεχνητή όραση είναι αρκετά διαδεδομένη, τόσο ώστε η «ιατρική τεχνητή όραση» να είναι από μόνη της ερευνητικό πεδίο. Η βασικότερη χρήση της είναι η εξαγωγή συμπερασμάτων ή πληροφοριών από ιατρικές εικόνες όπως ακτινογραφίες, αγγειογραφίες μαγνητικές τομογραφίες και άλλοι τύποι ιατρικών δεδομένων που εμπεριέχουν εικόνα, για να υποβοηθήσει τον γιατρό στις διαδικασίες διάγνωσης και θεραπείας. Συχνά χρησιμοποιείτε για τον εντοπισμό όγκων ή άλλων παθογόνων στοιχείων και για να λύνει προβλήματα μετρήσεων . Στη χειρουργική μπορεί να βοηθήσει στον χειρισμό ιατρικών μηχανημάτων, ελευθερώνοντας έτσι τα χέρια του χειρούργου.



Εικόνα 16: Εφαρμογή τεχνητής όρασης για την παρακολούθηση κυττάρων

Ο στρατός είναι ο μεγαλύτερος επενδυτής σε συστήματα τεχνητής όρασης. Προφανή παραδείγματα είναι ο εντοπισμός εχθρικών στόχων και η καθοδήγηση πυραύλων. Τα ποιο εξελιγμένα συστήματα στέλνονται σε μια περιοχή και εντοπίζουν δυναμικά τον στόχο, αντί να προγραμματίζονται εκ τον προτέρων. Επίσης ο στρατός χρησιμοποιεί πλέων και μη επανδρωμένα αεροσκάφη και οχήματα ο χειρισμός των οποίων χρησιμοποιεί συνήθως και τεχνητή όραση.

Μια άλλη εφαρμογή της τεχνητής όρασης είναι στη βιομηχανία. Σε αυτή την περίπτωση συχνά ονομάζετε και μηχανική όραση. Στις βιομηχανικές εφαρμογές οι πληροφορίες που εξάγονται με τη χρήση της τεχνητής όρασης χρησιμοποιούνται για να υποστηρίξουν την παραγωγική διαδικασία. Ένα τέτοιο παράδειγμα είναι ο ποιοτικός έλεγχος. Ένα σύστημα τεχνητής όρασης μπορεί να ελέγχει για ελαττωματικά προϊόντα που περνάνε από μπροστά του σε ένα διάδρομο και να τα απορρίπτει ή να τους αλλάζει πορεία ούτως ώστε να ελεγχθούν περεταίρω από κάποιο άνθρωπο. Ένα άλλο παράδειγμα είναι ένα βιομηχανικό ρομπότ ή ένας βραχίονας ό οποίος χρησιμοποιεί την τεχνητή όραση για να χειρίζεται αντικείμενα. Η τεχνητή όραση χρησιμοποιείτε και στην αγρό-βιομηχανία για να διαχωρίζει τρόφιμα.

Στα κινηματογραφικά στούντιο και στα εργαστήρια παιχνιδιών για υπολογιστές η τεχνητή όραση χρησιμοποιείτε για να παράγει αληθοφανή οπτικά εφέ [16]. Αρχικά ανιχνεύετε η κίνηση σε ένα πραγματικό ηθοποιό και στη συνέχεια προσαρμόζετε σε ένα τρισδιάστατο μοντέλο. Έτσι η κίνηση στο μοντέλο φαίνεται ποιο φυσική και ρεαλιστική. Τα δεδομένα που εξάγονται με ένα σύστημα τεχνητής όρασης εισάγονται σε ένα σύστημα

γραφικής ή οποία είναι κατά κάποιο τρόπο το αντίθετο της τεχνητής όρασης. Στη γραφική η είσοδος του συστήματος είναι δεδομένα και η έξοδος είναι εικόνα.

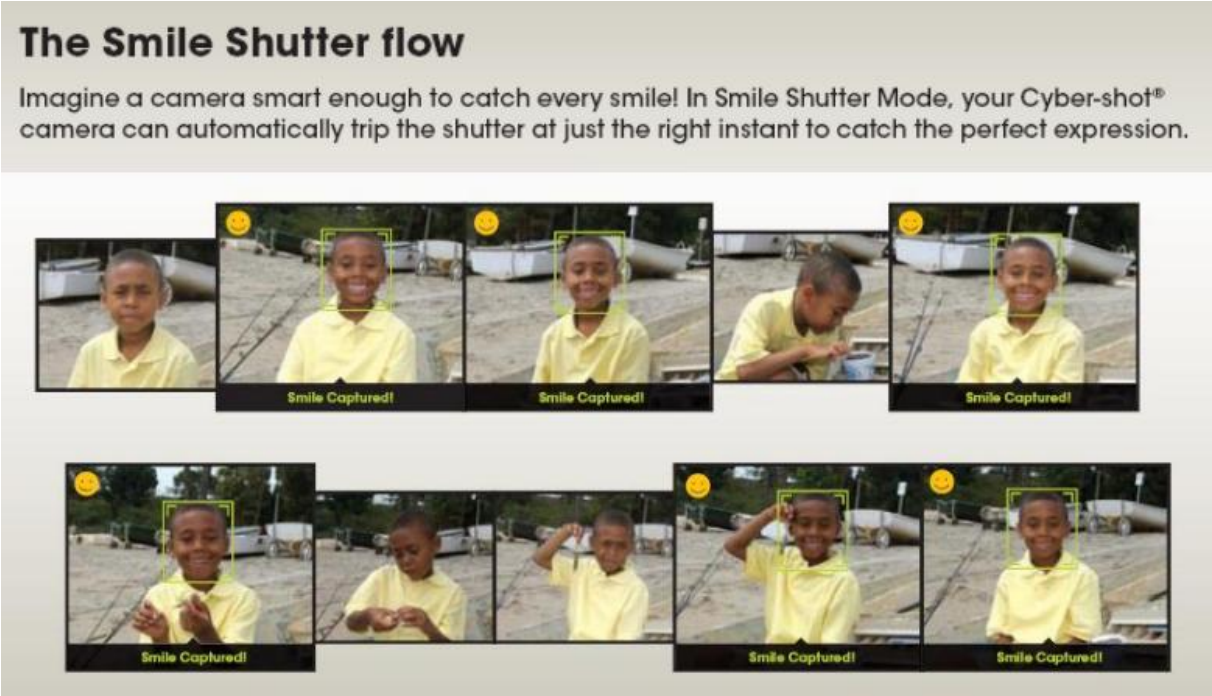
Εικόνα 17: Χρήση τεχνητής όρασης και γραφικής για οπτικά εφέ.

Στη βιομετρική η τεχνητή όραση χρησιμοποιείται για την αναγνώριση δακτυλικών αποτυπωμάτων. Πολλά συστήματα πλέων αντί για κωδικό πρόσβασης σαν συνθηματικό δέχονται την έξοδο τέτοιων συσκευών. Ακόμα η ταυτοποίηση μπορεί να γίνεται με την αναγνώριση της ίριδος του ματιού, η οποία επίσης είναι μοναδική σε κάθε άνθρωπο. Το ντοκιμαντέρ του National Geographic που αναλύει την ιστορία ενός Αφγανού κοριτσιού [17] που φωτογραφήθηκε σε ηλικία 12 ετών το 1984 και εντοπίστηκε με τη χρήση αυτής της μεθόδου από άλλη φωτογραφία 18 χρόνια αργότερα σε κάποια απομακρυσμένο σημείο του Αφγανιστάν είναι ένα τρανό παράδειγμα της αποτελεσματικότητας της μεθόδου.

Εικόνα 18: Το Αφγανό κορίτσι στην πρώτη φωτογραφία και 18 χρόνια αργότερα.

Στα κινητά τηλέφωνα η εφαρμογές της τεχνητής όρασης αυξάνονται με ραγδαίο ρυθμό. Από απλή αναγνώριση κειμένου σε φωτογραφίες τραβηγμένες από κινητό μέχρι πολύπλοκες εφαρμογές όπως το Google Goggles [18], το οποίο κάνει αναζήτηση στο διαδίκτυο με την χρήση των εικόνων αυτόν. Άλλες εφαρμογές όπως αυτές που συναντάμε στις ψηφιακές φωτογραφικές μηχανές εντοπίζουν το πρόσωπο του ατόμου που φωτογραφίζεται και μπορούν να εφαρμόσουν διάφορα εφέ, ή ακόμα και να εντοπίσουν το χαμόγελο [19] ούτως ώστε η φωτογραφία να βγαίνει αυτόματα μόλις αυτό εντοπιστεί.

Εικόνα 19: Εφαρμογή Google Goggles για κινητά android

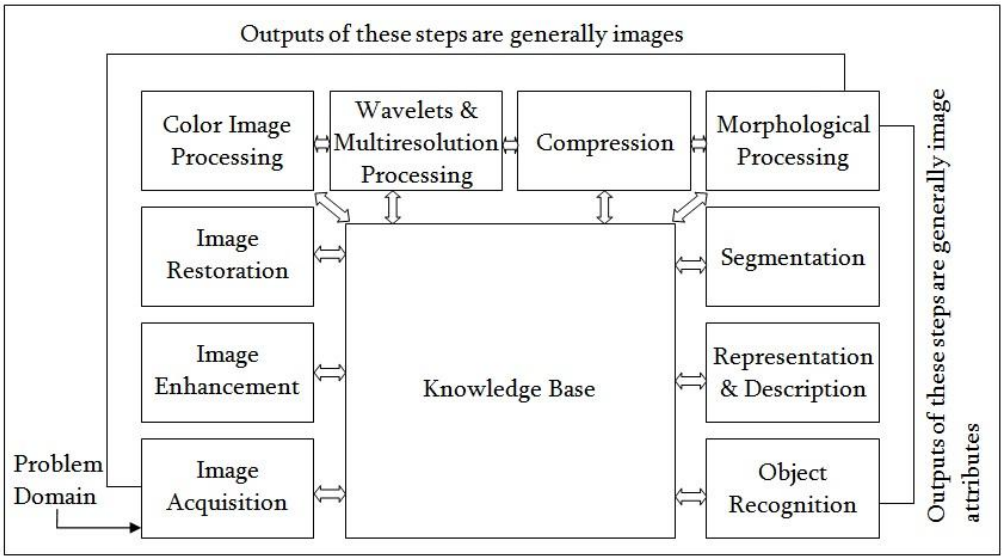


Εικόνα 20: Εντοπισμός χαμόγελου με ψηφιακή φωτογραφική μηχανή

ΣΥΣΤΗΜΑΤΑ ΤΕΧΝΗΤΗΣ ΟΡΑΣΗΣ

Η οργάνωση ενός συστήματος τεχνητής όρασης εξαρτάται σε μεγάλο βαθμό από την εφαρμογή για την οποία προορίζεται. Μερικά συστήματα είναι αυτοδύναμα και λύνουν συγκεκριμένα προβλήματα μετρήσεων ή εντοπισμού, ενώ άλλα αποτελούν υποσυστήματα ενός μεγαλύτερου συστήματος το οποίο μπορεί να περιλαμβάνει και άλλα υποσυστήματα όπως για παράδειγμα ένα υποσύστημα για τον έλεγχο των μηχανικών μερών σε ένα μηχάνημα ή ένα ρομπότ. Η σύσταση του συστήματος εξαρτάται επίσης από το αν η λειτουργικότητα του προγράμματος είναι προκαθορισμένη ή σε κάποια κομμάτια τροποποιείται π.χ. εκπαιδεύεται κατά την χρήση.

Πολλές από τις συναρτήσεις σε ένα σύστημα μπορεί να υπάρχουν αποκλειστικά σε αυτό, αλλά υπάρχουν και τυπικές συναρτήσεις που συναντάμε σε πολλά συστήματα τεχνητής όρασης όπως αυτές που αναφέρουμε παρακάτω.



Εικόνα 21: Διεργασίες που μπορεί να περιλαμβάνει ένα σύστημα τεχνητής όρασης

- **Λήψη εικόνας**

Μια ψηφιακή εικόνα [4] που δημιουργείται από ένα ή περισσότερους αισθητήρες οι οποίοι εκτός από τους φωτοευαίσθητους αισθητήρες μπορεί να είναι αισθητήρες απόστασης, συσκευές τομογραφίας, κάμερες υπερήχων, κλπ. Ανάλογα με τον τύπο του αισθητήρα τα δεδομένα εικόνας που παράγονται μπορεί να είναι κανονικές δισδιάστατες εικόνες, τρισδιάστατα δεδομένα όγκου, ή σειρές εικόνων. Τα δεδομένα των εικονοστοιχείων μπορεί να αντιπροσωπεύουν τιμές φωτεινότητας στο ορατό φάσμα (όπως στις έγχρωμες εικόνες ή στις εικόνες της κλίμακας του γκρι), αλλά μπορεί και να συσχετίζονται και με άλλες φυσικές μετρήσεις όπως το βάθος, η απορρόφηση ή η αντανάκλαση υπερήχων ή ηλεκτρομαγνητικών κυμάτων.

- **Προ-επεξεργασία**

Πριν εφαρμοστεί μια μέθοδος τεχνητής όρασης για την εξαγωγή πληροφορίας από τις εικόνες, είναι πολλές φορές απαραίτητο να γίνει κάποια επεξεργασία για να εξασφαλιστεί ότι η εικόνα πληροί κάποιες προϋποθέσεις που πιθανόν να απαιτεί η μέθοδος. Παραδείγματα είναι :

  - Τροποποίηση του μεγέθους της εικόνας για να συμβαδίζει με τις απαιτήσεις του συστήματος.
  - Μείωση του θορύβου για να επιβεβαιωθεί ότι ο θόρυβος από τον αισθητήρα δεν θα παράγει λάθος αποτελέσματα .
  - Ενίσχυση της αντίθεσης για να γίνουν ποιο ευδιάκριτα τα χαρακτηρίστηκα της εικόνας όπως ακμές και επιφάνειες.
  - Μετασχηματισμός της εικόνας για γενικοποίηση των χαρακτηριστικών (Scale space representation).

- **Εξαγωγή Χαρακτηριστικών**

Τα χαρακτηριστικά της εικόνας εξάγονται από την εικόνα. Τα χαρακτηριστικά αυτά μπορεί να έχουν διάφορα επίπεδα πολυπλοκότητας. Παραδείγματα είναι:

  - Ακμές , Γραμμές και Κορυφογραμμές
  - Τοπικά σημεία ενδιαφέροντος όπως Γωνίες και κηλίδες
  - Άλλα πολύπλοκα χαρακτηριστικά που μπορεί να σχετίζονται με την υφή ή την κίνηση κλπ

- **Εντοπισμός/Κατάτμηση**

Ορισμένα σημεία ή περιοχές ενδιαφέροντος εντοπίζονται στην εικόνα που θεωρούνται σχετικά και χρειάζονται περισσότερη επεξεργασία. Παραδείγματα είναι:

- Μια επιλογή από μια συγκεκριμένη ομάδα σημείων ενδιαφέροντος
- Κατάτμηση μιας ή περισσότερων περιοχών από την εικόνα που περιέχουν ένα αντικείμενο ενδιαφέροντος.

- **Υψηλού επιπέδου επεξεργασία**

Σε αυτό το σημείο η είσοδος είναι συνήθως ένα μικρό σύνολο από δεδομένα όπως για παράδειγμα ένα σύνολο από σημεία ή μια περιοχή της εικόνας που υποτίθεται ότι περιέχει κάποιο αντικείμενο. Στη συνέχεια η εφαρμογή εκτελεί εργασίες όπως:

- Επαληθεύει ότι δα δεδομένα ικανοποιούν τις υποθέσεις της εφαρμογής ή ότι συμπίπτουν με κάποιο μοντέλο
- Υπολογίζει την θέση ή το μέγεθος του αντικειμένου
- Κατηγοριοποιεί το αντικείμενο σε μια κλάση αντικειμένων
- Ταυτοποιεί το αντικείμενο με μια άλλη εικόνα του ίδιου αντικειμένου σε διαφορετική θέση

- **Λήψη αποφάσεων**

Λαμβάνει τις τελικές αποφάσεις που απαιτούνται από την εφαρμογή όπως για παράδειγμα:

- Έγκριση/ Απόρριψη σε εφαρμογές επιθεώρησης
- Αναγνώριση/μη-αναγνώριση σε εφαρμογές αναγνώρισης
- Σήμανση για περεταίρω ανθρώπινη επεξεργασία σε εφαρμογές ιατρικής, στρατιωτικές εφαρμογές ή εφαρμογές ασφάλειας και αναγνώρισης.

ΑΛΓΟΡΙΘΜΟΙ

ΑΝΙΧΝΕΥΣΗ ΑΚΜΩΝ

Με τον όρο ακμές [21] για μια ασπρόμαυρη εικόνα, αναφερόμαστε σε αλλαγές της φωτεινότητας μεταξύ γειτονικών περιοχών της. Αλλαγές της φωτεινότητας συνήθως αντιστοιχούν σε διαφοροποίηση ιδιοτήτων της απεικόνισης τρισδιάστατων αντικειμένων όπως αλλαγές της υφής, του βάθους, όρια αντικειμένων, διαφορετικό φωτισμό και αντανάκλαση. Έτσι με την ανίχνευση ακμών μπορούμε να αντλήσουμε πληροφορίες για φυσικές ιδιότητες για τα εικονιζόμενα πραγματικά αντικείμενα.

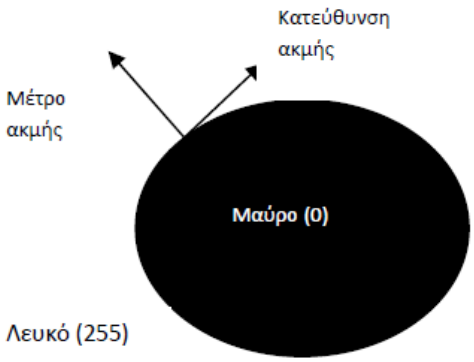
Η ανίχνευση ακμών μιας εικόνας παρουσιάζει αρκετές δυσκολίες. Οι ακμές μπορεί να χαρακτηρίζονται από προοδευτικές ή ακόμα και πολύ μικρές αλλαγές στην φωτεινότητα της εικόνας. Η παρουσία θορύβου σε μια εικόνα μπορεί να οδηγήσει στην ανίχνευση εσφαλμένων ακμών αλλοιώνοντας τα όρια των αντικειμένων. Ο διαφορετικός φωτισμός και η σκίαση μπορεί να ανιχνευτούν σαν ψευδοακμές ενώ δεν αντιστοιχούν σε φυσική ακμή. Ακόμα και αντικείμενα διαφορετικής κλίμακας πιθανό να βρίσκονται στην ίδια εικόνα.
Σε συστήματα βιολογικής όρασης υπάρχουν νευροβιολογικές και ψυχοφυσικές ενδείξεις ότι στα πρώτα στάδια επεξεργασίας της οπτικής πληροφορίας γίνεται κάποιο είδος ανίχνευσης ακμών. Αυτή η επεξεργασία μοιάζει με ζωνοπερατά επιλεκτικά φίλτρα ή ισοδύναμα με συνέλιξη της οπτικής πληροφορίας με νευρικές αποκρίσεις. Αυτά τα φίλτρα έχουν μοντελοποιηθεί ως διαφορές από Gabor ή Gaussian φίλτρα.

Η ανίχνευση ακμών αποτελεί την βάση για μετέπειτα επεξεργασία μια εικόνας ή ακολουθίας εικόνων με αλγορίθμους υπολογιστικής όρασης, όπως ανάλυση υφής, τμηματοποίησης, ανίχνευσης κίνησης, στερέοψης και αναγνώρισης προτύπων. Γι’ αυτό πρέπει να δίνει αξιόπιστα αποτελέσματα και να υλοποιείται αποδοτικά.

ΤΥΠΟΙ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΑΚΜΩΝ

Υπολογιστικά οι ακμές (αλλαγές στην συνάρτηση της έντασης) για συνεχείς συναρτήσεις μπορούν να υπολογιστούν με τον υπολογισμό της πρώτης παραγώγου και εντοπισμό των τοπικών μέγιστων. Μια δεύτερη μέθοδος με πλεονεκτήματα σε αξιοπιστία στηρίζεται στις διελεύσεις της δεύτερης παραγώγου από το μηδέν (zero crossing). Φυσικά επειδή έχουμε συναρτήσεις δύο μεταβλητών (x,y συντεταγμένη) θα υπολογίζουμε τις μερικές παραγώγους.

Μια μεταβολή της συνάρτησης της εικόνας μπορεί να περιγραφεί με την βάθμωση (gradient) προς την κατεύθυνση της μέγιστης μεταβολής. Μια ακμή είναι ιδιότητα του κάθε εικονοστοιχείου ξεχωριστά και υπολογίζεται από την συμπεριφορά της συνάρτησης της εικόνας σε μια περιοχή γειτονικών εικονοστοιχείων. Πρόκειται για διανυσματική μεταβλητή με μέτρο και κατεύθυνση.



Εικόνα 22: Κατεύθυνση και μέτρο ακμής

Το μέτρο της ακμής μας δείχνει πόσο μεγάλη είναι μεταβολή της συνάρτησης φωτεινότητας (ισχυρή, αδύναμη ακμή) και η κατεύθυνση μας δίνει τον προσανατολισμό της ακμής στην εικόνα, και υπολογίζονται ως εξής.

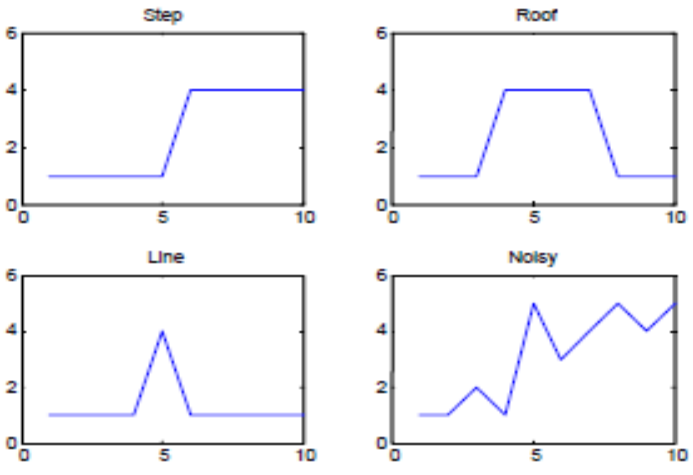
Για το μέτρο της ακμής:

$$|grad(I(x,y)| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

Και για την κατεύθυνση της ακμής:

$$\varphi = \arg\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$$

Τέλος υπάρχουν διάφορα είδη ακμών.



Εικόνα 23: είδη ακμών σε grayscale εικόνες

Η ακμή τύπου στέγης ανταποκρίνεται σε λωρίδες ίδιας έντασης στην εικόνα, και η ακμή τύπου γραμμής αναφέρεται σε μικρότερο εύρος. Η βηματική ακμή είναι η διαχωριστική επιφάνεια δύο αντικειμένων ή ενός αντικειμένου και του περιβάλλοντα χώρου. Η θορυβώδης

ακμή είναι μια βηματική ακμή αλλά με τα εικονοστοιχεία να λαμβάνουν ανομοιόμορφες τιμές φωτεινότητας κατά τη μετάβαση μεταξύ των δύο επιπέδων.

Όταν δεν μας ενδιαφέρει η κατεύθυνση παρά μόνο το μέτρο των ακμών τότε με ανίχνευση των διελεύσεων της δεύτερης παραγώγου από το μηδέν επιτυγχάνουμε καλύτερα αποτελέσματα σε αξιοπιστία και υπολογιστικό κόστος. Ο υπολογισμός της δεύτερης παραγώγου επιτυγχάνεται χρησιμοποιώντας μικρά μητρώα συνέλιξης που λειτουργούν σαν ψηφιακοί πυρήνες λαπλασιανών φίλτρων. Υπολογίζουμε δηλαδή:

$$Laplacian = \nabla^2 I(x, y)$$

Οι διάφοροι ανιχνευτές ακμών συνήθως σχεδιάζονται και είναι αποτελεσματικοί για ένα είδος ακμών. Στην συνέχεια της ανάλυσής μας θα ασχοληθούμε με τις βηματικές ακμές που είναι οι πιο συνηθισμένες και προσφέρουν τις περισσότερες πληροφορίες για μια εικόνα.

ΓΡΑΜΜΙΚΟΙ ΤΕΛΕΣΤΕΣ ΑΝΙΧΝΕΥΣΗΣ ΑΚΜΩΝ ΠΡΟΣΕΓΓΙΖΟΝΤΑΣ 1Η ΠΑΡΑΓΩΓΟ

Ιστορικά η πρώτη απόπειρα ανίχνευσης ακμών, που διήρκεσε περίπου 30 χρόνια (δεκαετία 50έως δεκαετία 70), έγινε υπολογίζοντας διακριτές προσεγγίσεις των μερικών παραγώγων κατά κατεύθυνση για την υπό επεξεργασία εικόνα. Αυτό γίνεται με την συνέλιξη της εικόνας και ενός μικρού μητρώου που στόχο έχει να ενισχύσει την ένταση των ακμών. Το πιο παλιό από αυτά τα μητρώα προτάθηκε από τον Roberts.

ΤΕΛΕΣΤΕΣ ROBERTS

Τα μητρώα που προτείνει ο Roberts [22] για τον υπολογισμό της πρώτης παραγώγου της συνάρτησης φωτεινότητας της εικόνας είναι τα εξής:

$$R_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, R_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Για μια εικόνα

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Το μητρώο R_1 συνελισσόμενο με την εικόνα δίνει στην έξοδο:

$$\begin{bmatrix} (-1)a_{11} & (0)a_{12} & a_{13} \\ (0)a_{21} & (1)a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \overset{*R_1}{\Rightarrow} \begin{bmatrix} a_{22} - a_{11} & a_{22} - a_{13} & .. \\ a_{32} - a_{21} & a_{33} - a_{22} & .. \\ .. & .. & .. \end{bmatrix}$$

Αντίστοιχα για το μητρώο R_2 παίρνουμε:

$$\begin{bmatrix} (0)a_{11} & (-1)a_{12} & a_{13} \\ (1)a_{21} & (0)a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \overset{*R_2}{\Rightarrow} \begin{bmatrix} a_{21} - a_{12} & a_{22} - a_{13} & .. \\ a_{31} - a_{22} & a_{32} - a_{23} & .. \\ .. & .. & .. \end{bmatrix}$$

Τώρα με χρήση κάποιας νόρμας μπορούμε να υπολογίσουμε το μέτρο των ακμών και με χρήση κατωφλίωσης να αποφανθούμε για τις ακμές της εικόνας. Οι πιο συνηθισμένες νόρμες που χρησιμοποιούνται είναι οι εξής:

$$\sqrt{f_x^2 + f_y^2} \quad (1)$$

$$|f_x| + |f_y| \quad (2)$$

$$\max(|f_x|, |f_y|) \quad (3)$$

Με χρήσης της νόρμας 2 για παράδειγμα προκύπτει ο πίνακας του μέτρου των ακμών. Τα στοιχεία του υπολογίζονται ως εξής:

$$Edge_{i,j} = |I(i,j) - I(i+1,j+1)| + |I(i,j+1) + I(i+1,j)|$$

Μετά τον υπολογισμό του μέτρου της ακμής με την κατάλληλη νόρμα, με την τεχνική της κατωφλίωσης ανιχνεύουμε τα τοπικά μέγιστα της φωτεινότητας της εικόνας και αποφασίζουμε τι θα δεχθούμε ως ακμές. Η κατωφλίωση θα οδηγήσει τα εικονοστοιχεία με τιμή έντασης μικρότερη από το κατώφλι στην δυαδική τιμή ‘0’ και αυτά με μεγαλύτερες τιμές στην δυαδική τιμή ‘1’(εικονοστοιχείο ακμής).

ΤΕΛΕΣΤΕΣ PREWITT

Οι τελεστές Prewitt [23] προσεγγίζουν την μερική παράγωγο πρώτης τάξης κατά κατεύθυνση για την εικόνα. Υπάρχουν 8 διαφορετικές κατευθύνσεις για τις οποίες μπορούμε να υπολογίσουμε την μερική παράγωγο, δύο όμως αρκούν για να εντοπίσουμε τις ακμές στην περίπτωση που μας ενδιαφέρει μόνο το μέτρο της ακμής.

$$P_{1=}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, P_{1=}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Οι νόρμες για τις υπόλοιπες κατευθύνσεις μπορούν να προκύψουν με απλή περιστροφή των περιφερειακών στοιχείων της P1. Η διαδικασία εντοπισμού των ακμών παραμένει ίδια με αυτή για τον τελεστή Roberts.

ΤΕΛΕΣΤΕΣ SOBEL

Και οι τελεστές Sobel, [24] όπως και οι επόμενοι που θα αναφέρουμε, προσεγγίζουν την πρώτη μερική παράγωγο κατά κατεύθυνση. Και αυτά τα μητρώα συνέλιξης (convolution kernels) είναι τρία επί τρία, και η διαδικασία για την ανίχνευση των ακμών ίδια με αυτή που χρησιμοποιήθηκε παραπάνω. Και σε αυτή την περίπτωση υπάρχουν οκτώ διαφορετικές κατευθύνσεις που μπορούμε να ανιχνεύσουμε ακμές. Δύο από αυτά τα μητρώα συνέλιξης είναι:

$$S_{1=}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, S_{2=}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

ΤΕΛΕΣΤΕΣ KIRCH, ROBINSON

Και οι τελεστές Kirch [25] και Robinson προσεγγίζουν την πρώτη παράγωγο. Τα μητρώα τους επίσης υπολογίζουν κατευθυντικές παραγώγους και έχουν τις ίδιες ιδιότητες με αυτές που έχουμε προαναφέρει. Οι πυρήνες τους είναι οι ακόλουθοι:

$$K = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}$$

Και αντίστοιχα για τον Robinson convolution kernel:

$$R = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

Εκτός του τελεστή Robinson όλοι οι άλλοι έχουν διαστάσεις 3 επί 3. Παρά το μικρό τους μέγεθος εισάγουν αρκετά μεγάλη πολυπλοκότητα. Για τον υπολογισμό ενός pixel εξόδου χρειάζονται 6 πολλαπλασιασμοί και 5 προσθέσεις για κάθε μια από τις κατευθύνσεις που υπολογίζουμε την πρώτη παράγωγο. Μια επιπλέον πρόσθεση χρειάζεται για να πάρουμε το τελικό μέτρο της ακμής. Συνολικά 12 πολλαπλασιασμοί και 11 προσθέσεις, για ένα και μόνο εικονοστοιχείο. Φυσικά παραγοντοποιώντας μπορούμε να μειώσουμε τους πολλαπλασιασμούς σε 2 καθώς όλα τα μητρώα έχουν μόλις 2 μη μηδενικές τιμές για τα στοιχεία τους.

Μια επίσης σημαντική παρατήρηση είναι ότι το άθροισμα των στοιχείων του κάθε μητρώου είναι πάντα μηδέν. Έτσι πάντα όταν βρίσκεται σε εσωτερική περιοχή ενός αντικειμένου (φωτεινότητα σταθερή) η έξοδος είναι πάντα μηδέν. Όταν βρεθούμε όμως σε ακμή η έξοδος παίρνει μεγάλες τιμές. Αυτή είναι η ενίσχυση της ακμής και με αυτό τον τρόπο λειτουργούν τα μητρώα συνέλιξης που προσεγγίζουν την πρώτη παράγωγο.

ΓΡΑΜΜΙΚΟΙ ΤΕΛΕΣΤΕΣ ΑΝΙΧΝΕΥΣΗΣ ΑΚΜΩΝ ΠΟΥ ΠΡΟΣΕΓΓΙΖΟΥΝ ΤΗΝ 2Η ΠΑΡΑΓΩΓΟ.

Ένας εναλλακτικός τρόπος εύρεσης ακμών είναι με τον εντοπισμό των διελεύσεων της δεύτερης παραγώγου από το μηδέν (zero crossing). Οι εικόνες είναι συναρτήσεις δυο μεταβλητών κι έτσι η λαπλασιανή υπολογίζει το μέτρο (magnitude) της δεύτερης παραγώγου, και χωρίς να δίνει πληροφορία για την κατεύθυνση της ακμής. Αυτό όμως δεν μας δημιουργεί πρόβλημα αναφορικά με την εύρεση των ακμών, καθώς αυτό που μας ενδιαφέρει στις περισσότερες εφαρμογές είναι το μέτρο των ακμών και μόνο.

ΛΑΠΛΑΣΙΑΝΟΣ ΤΕΛΕΣΤΗΣ (LAPLACIAN OPERATOR)

Για μια συνεχή συνάρτηση η λαπλασιανή δίνεται από τον τύπο,

$$Laplacian = \nabla^2 I(x,y)$$

Για μια διακριτή συνάρτηση όπως η εικόνα, μπορεί να προσεγγιστεί από μικρά μητρώα συνέλιξης. Τα πιο δημοφιλή είναι :

$$L_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad L_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Η διαφορά των δύο μητρώων είναι η συσχετιστικότητα με τα γειτονικά εικονοστοιχεία. Το πρώτο λέμε ότι έχει συσχετιστικότητα 8, δηλαδή η έξοδος μετά την πράξη της συνέλιξης

εξαρτάται από τα 8 γειτονικά εικονοστοιχεία του εξεταζόμενου. Ενώ για το το δεύτερο μητρώο η συσχετιστικότητα είναι 4 καθώς εκτός του κεντρικού εικονοστοιχείου μόνο 4 ακόμη έχουν μη μηδενικές τιμές.

Με την χρήση αυτού του τελεστή μειώνουμε την πολυπλοκότητα υπολογισμού των ακμών σε σχέση με του τελεστές που προσεγγίζουν την πρώτη παράγωγο που προαναφέραμε. Με χρήση του τελεστή 8 συσχετιστικότητας, για κάθε εικονοστοιχείο ακμών χρειαζόμαστε 9 πολλαπλασιασμούς και 8 προσθέσεις, ενώ για το μητρώο με συσχετιστικότητα 4 ο αριθμός πέφτει σε 5 πολλαπλασιασμούς και 4 προσθέσεις. Φυσικά με παραγοντοποίηση και στις δύο περιπτώσεις η απαίτηση για πολλαπλασιασμούς πέφτει στους 2.

CANNY EDGE DETECTOR

Ο αλγόριθμος που πρότεινε ο Canny [20] για ανίχνευση ακμών σε εικόνες θεωρείται ο βέλτιστος που μπορούμε να ακολουθήσουμε για ανίχνευση ακμών παρουσία λευκού θορύβου. Για την υλοποίησή του απαιτούνται συγκεκριμένα βήματα όπως αναφέρει ο ίδιος στην δημοσίευσή του, *A Computational Approach to Edge Detection.*

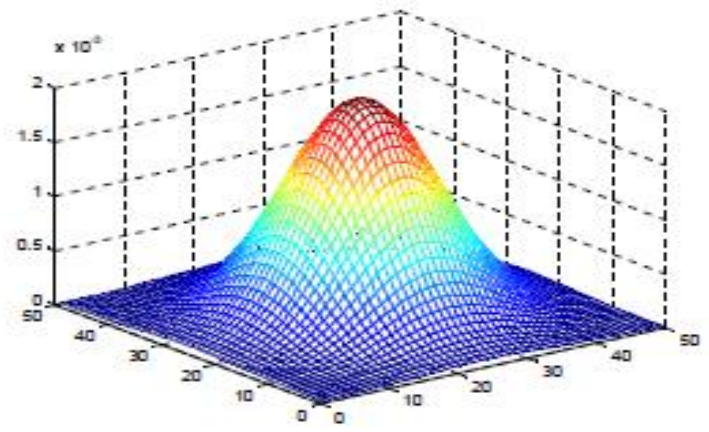
Πρόθεση του Canny ήταν να βελτιώσει τους ήδη υπάρχοντες αλγόριθμους όταν ερευνούσε την περιοχή της ανίχνευσης ακμών. Για να το πετύχει αυτό όρισε κάποια κριτήρια για να αξιολογήσει την αποτελεσματικότητα των αλγόριθμων αυτών.

Πρώτο και πιο προφανές κριτήριο ήταν η ελαχιστοποίηση του σφάλματος. Είναι πολύ σημαντικό να ανιχνεύονται όλες οι πραγματικές ακμές (πραγματική είναι μια ακμή που υφίσταται και στον τρισδιάστατο πραγματικό κόσμο), και ταυτόχρονα να μην ανιχνεύονται ακμές που δεν υπάρχουν, ή να έχουμε «διπλές» αποκρίσεις σε μια ακμή. Δεύτερο κριτήριο ήταν οι ακμές να είναι σωστά τοποθετημένες τοπικά. Η απόσταση μεταξύ της πραγματικής ακμής και της ακμής που εντοπίζει ο αλγόριθμος πρέπει να ελαχιστοποιηθεί. Επίσης η ακμή πρέπει να ορίζεται σαφώς και όχι να παίρνει εκτεταμένες διαστάσεις.

Βασιζόμενος σε αυτά τα κριτήρια ο Canny κατέληξε σε έναν αλγόριθμο όπου αρχικά στην εικόνα εφαρμόζεται ένα γκαουσιανό ψηφιακό φίλτρο (Gaussian). Αυτό στοχεύει στην ελαχιστοποίηση της επίδρασης του θορύβου, και η διαδικασία ονομάζεται ομαλοποίηση της εικόνας (smoothing). Η ψηφιακή μορφή του φίλτρου είναι ένα τετραγωνικό μητρώο συνέλιξης. Όσο μεγαλώνει η διάσταση του φίλτρου και η τυπική απόκλιση (σ) της γκαουσιανής δυσδιάστατης κατανομής, τόσο περισσότερο εξομαλύνεται η εικόνα και μειώνεται η επίδραση του λευκού θορύβου. Οι τιμές του γκαουσιανού φίλτρου δίνονται από την σχέση:

$$G(x,y) = \frac{e^{\frac{-(x^2+y^2)}{2\sigma^2}}}{\sum_{\forall x}\sum_{\forall y} e^{\frac{-(x^2+y^2)}{2\sigma^2}}}$$

και έχει την μορφή του σχήματος στην εικόνα



Εικόνα 24: Δισδιάστατο γκαουσιανό φίλτρο

Στον ανιχνευτή ακμών που πρότεινε ο Canny[20] μας δίνεται η δυνατότητα, ανάλογα με την τιμή της τυπικής απόκλισης που διαλέγουμε για το γκαουσιανό φίλτρο, να ανιχνεύσουμε λεπτομερείς ή γενικότερες ακμές. Η διαφορά στο αποτέλεσμα του αλγόριθμου για διάφορες τιμές τυπικής απόκλισης οφείλεται στο ότι μεγαλώνοντας η τιμή της τυπικής απόκλισης του φίλτρου τόσο περισσότερο ομαλοποιεί την εικόνα και ακμές με πλάτος μικρότερο (τύπου γραμμής και στέγης) από αυτό του πυρήνα της συνέλιξης ουσιαστικά εξαλείφονται από το φίλτρο.

Παρόλο που ο αλγόριθμος του Canny βελτιώνει αρκετά την ποιότητα των ανιχνευόμενων ακμών, είναι σαφές ότι αυξάνει κατά πολύ την πολυπλοκότητα αφού απαιτεί την συνέλιξη της εικόνας με δύο μητρώα συνέλιξης ένα εκ των οποίων μάλιστα (γκαουσιανό φίλτρο) πολύ μεγάλης διάστασης όσο αυξάνεται η τιμή της τυπικής απόκλισης *σ*. Επίσης και η κατωφλίωση με υστέρηση που ακολουθεί για την επιλογή των εικονοστοιχείων που απαρτίζουν ακμές εισάγει πρόσθετη πολυπλοκότητα και απαιτήσεις μνήμης.

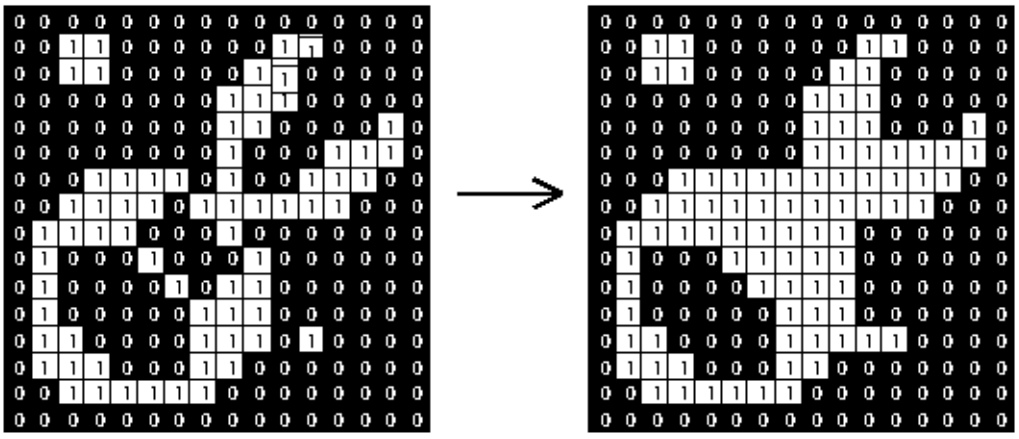
ΜΟΡΦΟΛΟΓΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Το Closing[12] (κλείσιμο) είναι ένας σημαντικός αλγόριθμος (operator) από τον τομέα της μαθηματικής μορφολογίας. Όπως και ο αντίστοιχος αλγόριθμος opening (άνοιγμα) μπορεί να εξαχθεί μέσα τις βασικές διεργασίες της συστολής και διαστολής. Χρησιμοποιείται συνήθως σε δυαδικές εικόνες (binary images), υπάρχουν όμως και εκδοχές στην κλίμακα του γκρίζου (graylevel versions). Το closing μοιάζει κάπως με την διαστολή στο ότι τείνει να μεγεθύνει τα όρια των φωτεινών περιοχών σε μια εικόνα (συρρικνώνοντας κενά στο φόντο σε τέτοιες περιοχές) αλλά είναι λιγότερο καταστροφικό από το αρχικό οριακό σχήμα. Όπως και με άλλους μορφολογικούς αλγόριθμους, η ακριβής λειτουργία του καθορίζεται από ένα δομικό στοιχείο. Ο σκοπός της χρήσης του αλγόριθμου αυτού είναι να διατηρηθούν περιοχές του φόντου οι οποίες έχουν ένα σχήμα παρόμοιο με αυτό το δομικό στοιχείο, ή οι οποίες μπορεί να περιλαμβάνουν εξ ολοκλήρου το δομικό στοιχείο, εξαλείφοντας παράλληλα όλες τις άλλες περιοχές με pixels που βρίσκονται στο υπόλοιπο φόντο.

Η διεργασία που ακολουθεί το closing είναι η αντίστροφη αυτής του opening. Μπορεί να προσδιοριστεί απλά ως μια διαστολή ακολουθούμενη από μια συστολή χρησιμοποιώντας το ίδιο δομικό στοιχείο και για τις δυο διεργασίες. Ο αλγόριθμος του closing έχει δύο προαπαιτούμενα, μια εικόνα η οποία να θέλει κλείσιμο και ένα δομικό στοιχείο. Ένα graylevel closing αποτελείται αποκλειστικά από μια graylevel διαστολή ακολουθούμενη από μια graylevel συστολή.

Το closing είναι ουσιαστικά το αντίστοιχο του opening, διότι εφαρμόζοντας τον αλγόριθμο του closing σε φωτεινά pixels με ένα συγκεκριμένο δομικό στοιχείο, αυτό ισοδυναμεί με το closing των pixels του φόντου με το ίδιο δομικό στοιχείο. Μια από τις χρήσεις της διαστολής είναι να γεμίζει μικρές τρύπες χρώματος στις εικόνες. Ένα πρόβλημα που προκύπτει από αυτή τη μέθοδο, είναι ότι η διαστολή διαταράσσει όλες τις περιοχές των pixel χωρίς διάκριση. Με την εφαρμογή συστολής στην εικόνα μετά από τη διαστολή (δηλαδή ο αλγόριθμος closing) μειώνεται κάπως αύτη η παρενέργεια. Μπορούμε να καταλάβουμε περίπου το αποτέλεσμα του closing, παίρνοντας το δομικό στοιχείο και μετακινώντας το εξωτερικά γύρω από κάθε φωτεινή (foreground) περιοχή, χωρίς να αλλάξουμε την κατεύθυνση του. Για οποιοδήποτε οριακό σημείο στο φόντο, αν το δομικό στοιχείο μπορεί να φτάσει αυτό το σημείο χωρίς κανένα κομμάτι του δομικού στοιχείου να βρίσκεται σε κάποια φωτεινή περιοχή, τότε αυτό το σημείο παραμένει ως φόντο. Αν κάτι τέτοιο δεν είναι δυνατό τότε το συγκεκριμένο pixel μπαίνει στη φωτεινή περιοχή.

Όταν η διεργασία του closing ολοκληρωθεί, η περιοχή του φόντου θα είναι έτσι σχηματισμένη ώστε το δομικό στοιχείο μπορεί να χρησιμοποιηθεί για να καλύψει οποιοδήποτε σημείο στο φόντο χωρίς κανένα κομμάτι του να καλύπτει παράλληλα και σημεία της φωτεινής περιοχής, και έτσι άλλες εφαρμογές του closing δεν θα έχουν κανένα αποτέλεσμα.



Εικόνα 25: Εφαρμογή του Closing με την χρήση ενός 3×3 τετράγωνου δομικού στοιχείου.

Όπως με την συστολή και με τη διαστολή, το συγκεκριμένο 3×3 δομικό στοιχείο είναι αυτό που χρησιμοποιείται συχνότερα, και είναι ήδη ενσωματωμένο στον κώδικα πολλών εφαρμογών, στις οποίες έτσι δεν είναι απαραίτητο να υπάρχει ένα ξεχωριστό δομικό εργαλείο. Για να επιτευχθεί το αποτέλεσμα ενός αλγόριθμου closing με ένα μεγαλύτερο δομικό στοιχείο, είναι εφικτό να γίνουν πολλαπλές διαστολές ακολουθούμενες από τον αντίστοιχο αριθμό συστολών.

Ο αλγόριθμος του closing μπορεί να χρησιμοποιηθεί για να γεμίσει μεμονωμένες περιοχές του φόντου μιας εικόνας. Αυτό μπορεί να επιτευχθεί αν βρεθεί το κατάλληλο δομικό στοιχείο το οποίο να ταιριάζει σε περιοχές της εικόνας που πρέπει να διατηρηθούν, αλλά δεν ταιριάζει σε περιοχές που πρέπει να αφαιρεθούν.

Εικόνα 26: Εικόνα πριν το closing

Η πιο πάνω εικόνα περιέχει μεγάλες και μικρές τρύπες. Αν θέλουμε να αφαιρέσουμε τις μικρές τρύπες διατηρώντας όμως παράλληλα τις μεγάλες τρύπες μπορούμε να εφαρμόσουμε ένα αλγόριθμο με ένα δισκοειδές δομικό στοιχείο το οποίο να έχει διάμετρο μεγαλύτερη από τις μικρές τρύπες αλλά μικρότερη από τις μεγάλες τρύπες.

Εικόνα 27: Εικόνα μετά το closing

Η δεύτερη εικόνα είναι το αποτέλεσμα ενός αλγόριθμου με ένα δίσκο διαμέτρου 22 pixel. Μαζί με τις μικρές τρύπες έχει γεμίσει παράλληλα και ο λεπτός λευκός κύκλος ως αποτέλεσμα της διεργασίας του closing. Σε εφαρμογές στον πραγματικό κόσμο ο αλγόριθμος του closing μπορεί να χρησιμοποιηθεί για να ενισχύσει δυαδικές εικόνες αντικειμένων που λαμβάνονται μέσα από την κατωφλίωση

ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Με τον όρο «αναγνώριση αντικειμένων» συνήθως αναφερόμαστε στον χαρακτηρισμό του αντικειμένου που αναγνωρίζετε ως μέλος κάποιας κλάσης ή κατηγορίας αντικειμένων με κάποια κοινά χαρακτηρίστηκα, πχ «Το αντικείμενο είναι ένα ποδήλατο». Η αναγνώριση μπορεί επίσης να αναφέρετε και στην ταυτοποίηση αντικειμένων ως συγκεκριμένες υποστάσεις, χωρίς απαραίτητα να γνωρίζει το σύστημα τα χαρακτηριστικά της κλάσης, όπως πχ «Το αντικείμενο της εικόνας Α είναι το ίδιο αντικείμενο με αυτό της εικόνας Β» ή «Το ποδήλατο του Γιώργου». Για να μπορεί ο υπολογιστής να περιγράψει μια εικόνα αποτελεσματικά θα πρέπει να εξαχθούν από την εικόνα χαρακτηρίστηκα που την κάνουν να διαφέρει από άλλες εικόνες.

ΕΞΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΕΙΚΟΝΑΣ

Υπάρχουν μέχρι στιγμής δύο κατηγορίες χαρακτηριστικών που μπορούν να περιγράψουν διανυσματικά ένα αντικείμενο. Τα εξωτερικά χαρακτηριστικά που δίνουν έμφαση στο σχήμα του αντικειμένου, όπως για παράδειγμα το περίγραμμα του και τα εσωτερικά χαρακτηριστικά που αξιοποιούν σημεία που εντοπίζονται ως κλειδιά στην εικόνα και έχουν να κάνουν με τη διαβάθμιση του εικονοστοιχείου σε σχέση με τα γειτονικά του. Συνήθως τα συστήματα τεχνητής όρασης επιστρατεύουν έναν από τους δύο τρόπους αναπαράστασης αλλά μερικές φορές γίνεται και συνδυασμός των δύο.

Για το ποια στοιχεία θα επιλεγούν ως σημεία κλειδιά για την αναπαράσταση των εικόνων με βάση τα εσωτερικά χαρακτηρίστηκα τους, καθώς και για την μορφή των διανυσμάτων θα αναφέρουμε επιγραμματικά δύο βασικούς αλγόριθμους που χρησιμοποιούνται ευρέως στη τεχνητή όραση, τον SIFT [26] (Scale Invariant Feature Transform) και τον SURF [27](Speeded Up Robust Features).

ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΗΝ ΕΞΑΓΩΓΗ ΕΣΩΤΕΡΙΚΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ.

ΑΛΓΟΡΙΘΜΟΣ SIFT

Ο SIFT αλγόριθμος είναι ίσως ο πιο δημοφιλής αλγόριθμος εξαγωγής χαρακτηριστικών που χρησιμοποιείται στην τεχνητή όραση. Το όνομα του προκύπτει από τα αρχικά Scale Invariant Feature Transform και έχει προταθεί από τον David Lowe το 1999. Μια δεύτερη ποιο εκτεταμένη εργασία έγινε από τον ίδιο το 2004.

Για κάθε αντικείμενο σε μια εικόνα μπορούν να εντοπιστούν κάποια σημεία ενδιαφέροντος τα οποία μπορούν να χρησιμοποιηθούν για να περιγράψουν το αντικείμενο με μαθηματικό τρόπο. Στην περίπτωση του αλγόριθμου SIFT αυτά τα σημεία εντοπίζονται κοντά στις γωνίες των αντικειμένων και η δουλεία του αλγόριθμου είναι αρχικά να εντοπίσει αυτά τα σημεία, να τους ορίσει μια κατεύθυνση και να δημιουργήσει μια μοναδική περιγραφή για το κάθε σημείο που θα είναι παρόμοια για παρόμοια σημεία αλλά και αρκετά διαφορετική για σημεία που δεν μοιάζουν. Ο SIFT δεν ξεκίνησε από το μηδέν, είναι η εξέλιξη του Harris corner detector [28] μόνο που αντί να χρησιμοποίει τον LoG [29] (Laplacian of Gaussian) τελεστή για τον εντοπισμό των σημείων ενδιαφέροντος, ο οποίος είναι υπολογιστικά ακριβός, χρησιμοποιεί μια προσέγγιση του LoG υπολογίζοντας τις DoG [30] (Difference of Gaussian) σε μια σειρά από μέγεθο-χωρικές [31] εικόνες που παράγονται από την αρχική εικόνα (Scale space representation).

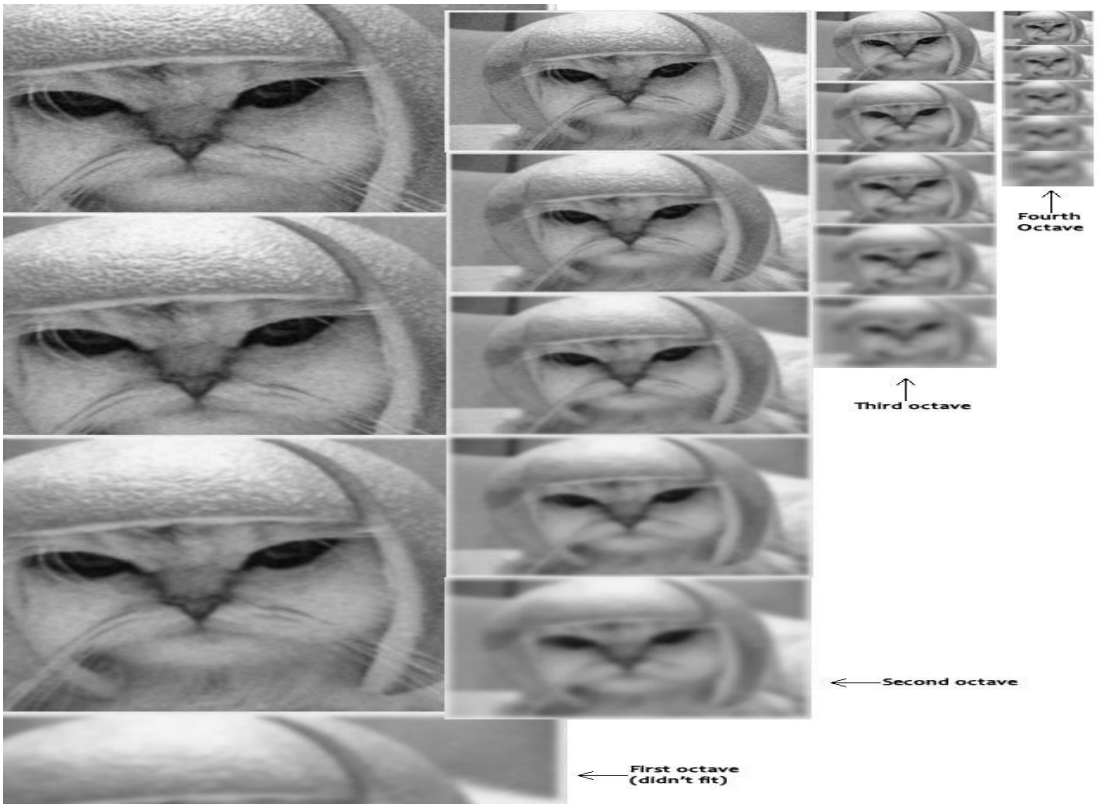
Παρόλο που η μέθοδος αυτή προσεγγίζει τον LoG έχει ορισμένα σοβαρά πλεονεκτήματα απέναντι του τα οποία είναι το κόστος υπολογισμού του, η αδιαφορία του σε σχέση με το μέγεθος (Scale Invariant) και αδιαφορία σε σχέση με την περιστροφή (Orientation Invariant). Επιπλέον, ο τρόπος με τον οποίο υπολογίζονται οι περιγραφείς στον SIFT δίνει και μια σχετική αδιαφορία στην φωτεινότητα (Illumination Invariant).

Αυτά τα χαρακτηριστικά αδιαφορίας ως προς το μέγεθος, την περιστροφή, την θέση και την φωτεινότητα του σημείου ενδιαφέροντος που εντοπίζει ο sift τον κάνουν να έχει καλά αποτελέσματα όταν χρησιμοποιείτε για να αναγνωρίσει αντικείμενα όπως αυτά του πραγματικού κόσμου, πράγμα στο οποίο αποτυγχάνουν μερικός ή ολικός οι αλγόριθμοι που υπήρχαν πριν από τον SIFT.

Τα βήματα που ακολουθεί ο SIFT για την εξαγωγή των περιγραφέων (Descriptors) εξηγούνται επιγραμματικά παρακάτω.

a) Δημιουργία Scale-Space αναπαράστασης της εικόνας.

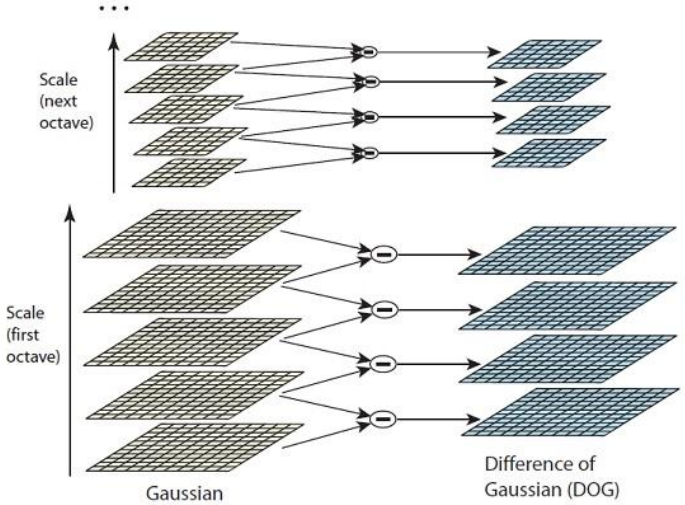
   Η αρχική προετοιμασία. Δημιουργούνται εσωτερικά αντίγραφα της εικόνας σε ομάδες (π.χ. 5 αντίγραφα ανά ομάδα) δημιουργούνται ομάδες τόσες ανάλογα με το μέγεθος της εικόνας. Κάθε ομάδα έχει το μισό μέγεθος από την προηγούμενη. Εσωτερικά μέσα σε κάθε ομάδα γίνεται ένα σταδιακό θόλωμα Gaussian. Αυτή η διαδικασία είναι απαραίτητη για να έχουμε scale invariance και είναι επίσης η βάση για το επόμενο βήμα. Η Εικόνα 28 δείχνει ένα παράδειγμα.



Εικόνα 28: Scale space αναπαράσταση εικόνας

b) Δημιουργία προσέγγισης του LoG

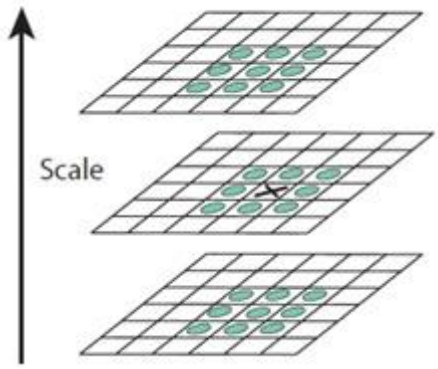
Σε αυτό το στάδιο υπολογίζετε ο LoG κατά προσέγγιση. Για να γίνει αυτό υπολογίζονται οι διαφορές των εικόνων scale space ανά 2. Επειδή το Gaussian φιλτράρισμα που έγινε στο scale space έχει σταδιακά αυξανόμενη ένταση προκύπτουν διαφορές σε κάθε εικόνα σε σχέση με την επόμενη. Αυτές οι διαφορές είναι περίπου αυτό που θα παίρναμε και με τον LoG τελεστή υπολογίζοντας δεύτερη παράγωγο άλλα με αυτό τον τρόπο (DoG) γίνεται απλά αφαίρεση.



Εικόνα 29: Υπολογισμός του LoG από τις διαφορές των Gaussian εικόνων σε όλες τις κλίμακες.

c) Εντοπισμός σημείων κλειδιών

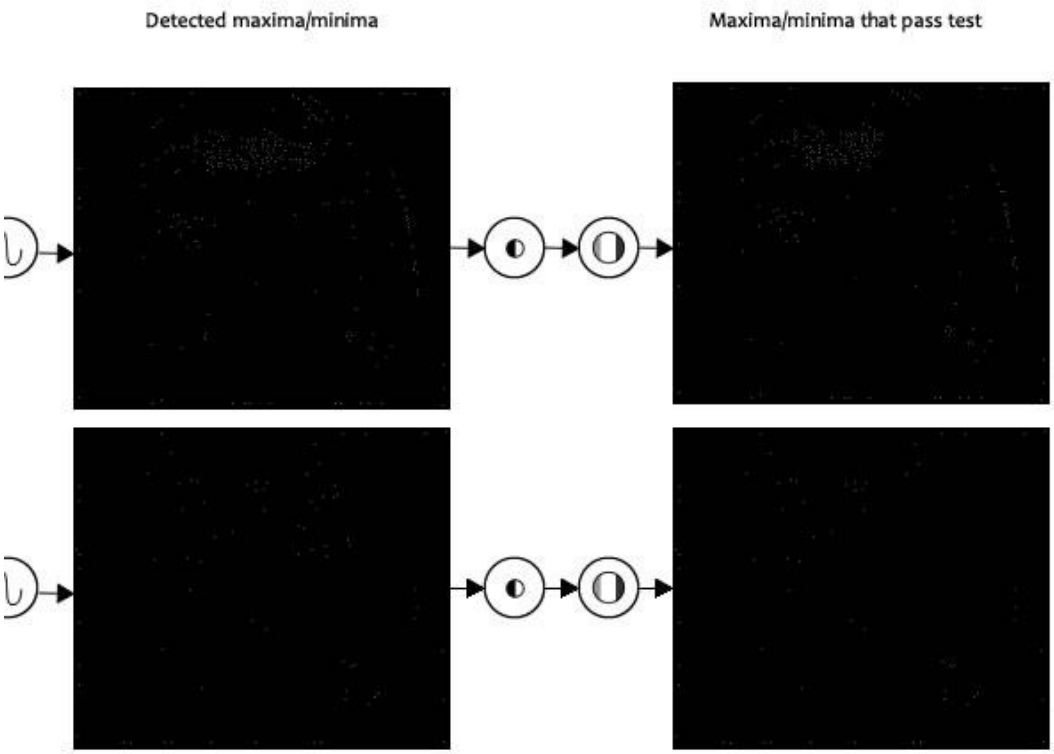
Για να εντοπιστούν τα σημεία κλειδιά βρίσκουμε τα ελάχιστα και τα μέγιστα στις εικόνες LoG. Για κάθε pixel εξετάζονται τα 8 γειτονικά του, όπως επίσης και 9 από την προηγούμενη και 9 από την επόμενη εικόνα LoG, δηλαδή 26 συγκρίσεις σε κάθε pixel. Αν βρεθεί σε αυτές τις συγκρίσεις ότι το συγκεκριμένο pixel έχει την μεγαλύτερη ή την μικρότερη τιμή της γειτονιάς του τότε το σημειώνεται ως σημείο κλειδί. Επειδή τα σημεία κλειδιά που εντοπίζονται ανάμεσα σε διαφορετικές κλίμακες δεν αντιστοιχούν σε κάποιο συγκεκριμένο pixel αλλά κάπου ενδιάμεσα, οι σειρές Taylor [32] μας βοηθούν να βρούμε τη θέση τους κατά προσέγγιση και την ονομάζουμε θέση subpixel.



Εικόνα 30: Εκτίμηση της θέσης του εικονοστοιχείου από τις εικόνες scale space

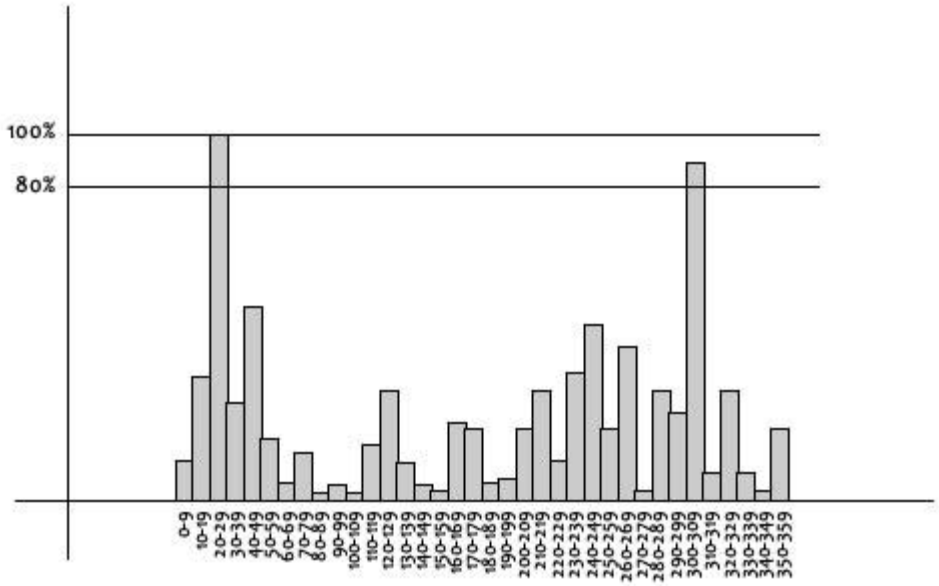
d) Απόρριψη σημείων κλειδιών που βρίσκονται σε περιοχές με μικρή αντίθεση ή σε ακμές

Τα σημεία κλειδιά που βρίσκονται πάνω σε ακμές ή σε περιοχές με χαμηλή αντίθεση δεν αποδίδουν χρήσιμα χαρακτηριστικά ενώ τα σημεία κλειδιά που ανήκουν σε γωνίες αντιθέτως είναι πολύ χρήσιμα. Για το λόγο αυτό γίνεται εντοπισμός των ακμών. Τα σημεία που ανήκουν σε γωνίες έχουν διαφορετικές διαβαθμίσεις σε σχέση με τα γειτονικά τους εικονοστοιχεία από ότι αυτά που ανήκουν σε επιφάνειες ή σε ακμές. Κρατούνται μόνο τα σημεία κλειδιά που βρίσκονται σε γωνίες, ενώ αυτά που βρίσκονται σε ευθείες ακμές ή σε επιφάνειες απορρίπτονται. Έτσι ο αριθμός των σημείων κλειδιών μειώνεται και η περιγραφή της εικόνας γίνεται ποιο απλή αλλά και ποιο ακριβείς.



Εικόνα 31: Φιλτράρισμα και απόρριψη σημείων που δεν ανήκουν σε γωνίες

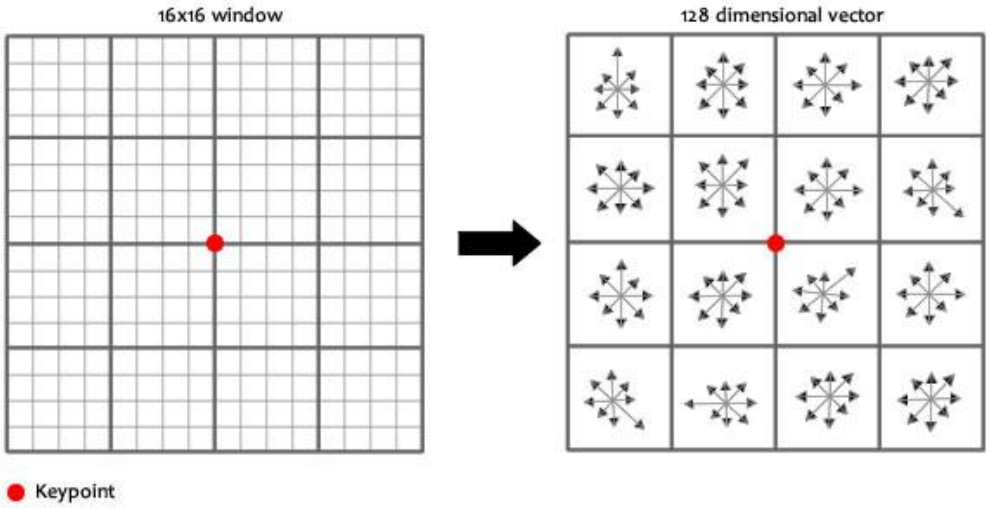
e) Προσδιορισμός κατεύθυνσης σημείων κλειδιών
Γίνεται υπολογισμός των διαβαθμίσεων σε μια περιοχή γύρω από το σημείο κλειδί και οι κατευθύνσεις των διαβαθμίσεων τοποθετούνται σε ένα ιστόγραμμα από 0^o μέχρι 360^o σε ομάδες των 10^o. Η μεγαλύτερη κορυφή του ιστογράμματος δίνει στο σημείο κλειδί την κατεύθυνση του. Αν υπάρχει και δεύτερη κορυφή με πλάτος μεγαλύτερο από 80% της πρώτης αυτή δημιουργεί και ένα δεύτερο σημείο κλειδί στο ίδιο σημείο αλλά με διαφορετική φορά.



Εικόνα 32: Ιστόγραμμα για υπολογισμό της κατεύθυνσης για το σημείο-κλειδί

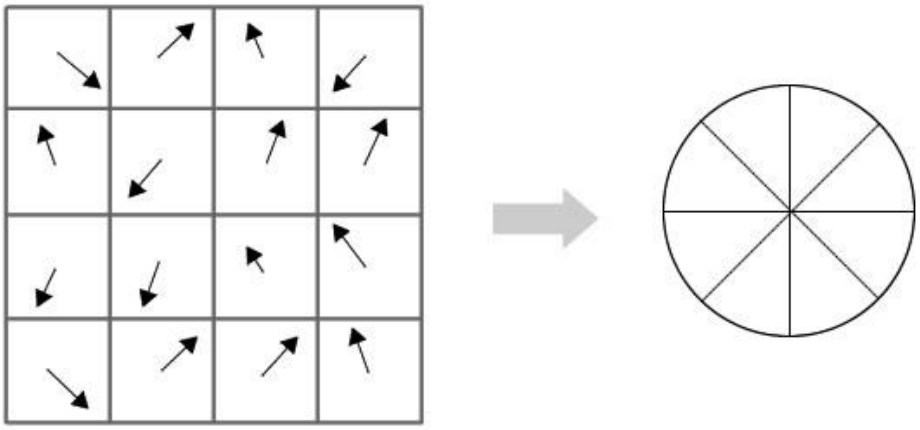
f) Υπολογισμός περιγραφέα για το κάθε σημείο κλειδί

Κάθε σημείο κλειδί αντιπροσωπεύει ένα χαρακτηριστικό που ο αλγόριθμος θεωρεί σημαντικό. Είναι προφανές λοιπόν ότι για να έχουν νόημα τα σημεία κλειδιά δεν είναι αρκετή μόνο η θέση και η κατεύθυνση που προσδιορίζει αλλά η πληροφορία της εικόνας σε εκείνη την περιοχή. Για αυτό είναι απαραίτητο το κάθε σημείο κλειδί να περιγραφεί με τρόπο ώστε να ξεχωρίζει από τα υπόλοιπα σημεία όταν περιγράφουν ένα διαφορετικό χαρακτηριστικό και να μοιάζει με σημεία όταν τα χαρακτηριστικά μοιάζουν. Για παράδειγμα ο περιγραφέας για ένα αντικείμενο όπως το μολύβι θα μοιάζει με τον περιγραφέα ενός άλλου μολυβιού, όμως θα διαφέρει σημαντικά από ένα περιγραφέα που υπολογίστηκε για ένα μαχαίρι. Για να υπολογιστούν οι περιγραφείς λαμβάνονται υπόψιν οι διαβαθμίσεις της εικόνας σε μια περιοχή 16*16 εικονοστοιχείων γύρω από το σημείο κλειδί. Αρχικά η 16*16 περιοχή χωρίζετε σε 16 υποπεριοχές που η κάθε μία έχει διαστάσεις 4*4 εικονοστοιχεία.



Εικόνα 33: Υπολογισμός των διαβαθμίσεων της περιοχής γύρο από το σημείο κλειδί

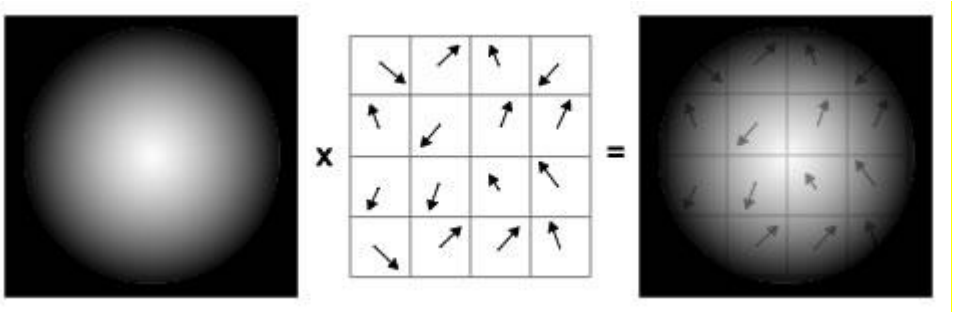
Στην συνέχεια υπολογίζονται το μέτρο και η διεύθυνση των διαβαθμίσεων για κάθε εικονοστοιχείο, και οι τιμές αυτές τοποθετούνται σε ένα ιστόγραμμα με 8 θέσεις.



Εικόνα 34: Υπολογισμός μέτρου και διεύθυνσης ομάδας

Κάθε κατεύθυνση διαβάθμισης στο εύρος 0-44 μοίρες τοποθετείται στην πρώτη θέση του ιστογράμματος. Από 45-89 μοίρες στη δεύτερη θέση του ιστογράμματος και ούτω καθεξής. Το ύψος το στηλών του ιστογράμματος εξαρτάται από το μέτρο της διαβάθμισης.

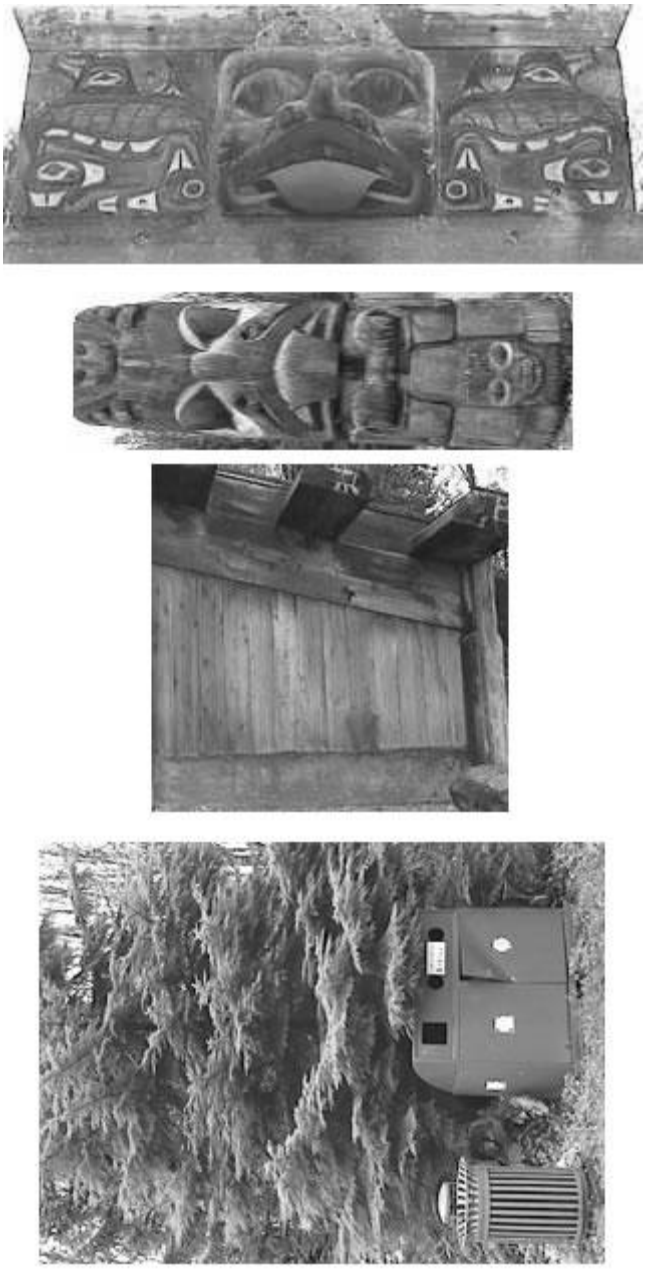
Εκτός όμως από αυτό και σε αντίθεση με την διαδικασία που ακολουθείται για την εύρεση της κατεύθυνσης για τα σημεία κλειδιά που ακολουθείτε σε προηγούμενο βήμα, τώρα το ποσοστό συμμετοχής του μέτρου διαβάθμισης στο ιστόγραμμα εξαρτάται και από την απόσταση του εικονοστοιχείου από το σημείο κλειδί. Κατ' αυτό το τρόπο διαβαθμίσεις που απέχουν μεγαλύτερη απόσταση από το σημείο κλειδί θα έχουν μικρότερο αντίκτυπο στο αποτέλεσμα του ιστογράμματος. Αυτό γίνεται με τη χρήση μιας Γκαουσιανής συνάρτησης βαρών. Τα μέτρα που υπολογίστηκαν πολλαπλασιάζονται με την συνάρτηση αυτή και έτσι αλλάζει το μέγεθος τους όπως φαίνεται και στην εικόνα 35.



Εικόνα 35: Πολλαπλασιασμός των μέτρων με την Γκαουσιανή συνάρτηση

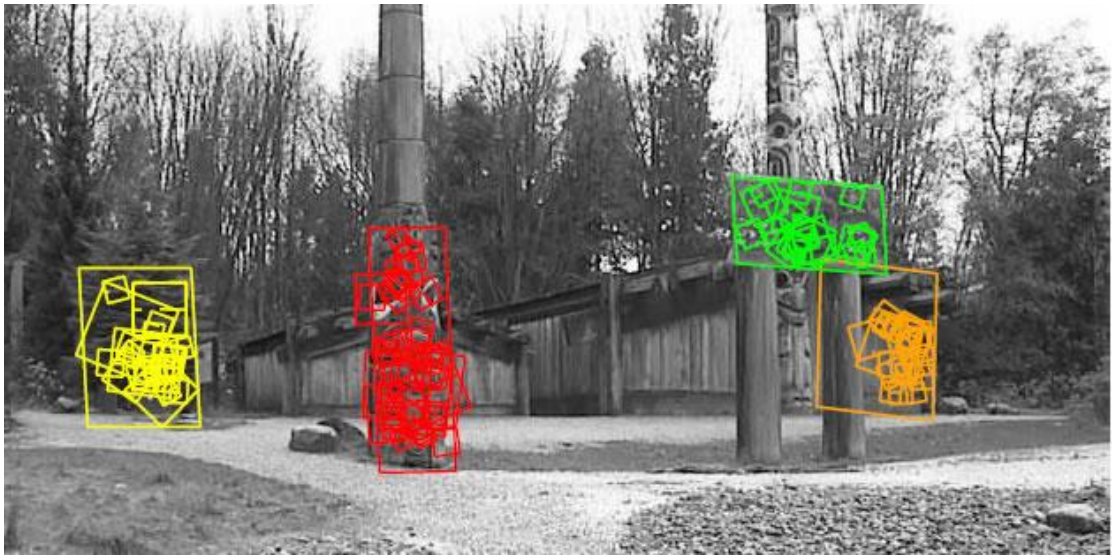
Κάνοντας αυτό για όλα τα 16 εικονοστοιχεία το αποτέλεσμα είναι 16 τυχαίες κατευθύνσεις χωρισμένες σε 8 προκαθορισμένες ομάδες. Στη συνέχεια γίνεται το ίδιο και για τις υπόλοιπες δεκαέξι 4×4 περιοχές. Τελικά έχουμε 4x4x8 = 128 αριθμούς, τους οποίους κανονικοποιούμε . Αυτοί οι 128 αριθμοί αποτελούν ένα διάνυσμα χαρακτηριστικών. Αυτό το σημείο κλειδί αναγνωρίζεται με αυτό το διάνυσμα χαρακτηριστικών.

Αυτό το διάνυσμα χαρακτηριστικών χρειάζεται ακόμα δύο διορθώσεις. Η πρώτη έχει να κάνει με την περιστροφή. Επειδή το διάνυσμα χαρακτηριστικών χρησιμοποιεί τις περιστροφές των διαβαθμίσεων, οποιαδήποτε περιστροφή στην εικόνα θα τον καταστήσει άχρηστο. Για να είναι ο περιγραφέας αμετάβλητος ως προς την περιστροφή , η περιστροφή του κάθε σημείου κλειδιού αφαιρείτε από τον τις τιμές του διανύσματος. Επίσης χρησιμοποιώντας κατωφλίωση σε αριθμούς που είναι μεγάλοι πετυχαίνουμε αδιαφορία ως προς τον φωτισμό. Για αυτό κάθε αριθμός του διανύσματος που έχει τιμή μεγαλύτερη από 0.2 αλλάζει σε 0.2. Αυτό έχει σαν αποτέλεσμα το διάνυσμα να κανονικοποιείται και πάλι. Έτσι το διάνυσμα είναι ανεξάρτητο από περιστροφή και φωτεινότητα.
Οι επόμενες εικόνες δείχνουν ορισμένα παραδείγματα για τη χρήση του SIFT σε εικόνες [33].

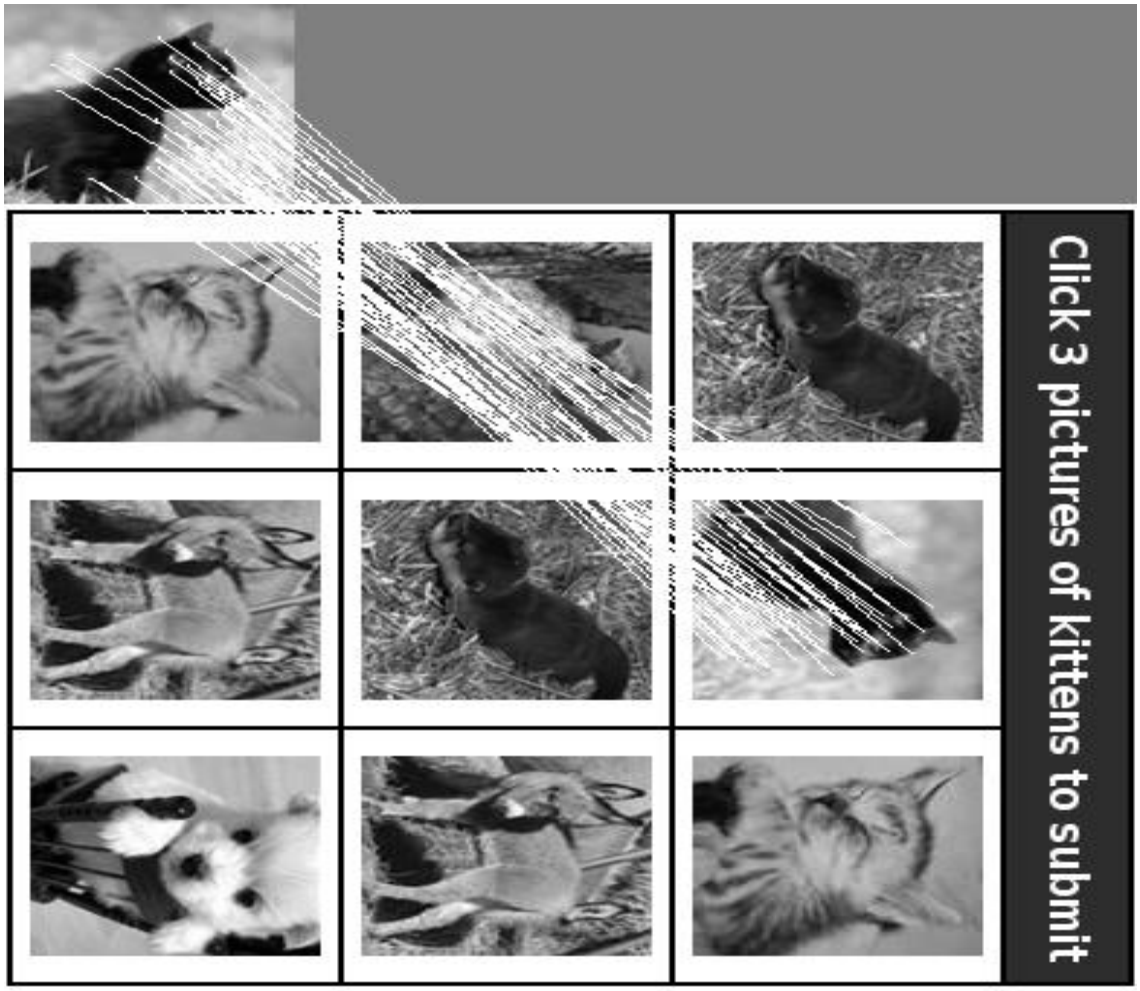


Εικόνα 36: Αντικείμενα που πρέπει να εντοπιστούν.

Εικόνα 37: εικόνα που μπορεί να περιέχει τα αντικείμενα που ψάχνουμε.



Εικόνα 38: Εντοπισμός με τη χρήση των SIFT περιγραφέων.



Εικόνα 39: Άλλο παράδειγμα χρήσης του SIFT για ταυτοποίηση αντικειμένων.

Ο ΑΛΓΟΡΙΘΜΟΣ SURF

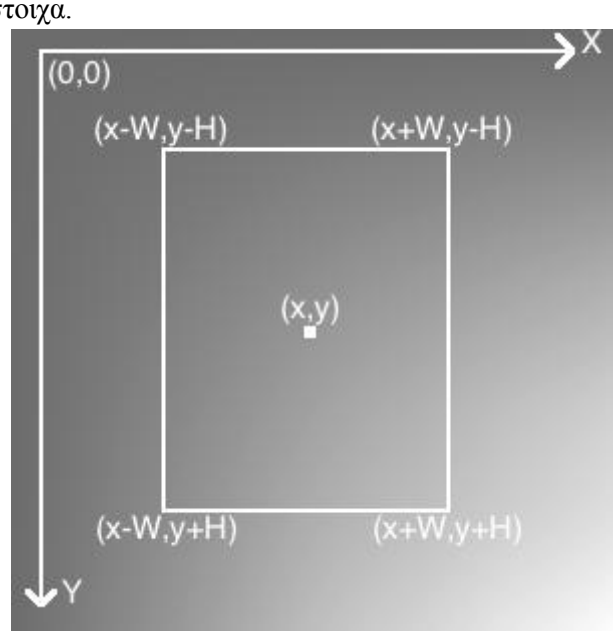
Ο αλγόριθμος SURF [34] (Speeded Up Robust Features) είναι ένας γρήγορος και εύρωστος αλγόριθμος για την αναπαράσταση εικόνων και την αναγνώριση τους χρησιμοποιώντας τοπικά χαρακτηριστικά της εικόνας. Η προσέγγιση του SURF στο πρόβλημα διαφέρει από αυτή του SIFT σε 2 σημεία που τον καθιστούν αρκετά ταχύτερο, κάνοντας τον έτσι καλύτερη επιλογή για πραγματικού χρόνου εφαρμογές αναγνώρισης αντικειμένων και παρακολούθησης. Τα αποτελέσματα του SURF όσον αφορά το ποσοστό επιτυχίας ταύτισης σε σχέση με αυτά του SIFT είναι παρόμοια, αν και οι συγγραφείς ισχυρίζονται ότι ο SURF πετυχαίνει καλύτερα αποτελέσματα από τον SIFT.
Ο SURF βασίζεται στη διδακτορική διατριβή του H. Bay και σε μια εργασία που συνέταξε ο ίδιος μαζί με τους A. Ess, T. Tuytelaars και L. Van Gool.

Ο SURF αποτελείται από τρία βασικά βήματα:

A) Ανίχνευση των σημείων ενδιαφέροντος.
B) Περιγραφή των σημείων ενδιαφέροντος.
Γ) Ταύτιση χαρακτηριστικών.

Όπως στον SIFT, έτσι και στον SURF τα δύο πρώτα βήματα στηρίζονται σε μια scale-space αναπαράσταση της εικόνας και σε διαφορικές παραγώγους πρώτης και δεύτερης τάξης. Αυτό που κάνει τον SURF να διαφέρει είναι ότι αυτές οι διαδικασίες επιταχύνονται κάνοντας

χρήση της εικόνας ολοκλήρωσης (integral image [35]) και τεχνικών φιλτραρίσματος κουτιού [36] (box filter technics) για την προσομοίωση του scale space και την επιλογή των σημείων ενδιαφέροντος αντίστοιχα.



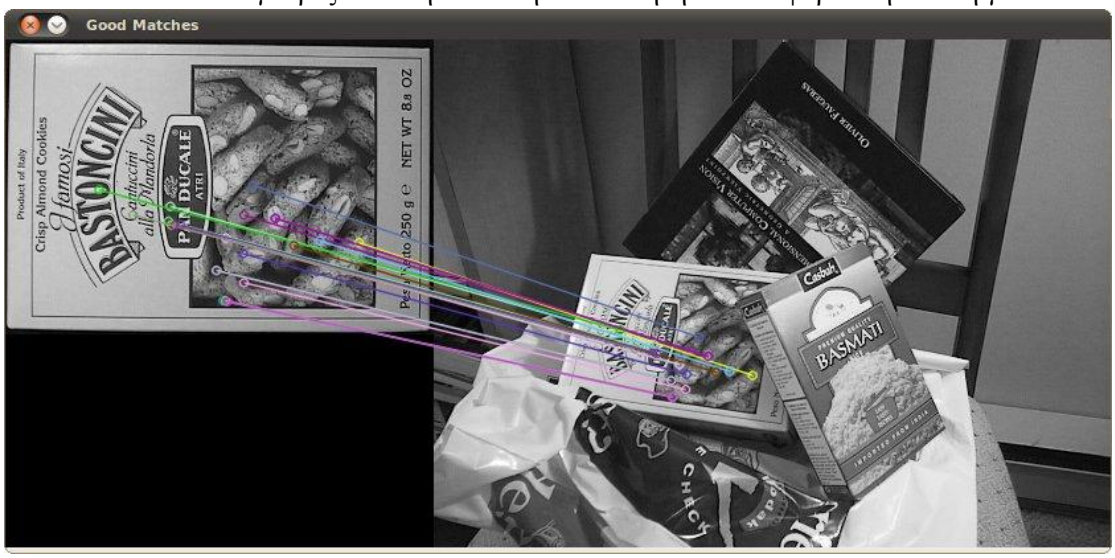
Εικόνα 40: Παράδειγμα εικόνας ολοκλήρωσης

Στο στάδιο της ανίχνευσης, υπολογίζονται τα τοπικά μέγιστα του καθοριστικού τελεστή Hessian [37] (Hessian determinant operator) και εφαρμόζονται στην scale-space αναπαράσταση για την επιλογή των υποψήφιων σημείων ενδιαφέροντος. Αυτά τα υποψήφια σημεία επικυρώνονται αν η ανταπόκριση είναι μεγαλύτερη από ένα συγκεκριμένο κατώφλι. Η κλίμακα όπως και η θέση αυτών των υποψηφίων στην συνέχεια φιλτράρονται χρησιμοποιώντας μια επαναλαμβανόμενη διαδικασία για να τοποθετηθούν σε μια εξίσωση $2^{ου}$ βαθμού. Τυπικά μερικές εκατοντάδες τέτοιων σημείων ενδιαφέροντος εντοπίζονται σε μια ψηφιακή εικόνα μεγέθους ενός Mega-pixel.

Εικόνα 41: Σημεία ενδιαφέροντος που εντοπίζονται

Ο σκοπός του δεύτερου βήματος είναι να κατασκευαστεί ένας περιγραφέας που να είναι αδιάφορος σε αλλαγές οπτικής γωνίας σε κάποιο βαθμό. Η θέση αυτού του σημείου στο scale-space δίνει αδιαφορία σε αλλαγές της κλίμακας και της τοποθέτησης του σημείου. Για να επιτευχθεί και αδιαφορία ως προς την περιστροφή ορίζετε μία κυρίαρχη κατεύθυνση παίρνοντας υπόψιν την τοπική κατανομή διαβαθμίσεων, υπολογισμένη με Haar wavelets [38]. Επιλέγοντας μια περιοχή γύρο από το σημείο κατασκευάζετε ένας περιγραφέας 64bit που αντιστοιχεί στο ιστόγραμμα της Haar wavelet ανταπόκρισης.

Στην κλασσική περίπτωση το τρίτο βήμα ταυτίζει τους περιγραφείς δύο εικόνων. Γίνονται όλες οι συγκρίσεις υπολογίζοντας την ευκλείδεια απόσταση ανάμεσα σε όλα τα πιθανά ζευγάρια περιγραφέων. Ένα κριτήριο ταύτισης κοντινότερου γείτονα-αναλογίας απόστασης χρησιμοποιείται στη συνέχεια για να περιορίσει τις εσφαλμένες ταυτίσεις, σε συνδυασμό με μια βασισμένη σε μια RANSAC τεχνική για τον έλεγχο της γεωμετρικής εγκυρότητας. Μετά το πέρασμα από τα παραπάνω φίλτρα οι ταυτίσεις που παραμένουν υποτίθεται ότι είναι έγκυρες και ανήκουν στην ίδια σκηνή από διαφορετική οπτική γωνία.



Εικόνα 42: Ταύτιση χαρακτηριστικών SURF

ΑΝΑΓΝΩΡΙΣΗ

Η αναγνώριση στην γενική περίπτωση έχει ως στόχο την ταύτιση των περιγραφέων ανάμεσα σε 2 εικόνες, για να αποφανθεί αν οι δύο εικόνες απεικονίζουν το ίδιο αντικείμενο ή σκηνή. Μια άλλη πιθανή χρήση της αναγνώρισης είναι η παρακολούθηση ενός αντικειμένου (tracking) που κινείται σε ένα βίντεο. Το μεγαλύτερο πρόβλημα που καλείται όμως να λύσει η αναγνώριση είναι η κατηγοριοποίηση ενός ή περισσότερων αντικειμένων σε γενικές κλάσεις. Αυτή η διαδικασία είναι ιδιαίτερα δύσκολη λόγω της αφθονίας των διαφορετικών κλάσεων που υπάρχουν στον πραγματικό κόσμο και κατά συνέπεια και στις εικόνες. Η αναπαράσταση των εικόνων με περιγραφείς όπως είδαμε παραπάνω μπορεί να είναι ικανοποιητική όταν το ζητούμενο είναι η αναγνώριση μιας συγκεκριμένης υπόστασης αντικειμένου σε 2 εικόνες, αλλά όταν πρόκειται για κατηγοριοποίηση τα πράγματα δεν είναι τόσο απλά. Δύο αντικείμενα μπορεί να ανήκουν στην ίδια κλάση αλλά να διαφέρουν σημαντικά οι περιγραφείς τους.

Παρακάτω παρουσιάζονται μερικοί από τους σημαντικότερους αλγόριθμους για την ταύτιση και αναγνώριση αντικειμένων.

Ο K-NEAREST NEIGHBOR

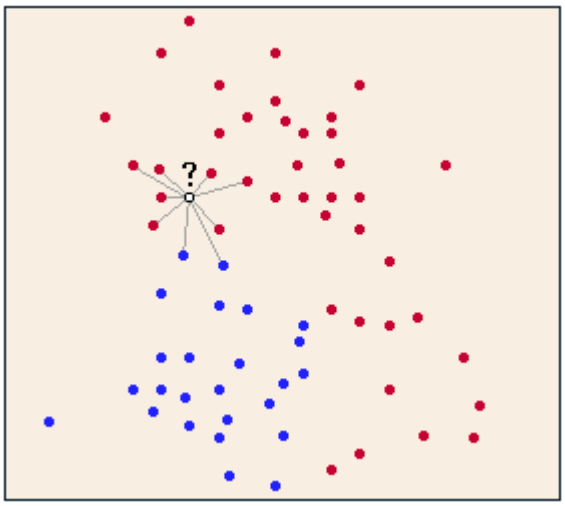
Ο αλγόριθμος k Nearest Neighbor [39] (ή kNN) είναι ένας από τους ποίο απλούς αλγόριθμους κατηγοριοποίησης αντικειμένων που χρησιμοποιείται μεταξύ άλλων και στη τεχνητή όραση για την αναγνώριση αντικειμένων. Συνήθως είναι η πρώτη επιλογή για συγκρίσεις γιατί έχει πολύ εύκολη υλοποίηση. Στην opencv υπάρχει ήδη βιβλιοθήκη για τον αλγόριθμο όπως επίσης και στο Matlab.

Παρά την απλότητα του ο αλγόριθμος μπορεί να είναι αρκετά αποτελεσματικός σε συγκεκριμένες εφαρμογές. Στην αναγνώριση χαρακτήρων ο αλγόριθμος χρησιμοποιείται ευρέως.

Η βασική αρχή λειτουργίας του είναι η εξής:

Έχουμε ένα σύνολο διανυσμάτων το οποίο χρησιμοποιείται για την εκπαίδευση του αλγόριθμου. Αυτό δεν είναι τίποτε άλλο από το να υπολογιστούν η αποστάσεις του κάθε διανύσματος από τα υπόλοιπα και να αποθηκευτούν στη μνήμη. Στη συνέχεια μπαίνει στο σύστημα το διάνυσμα, έστω το Β που πρέπει να κατηγοριοποιηθεί. Θεωρώντας ότι τα υπόλοιπα διανύσματα ανήκουν σε γνωστά αντικείμενα, ψάχνουμε για τα k(Ακέραιος) διανύσματα που απέχουν μικρότερη απόσταση από το διάνυσμα Β.

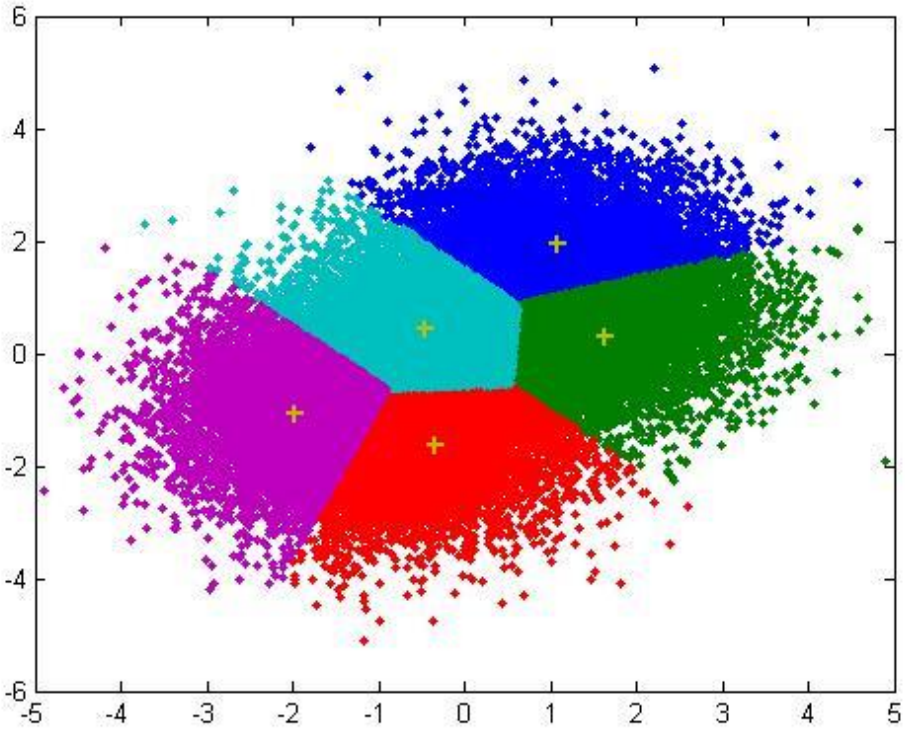
Για να επιτευχθεί αυτό υπολογίζετε η ευκλείδεια απόσταση του Β από όλα τα διανύσματα και επιλέγονται τα k με την μικρότερη τιμή και σημειώνονται οι κατηγορίες στις οποίες ανήκουν. Αυτή η διαδικασία μοιάζει με εκλογές, όπου το κάθε ένα διάνυσμα από τα k κοντινότερα που εντοπίστηκαν δίνει μία ψήφο για την κατηγορία που θα πρέπει να καταταχθεί το Β. Κάθε ένα από αυτά θα ψηφήσει το Β να καταταχτεί στην δική του κατηγορία ή αλλιώς κλάση, και η κλάση που θα συγκεντρώσει τους περισσότερους ψήφους θα είναι η κλάση στην οποία θα καταταχθεί το Β.



Εικόνα 43: Ο αλγόριθμος k-Nearest Neighbor με k=9, για 2 κλάσεις.

Μια καλύτερη αποδοτικά προσέγγιση του αλγόριθμου για περιπτώσεις όπου υπάρχουν πολλά διανύσματα για ένα αρκετά μικρότερο αριθμό κατηγοριών , είναι να υπολογιστεί για κάθε ομάδα διανυσμάτων που ανήκει στην ίδια κατηγορία ένα διάνυσμα του οποίου το άθροισμα

των αποστάσεων από όλα τα διανύσματα της ομάδας του θα είναι το ελάχιστο δυνατό. Στη συνέχεια αυτό το διάνυσμα θεωρείτε το διάνυσμα για αυτή την κατηγορία και περιορίζετε έτσι δραματικά ο αριθμός των συγκρίσεων χωρίς να επηρεαστεί σημαντικά η έξοδος.



Εικόνα 44 : Πολλά διανύσματα που ανήκουν σε 5 κατηγορίες αντιπροσωπεύονται τελικά από τα κέντρα τους. (k-means clustering [40])

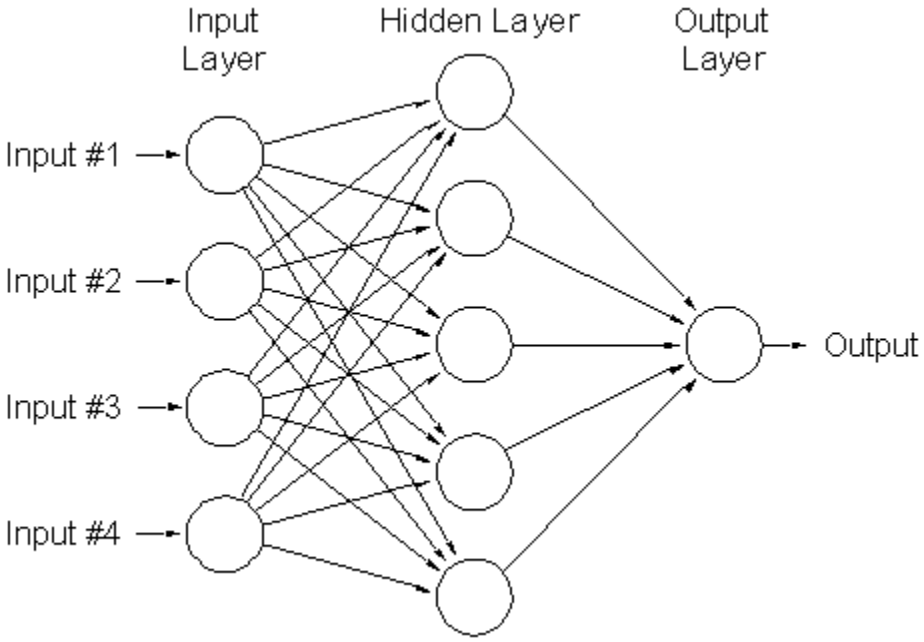
ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

Με την εξέλιξη της επιστήμης των υπολογιστών έγινε γρήγορα φανερό ότι οι υπολογιστές έχουν την δυνατότητα να εκτελούν πολύπλοκες μαθηματικές πράξεις πολύ ποιο γρήγορα από τον ανθρώπινο εγκέφαλο. Από την άλλη όμως, οι υπολογιστές αδυνατούν να αποφασίσουν αποτελεσματικά για προβλήματα τα οποία δεν μοντελοποιούνται μαθηματικά και κατά συνέπεια δεν προγραμματίζονται με συμβατικό προγραμματισμό. Σε τέτοιου είδους προβλήματα ο ανθρώπινος εγκέφαλος ανταποκρίνεται ταχύτατα, και αυτή η παρατήρηση ενέπνευσε την δημιουργία των τεχνητών νευρωνικών δικτύων.

Τα τεχνητά νευρωνικά δίκτυα [41] έχουν σχεδιαστεί για να μιμηθούν τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Ο ανθρώπινος εγκέφαλος περιέχει περίπου 100 δισεκατομμύρια νευρώνες, κάθε ένας από τους οποίους συνδέετε περίπου με 10000 άλλους νευρώνες, ένα εξαιρετικά πολύπλοκο νευρωνικό δίκτυο που είναι αδύνατο να υλοποιηθεί με τη σημερινή τεχνολογία. Απλούστερες όμως εκδοχές που περιέχουν από ένα μέχρι μερικές χιλιάδες νευρώνες χρησιμοποιούνται αποτελεσματικά για να λύσουν συγκεκριμένα προβλήματα αποφάσεων.

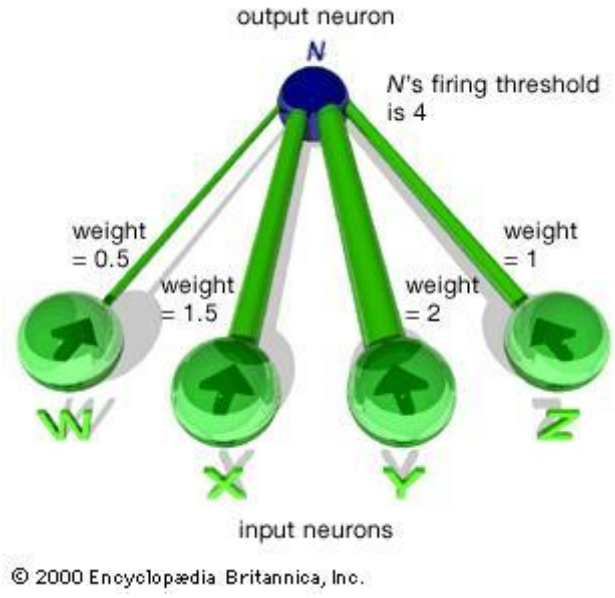
Υπάρχουν πολλές διαφορετικές αρχιτεκτονικές τεχνητών νευρωνικών δικτύων. Η αρχιτεκτονική είναι ο τρόπος που οι τεχνητοί νευρώνες είναι συνδεδεμένοι μεταξύ τους. Κάθε αρχιτεκτονική είναι αποτελεσματική για συγκεκριμένα προβλήματα.

Ο κάθε νευρώνας ενός τεχνητού νευρωνικού δικτύου έχει μία ή περισσότερες εισόδους και μία ή περισσότερες εξόδους. Οι νευρώνες είναι οργανωμένοι σε επίπεδα με τέτοιο τρόπο ώστε οι έξοδοι ενός επιπέδου να αποτελούν τις εισόδους για το επόμενο επίπεδο, μέχρι το τελευταίο επίπεδο που αποτελεί την τελική έξοδο.



Εικόνα 45: Παράδειγμα επιπέδων Τεχνητού Νευρωνικού δικτύου

Κάθε είσοδος και έξοδος σε ένα τεχνητό νευρώνα πολλαπλασιάζεται με ένα βάρος w_i. Έτσι ρυθμίζοντας τα βάρη των νευρώνων μπορούμε να επιτύχουμε την επιθυμητή έξοδο του δικτύου για μια συγκεκριμένη είσοδο του δικτύου.



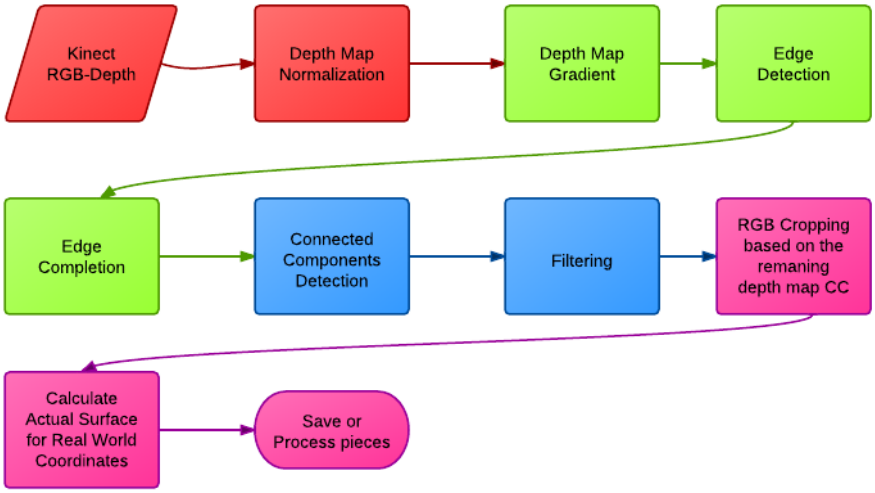
Εικόνα 46: Τα βάρη επηρεάζουν την έξοδο των νευρώνων εισόδου

Στην πράξη η ρύθμιση των βαρών δεν γίνεται χειροκίνητα αλλά αυτόματα κατά την διάρκεια εκπαίδευσης του συστήματος. Η διαδικασία της εκπαίδευσης είναι διαφορετική για διαφορετικούς τύπους και αρχιτεκτονικές νευρωνικών δικτύων. Μια γενική περίπτωση για κατηγοριοποίηση περιγράφεται παρακάτω.

Αρχικά όλα τα βάρη έχουν την ίδια τιμή , πχ 1. Δίνονται στις εισόδους περιγραφείς αντικειμένων των οποίων η σωστή έξοδος (κατηγορία στην οποία ανήκουν τα αντικείμενα) είναι γνωστή και το σύστημα προσπαθεί με πολλαπλές επαναλήψεις να ρυθμίσει τα βάρη έτσι ώστε η έξοδος να πλησιάζει την πραγματική. Αν το δίκτυο είναι πολύ μικρό, πιθανόν να μην μπορεί να προσαρμοστεί για ένα μεγάλο αριθμό περιγραφέων, αφού κάθε περιγραφέας που μπαίνει στο δίκτυο για την εκπαίδευση θα χαλάει την έξοδο για τους υπολοίπους. Αν το δίκτυο είναι αρκετά μεγαλύτερο από το απαιτούμενο υπάρχει ο κίνδυνος μετά από αρκετές επαναλήψεις να προσαρμοστεί πλήρως στα δεδομένα εισόδου και να αποτυγχάνει να αναγνωρίσει παρόμοια αντικείμενα.

ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ

Το προτεινόμενο σύστημα χωρίζεται σε 2 τμήματα, το πρώτο τμήμα είναι αυτό του εντοπισμού και υπολογισμού του πραγματικού μεγέθους αντικειμένων από πολύπλοκες σκηνές με πολλά αντικείμενα. Το δεύτερο τμήμα είναι αυτό της εκπαίδευσης ενός υποσυστήματος για την αναγνώριση των αντικειμένων και η αναγνώριση τους με την χρήση του συστήματος αυτού. Πρώτα θα παρουσιάσουμε το κομμάτι του εντοπισμού το οποίο χωρίζετε σε τέσσερεις βασικές ενότητες όπως δείχνει η Εικόνα 1 (με διαφορετικά χρώματα). Κάθε ενότητα αναλύεται σε βήματα που αναπαρίστανται σαν κουτιά στο ίδιο διάγραμμα.



Εικόνα 47: Διάγραμμα ροής του τμήματος εντοπισμού και υπολογισμού του πραγματικού μεγέθους των αντικειμένων για το προτεινόμενο σύστημα.

ΛΗΨΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΠΡΟ-ΕΠΕΞΕΡΓΑΣΙΑ (ΚΟΚΚΙΝΗ ΕΝΟΤΗΤΑ)

ΛΗΨΗ RGB-DEPTH ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ KINECT

Το πρώτο βήμα είναι να πάρουμε από το Kinect μια RGB εικόνα και μια εικόνα βάθους (χάρτης βάθους) όπως αυτές της εικόνας 48. Τα δεδομένα αυτά εισάγονται στο Matlab με τη μορφή πινάκων 640 × 480 × 3 για την RGB εικόνα. Η κάθε διάσταση για τον RGB πίνακα αντιστοιχεί σε μια χρωματική συνιστώσα (Κόκκινο, Πράσινο και Μπλε), και κάθε μία από αυτές έχει 640 στήλες και 480 γραμμές. Οπότε το κάθε κελί του πίνακα στις θέσεις x,y,1 αντιπροσωπεύει την τιμή της φωτεινότητας για το κόκκινο χρώμα η x,y,2 για το Πράσινο και x,y,3 για το Μπλε, όπου x,y είναι οι θέσεις του πίνακα για τις 2 πρώτες διαστάσεις του πίνακα. Όπως γίνεται αντιληπτό, κάθε θέση x,y του πινάκα αντιστοιχεί σε ένα εικονοστοιχείο (pixel) της RGB εικόνας και για κάθε εικονοστοιχείο ο συνδυασμός των φωτεινοτήτων των τριών χρωματικών συνιστωσών μας δίνει τελικά το χρώμα του κάθε εικονοστοιχείου.

Κάθε συνιστώσα για κάθε εικονοσοιχείου μπορεί να παίρνει τιμή από 0 μέχρι 255 (Αυτό προκύπτει από τη μετατροπή του byte = 8 bit => 2^8=256 διαφορετικοί συνδυασμοί). Άρα για κάθε εικονοστοιχείο έχουμε 256 × 256 × 256=16777216 διαφορετικούς συνδυασμούς των χρωματικών συνιστωσών και κάθε συνδυασμός αντιστοιχεί σε ένα διαφορετικό χρώμα για το εικονοστοιχείο. Ο πίνακας βάθους (ή χάρτης βάθους όπως συνηθίζεται) μπορούμε να φανταστούμε ότι είναι μια τέταρτη διάσταση του RGB πίνακα

όπου το κάθε κελί στη θέση x,y αντιστοιχεί την απόσταση του αντίστοιχου εικονοστοιχείου από τον αισθητήρα βάθους.

Κάθε x,y κελί του πίνακα βάθους μπορεί να παίρνει τιμές από 0 μέχρι 8000 και οι τιμές αυτές αντιστοιχούν σε μιλίμετρα. Στην πραγματικότητα είναι ποιο βολικό να αντιμετωπίζουμε τον πίνακα βάθους σαν ένα ξεχωριστό πίνακα. Μια οπτική αναπαράσταση του πίνακα βάθους μπορεί να προκύψει αν αντιστοιχίσουμε τις τιμές βάθους σε διαφορετικά χρώματα (όπως κάναμε στην εικόνα 48). Για να μπορεί το Matlab να πάρει αυτά τα δεδομένα από το Kinect θα πρέπει πρώτα να μπορεί να επικοινωνεί με τη συσκευή.

Αυτή δεν είναι μια εύκολη διαδικασία γιατί το Kinect έχει κατασκευαστεί (σχεδιάστηκε και προγραμματίστηκε) για να επικοινωνεί με την παιχνιδομηχανή Xbox360 (κονσόλα Xbox360), οπότε το πρώτο πρόβλημα είναι η επικοινωνία της με το λειτουργικό σύστημα windows 7. Ευτυχώς σε πολύ σύντομο χρονικό διάστημα μετά την κυκλοφορία του Kinect διάφορες ομάδες ανθρώπων άρχισαν να εκδίδουν προγράμματα οδήγησης (driver) για το Kinect σε διάφορα λειτουργικά συστήματα για PC και MAC. Επιλέξαμε να χρησιμοποιήσουμε το πακέτο προγραμμάτων οδήγησης OpenNI [42] (Πρόγραμμα οδήγησης για το ASUS Xtion pro, συμβατό και με το Kinect) γιατί ήτανε το πρώτο (και το μοναδικό την στιγμή της υλοποίησης) που μπορούσε να μας δώσει την λύση στο δεύτερο πρόβλημα που αντιμετωπίσαμε, την επικοινωνία Matlab-Kinect Driver. Η αιτία του προβλήματος ήταν ότι όλα τα υπόλοιπα προγράμματα οδήγησης Kinect λειτουργούσαν είχαν προσβάσιμο προγραμματιστικό πακέτο μόνο για τις ποιο ευρέως διαδεδομένες γλώσσες προγραμματισμού όπως την C, C++ .

Το Matlab δεν χρησιμοποιεί μεταγλωττιστή (compiler) όπως ή C αλλά διερμηνέα (interpreter). Το πρόβλημα λοιπόν ήταν ότι έπρεπε να γραφτούν συναρτήσεις Matlab οι οποίες θα επέτρεπαν στο Matlab να χρησιμοποιεί τις βιβλιοθήκες του προγραμματιστικού πακέτου που είναι εκτελέσιμα αρχεία. Τέτοιες συναρτήσεις ονομάζονται MEX (Matlab Executable) και πρέπει να γραφτούν αρχικά σε κάποια γλώσσα προγραμματισμού που χρησιμοποιεί compiler (συνήθως C++) και στη συνέχεια να ορίσουμε στις ρυθμίσεις του Matlab να το μονοπάτι του compiler (εμείς χρησιμοποιήσαμε το Microsoft Visual Studio 2010 με C++) και τα αρχεία πηγαίου κώδικα που θέλουμε να μετατρέψουμε σε MEX αρχεία.

Ευτυχώς και πάλι κάποιος [43] είχε ήδη υλοποιήσει τα αρχεία πηγαίου κώδικα για σύνδεση Matlab-OpenNI οπότε το μόνο που χρειάστηκε ήταν να μεταγλωττίσουμε αυτά τα αρχεία σε συναρτήσεις MEX. Μετά από αυτό το κρίσιμο βήμα αποκτήσαμε την δυνατότητα να καλούμε κατευθείαν μέσα από το Matlab τις MEX συναρτήσεις και να έχει τελικά το Matlab πρόσβαση στα δεδομένα που επιστρέφει η συσκευή Kinect. Αξίζει να σημειωθεί ότι οι MEX συναρτήσεις λειτούργησαν σωστά μόνο στην 32bit έκδοση του Matlab.

Εικόνα 48: RGB εικόνα και εικόνα βάθους

ΚΑΝΟΝΙΚΟΠΟΙΗΣΗ ΤΟΥ ΧΑΡΤΗ ΒΑΘΟΥΣ

Η ‘ακατέργαστη’ εικόνα βάθους που παίρνουμε από το Kinect περιέχει κάποιες περιοχές που δεν περιέχουν καθόλου πληροφορία για το βάθος αυτής της περιοχής. Αυτό οφείλεται στο γεγονός ότι η υπέρυθρη ακτινοβολία δεν αντανακλά το ίδιο καλά σε όλες τις επιφάνειες. Επίσης στο γεγονός ότι το Kinect (για Xbox360) έχει σχεδιαστεί για να υπολογίζει αποστάσεις από 60cm μέχρι και μερικά μέτρα. Όποια αντικείμενα βρίσκονται εκτός αυτού εύρους παίρνουν τιμή βάθους 0. Αυτές οι περιοχές θα παράξουν λάθος αποτελέσματα αν δεν διορθωθούν.

Ένα πρόβλημα που αντιμετωπίσαμε είναι ότι η όποια διόρθωση του χάρτη βάθους θα έπρεπε να γίνει σε πολύ ένα πολύ μικρό χρονικό περιθώριο (να είχε δηλαδή μικρό υπολογιστικό κόστος) για να μην δημιουργήσει καθυστέρηση και να μπορεί το σύστημα να εξακολουθεί να λειτουργεί σε πραγματικό χρόνο. Να θυμηθούμε λίγο ότι το Kinect μας δίνει τη δυνατότητα να παίρνουμε από τη συσκευή ζευγάρια δεδομένων RGB και depth με ρυθμό 30 στιγμιότυπων ανά δευτερόλεπτο (30fps).

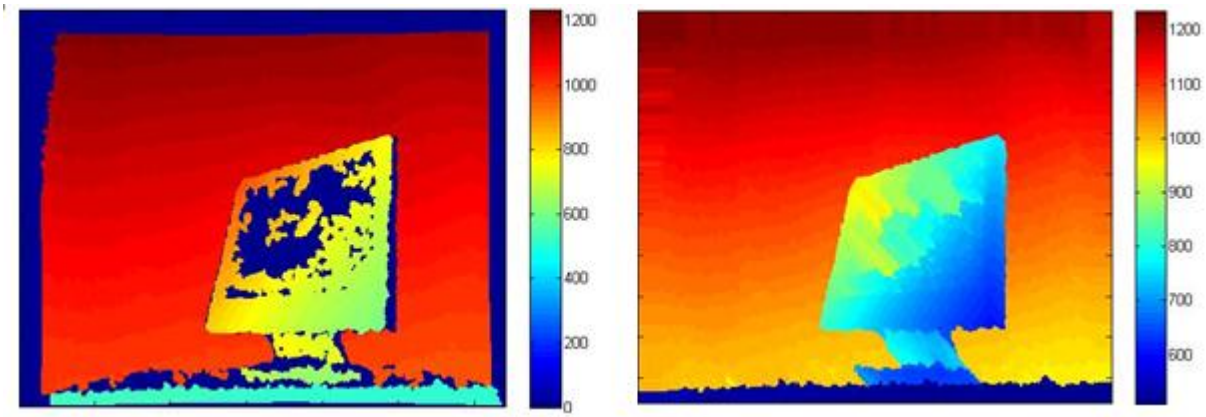
Το ιδανικό σύστημα πραγματικού χρόνου θα έπρεπε λοιπόν να μπορεί να παράγει αποτελέσματα με τον ίδιο ρυθμό. Κάτι τέτοιο θα σήμαινε ότι το κάθε στιγμιότυπο θα πρέπει να φτάσει στην από την είσοδο στην έξοδο σε χρόνο μικρότερο από 1/30 του δευτερόλεπτου. Στην πραγματικότητα κάτι τέτοιο δεν ήταν εφικτό καθώς το Matlab λόγω του ότι λειτουργεί με διερμηνέα και κάθε εντολή πρέπει να περνάει από αυτόν πριν προχωρήσει στον επεξεργαστή. Επίσης το Matlab δεν έχει δυνατότητα χρήσης της κάρτας γραφικών του υπολογιστή η οποία αποδίδει πολύ καλύτερα όταν πρόκειται για επεξεργασία εικόνας. Με αυτά τα δεδομένα στη διάθεση μας έπρεπε να κάνουμε την όποια επεξεργασία στο συντομότερο δυνατό χρονικό διάστημα.

Όταν αναφερόμαστε στην επεξεργασία ενός πίνακα 640 × 480, δηλαδή ενός πίνακα 307200 στοιχείων γίνετε αμέσως αντιληπτό ότι και μόνο η πράξη του ελέγχου που θα πρέπει να γίνει για να διαπιστωθεί αν κάποιο στοιχείο περιέχει μηδενική τιμή είναι μια διαδικασία η οποία στοιχίζει σημαντικά σε υπολογιστική ισχύ. Έπρεπε λοιπόν να δημιουργηθεί μια συνάρτηση που θα δεχόταν ως είσοδο της την αρχική εικόνα βάθους και θα επέστρεφε μια εικόνα βάθους χωρίς μηδενικές τιμές. Το δεύτερο πρόβλημα που αντιμετωπίσαμε ήταν και το ποίο κρίσιμο. Με τι θα αντικαθιστούσαμε τις μηδενικές τιμές όταν εντοπίζονταν; Στην ουσία μια μηδενική τιμή υποδεικνύει ότι δεν έχουμε πληροφορία απόστασης του συγκεκριμένου εικονοστοιχείου από τον αισθητήρα.

Ο ιδανικός συνδυασμός ταχύτητας-ποιότητας ανάκτησης της χαμένης πληροφορίας αποτελεί μεγάλο πρόβλημα στην ψηφιακή επεξεργασία εικόνας και είναι ένα πρόβλημα που διαρκώς επιδέχεται βελτιώσεις. Στην περίπτωση μας η τεχνική που ακολουθήσαμε ήταν να αντικαταστήσουμε κάθε μηδενική τιμή με μια κοντινή μη μηδενική τιμή. Εδώ υπήρξε μια ανταλλαγή ανάμεσα στην ποιότητα του αποτελέσματος και στον χρόνο υπολογισμού του, αφού για να βρούμε την κοντινότερη μηδενική τιμή θα πρέπει να υπολογίζουμε την ευκλείδεια απόσταση του μηδενικού εικονοστοιχείου με τα υπόλοιπα μη μηδενικά εικονοστοιχεία και μετά να συγκρίνουμε τις αποστάσεις και να αποφασίσουμε ποια απόσταση είναι η μικρότερη για να αντικαταστήσουμε τελικά την μηδενική τιμή.

Αυτό θα έπρεπε να γίνει για κάθε μηδενική τιμή. Αντί λοιπόν της κοντινότερης δυνατής τιμής και για λόγους εξοικονόμησης επεξεργαστικής ισχύος αρκεστήκαμε στο να βρούμε απλά μια κοντινή τιμή προχωρώντας με ένα μη γραμμικά αυξανόμενο βήμα για την εξέταση των γειτονικών εικονοστοιχείων και επιλέγοντάς το πρώτο μη μηδενικό στοιχείο ως το κατάλληλο για να γίνει η αντικατάσταση. Με αυτό τον τρόπο και ρυθμίζοντας τις παραμέτρους του βήματος καταφέραμε να έχουμε ένα ποιοτικά ανεκτό αποτέλεσμα σε ένα

χρονικά ανεκτό υπολογιστικό κόστος. Δημιουργήσαμε μια συνάρτηση η οποία αντικαθιστά τις μηδενικές τιμές με τις κοντινές μη μηδενικές τιμές (Εικόνα 3), και έτσι λύσαμε αυτό το πρόβλημα.



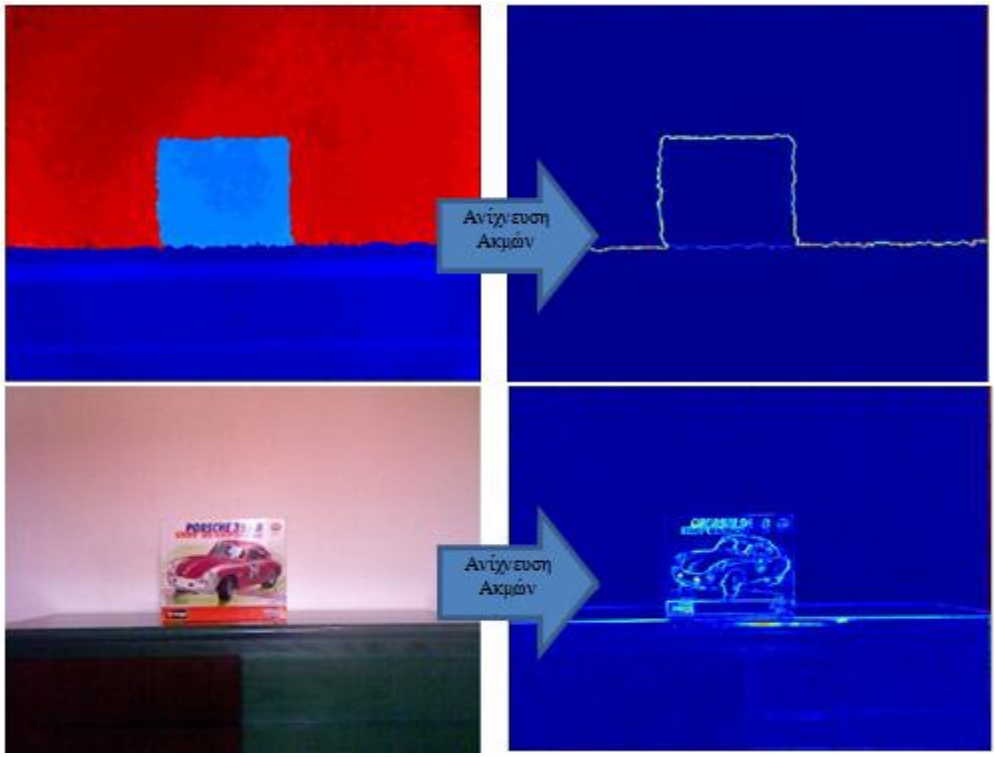
Εικόνα 49: Εικόνα βάθους πριν και μετά την κανονικοποίηση. Παρατηρήστε ότι στην δεύτερη εικόνα δεν υπάρχουν μηδενικές τιμές.

ΑΝΑΛΥΣΗ ΤΟΥ ΧΑΡΤΗ ΒΑΘΟΥΣ (ΠΡΑΣΙΝΗ ΕΝΟΤΗΤΑ)

ΑΝΙΧΝΕΥΣΗ ΑΚΜΩΝ ΕΙΚΟΝΑΣ ΒΑΘΟΥΣ

Η κανονικοποιημένη εικόνα βάθους δεν περιέχει πλέον μηδενικές τιμές, αλλά για να μπορεί να έχουμε ένα καλό αποτέλεσμα στον εντοπισμό τον αντικειμένων θα πρέπει να γίνει κάποια περεταίρω επεξεργασία της εικόνας βάθους. Ο απώτερος σκοπός αυτής της διαδικασίας είναι να προκύψει μία εικόνα η οποία θα επισημάνει τελικά τα όρια των αντικειμένων. Επειδή η κανονικοποιημένη εικόνα βάθους θα μας χρειαστεί στη μορφή που είναι και για άλλες διεργασίες του συστήματος όπως αυτό του φιλτραρίσματος για τον διαχωρισμό των έγκυρων αντικειμένων, και στο στάδιο του υπολογισμού του πραγματικού μεγέθους οι ακόλουθες διεργασίες δεν γίνονται στην κανονικοποιημένη εικόνα αλλά σε ένα αντίγραφο αυτής το οποίο αποθηκεύεται σε ξεχωριστό χώρο στη μνήμη (σε μια άλλη μεταβλητή). Η τεχνική που ακολουθούμε για την ανίχνευση ακμών στην εικόνα βάθους βασίζεται στον αλγόριθμο που πρότεινε ο Canny [20] για ανίχνευση ακμών σε εικόνες RGB. Ποιο συγκεκριμένα, υλοποιούμε 2 πίνακες που απορρέουν από τις διαβαθμίσεις (gradient) ενός δυσδιάστατου γκαουσιανού (Gaussian) φίλτρου μεγέθους 10X10 και με τυπική απόκλιση (Standard deviation, Sigma) 0.5 εικόνα.

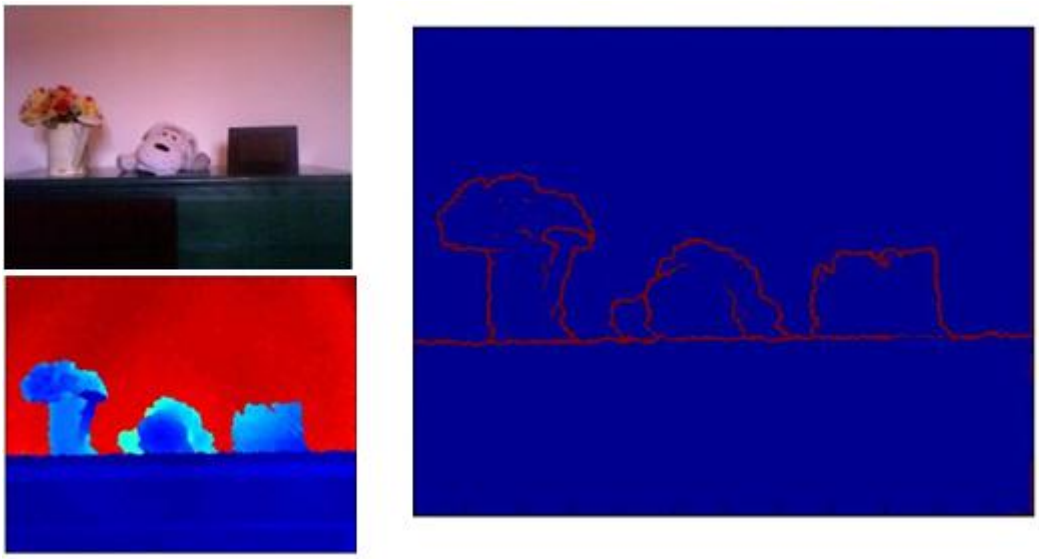
Οι δύο πίνακες που προκύπτουν είναι οι διαβαθμίσεις του γκαουσιανού φίλτρου ως προς τον άξονες x και y εικόνες. Στην συνέχεια υπολογίζουμε την συνέλιξη των δύο πινάκων με την εικόνα και αθροίζουμε τα αποτελέσματα τύπος. Αυτό έχει σαν αποτέλεσμα να παραχθεί μια εικόνα όπου οι ακμές των αντικειμένων έχουν μεγάλες τιμές ενώ όλα η υπόλοιπη περιοχή έχει τιμές σχεδόν μηδενικές (Εικόνα 50). Οι τιμές των ακμών μπορεί να είναι μεγάλες ή μικρές ανάλογα με το με το μέγεθος της αλλαγής του βάθους (μέτρο ακμής). Έτσι μπορούμε να πούμε ότι έχουμε δυνατές και αδύνατες ακμές αντίστοιχα. Είναι δυνατόν να εντοπίσουμε κατ' αυτό τον τρόπο ακμές και στην RGB εικόνα αλλά αφού οι ακμές θα εμφανίζονταν όποτε είχαμε κάποια απότομη αλλαγή στην φωτεινότητα κάποιου χρώματος δεν θα αντιπροσώπευαν απαραίτητα τα όρια ενός αντικειμένου αλλά και αλλαγές στο χρώμα μέσα στο ίδιο το αντικείμενο. Αντίθετα με αυτά οι εικόνες βάθους που μας επιστρέφει το Kinect είναι ανεξάρτητες της φωτεινότητας.



Εικόνα 50: Οι ακμές στην εικόνα βάθους αποδίδουν καλύτερα τα όρια του αντικειμένου απ' ότι οι ακμές στην RGB εικόνα

ΚΑΤΩΦΛΙΩΣΗ ΑΚΜΩΝ

Αφού οι ακμές έχουν γίνει ορατές είναι απαραίτητο να παράγουμε μια δυαδική εικόνα όπως στην εικόνα 5, όπου μόνο οι ακμές θα έχουν τιμή ένα και όλα τα υπόλοιπα τιμή μηδέν. Σε αυτό το σημείο είναι απαραίτητο να ορίσουμε ένα κατώφλι ευαισθησίας. Αν χρησιμοποιήσουμε πολύ υψηλό κατώφλι θα έχει ως αποτέλεσμα να χάσουμε σημαντικό μέρος των ακμών, ενώ αν το κατώφλι είναι πολύ χαμηλό, πολλές ασήμαντες ακμές θα καταστήσουν την εικόνα άχρηστη. Η απλή κατωφλίωση δεν είναι όμως αρκετή για την σωστή οριοθέτηση των αντικειμένων. Οι ακμές ενός αντικειμένου μπορεί να είναι ισχυρές σε κάποιο σημείο και ασθενείς σε κάποιο άλλο σημείο, για αυτό το λόγο χρησιμοποιούμε το κατώφλι, το οποίο θα αποδώσει τιμή ένα στα ισχυρά σημεία των ακμών και το ελάχιστο κατώφλι που θα αποδώσει τιμή ένα στο εικονοστοιχείο της δυαδικής εικόνας μόνο όταν αυτό συνορεύει με κάποια ισχυρή ακμή. Με αυτό τον τρόπο οι αδύναμες ακμές αγνοούνται μόνο αν είναι αδύναμες σε όλο τους το μήκος, ενώ τα αδύναμα τμήματα μιας δυνατής ακμής ανιχνεύονται κανονικά. Εμπειρικά μπορούμε να ρυθμίσουμε το σύστημα μας να ρυθμίζει αυτόματα τα κατώφλια ανιχνεύσεις ακμών. Αυτό προκύπτει από την παρατήρηση ότι διαφορετικά κατώφλια έχουν βέλτιστα αποτελέσματα ανάλογα με την απόσταση του αισθητήρα από την σκηνή. Η απόσταση του αισθητήρα στης περιπτώσεις που αυτός είναι αισθητήρας βάθους όπως το Kinect είναι γνωστή για κάθε pixel και μπορεί να χρησιμοποιήσουμε τη μέση ή τη μέγιστη τιμή με κάποιο εμπειρικό τύπο για να υπολογίσουμε αυτόματα τα κατώφλια ανίχνευσης ακμών.



Εικόνα 51: Ακμές που έχουν εντοπιστεί

ΣΥΜΠΛΗΡΩΣΗ ΑΚΜΩΝ

Οι ακμές εντοπίστηκαν ελέγχοντας τη τιμή του κάθε εικονοστοιχείου και αποδίδοντας του τιμή ένα ή μηδέν σύμφωνα με τον αλγόριθμο που περιγράψαμε στη προηγούμενη ενότητα. Σαν αποτέλεσμα η δυαδική εικόνα που παράγεται έχει στη μεγαλύτερη επιφάνεια τιμή μηδέν και στις δυνατές ακμές ή στις αδύναμες που συνορεύουν με δυνατές τιμή ένα. Στην εικόνα που προέκυψε μπορεί να λείπουν μικρά κομμάτια από ακμές, ή να εμφανίζονται μικρές περιοχές οι οποίες θεωρήθηκαν ακμές αλλά καταλαμβάνουν μόνο ένα ή και μερικά εικονοστοιχεία.

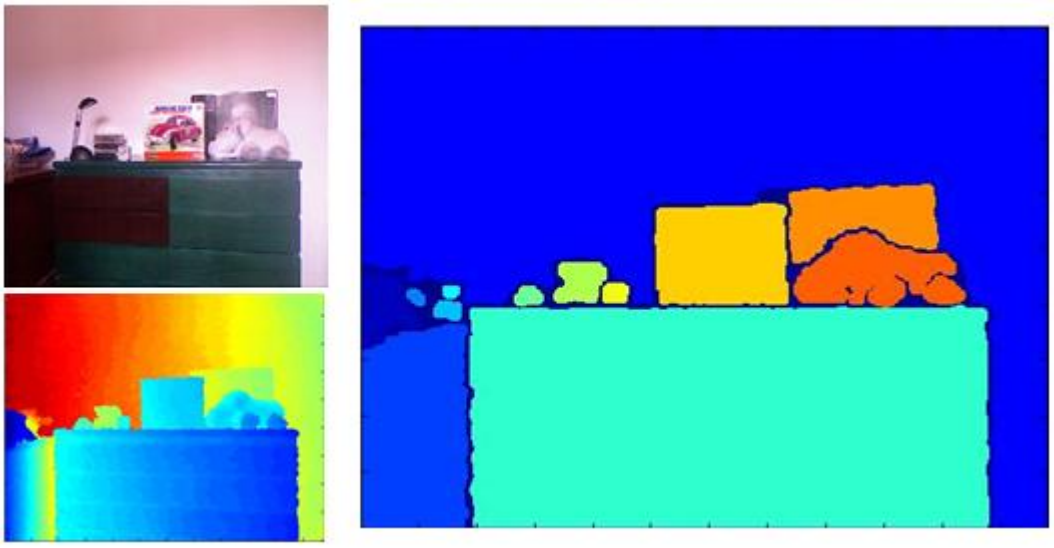
Για να γίνει σωστά ο διαχωρισμός των αντικειμένων με τον αλγόριθμο συνδεδεμένων στοιχείων [44] (Connected components algorithm) θα πρέπει τα όρια των αντικειμένων να είναι συμπαγή και να ξεχωρίζουν τα διαφορετικά αντικείμενα. Για αυτό χρησιμοποιούμε ένα μορφολογικό αλγόριθμο κλεισίματος [45](morphological closing) που με τη χρήση ενός δομικού στοιχείου συρρικνώνει (erode) τις ακμές όπου μοιάζουν στο δομικό στοιχείο και στη συνέχει τις διευρύνει (dilate). Αυτό έχει σαν αποτέλεσμα σημεία που είναι σχεδόν συνδεδεμένα να συνδεθούν μεταξύ τους ενώ μεμονωμένα τμήματα ακμών που είναι μικρότερα από το δομικό στοιχείο εξαφανίζονται. Επίσης αντιστρέφουμε όλες τις τιμές έτσι ώστε οι επιφάνειες να έχουν τιμή ένα και οι ακμές τιμή μηδέν (Εικόνα 6). Η εικόνα βάθους είναι τώρα έτοιμη για τον αλγόριθμο Connected Components [44] του Matlab.

Εικόνα 52: Συμπληρωμένες με closing ακμές

ΕΝΤΟΠΙΣΜΟΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΚΑΙ ΦΙΛΤΡΑΡΙΣΜΑ (ΜΠΛΕ ΕΝΟΤΗΤΑ)

ΑΛΓΟΡΙΘΜΟΣ CONNECTED COMPONENTS

Ο αλγόριθμος connected components αφού εφαρμοστεί στην επεξεργασμένη εικόνα βάθους θα επιστρέψει τις συντεταγμένες των εικονοστοιχείων κάθε κλειστής περιοχής και μαζί με τη συνάρτηση regionprops [46] του Matlab θα παράγουν χρήσιμα δεδομένα για κάθε περιοχή όπως το εμβαδόν σε εικονοστοιχεία, τα ακραία σημεία της περιοχής, συντεταγμένες του ορθογωνίου που περιβάλει την κάθε περιοχή κλπ. Η εικόνα 7 δείχνει τις συνδεδεμένες περιοχές που βρέθηκαν σε μια εικόνα. Τα τρία επόμενα βήματα είναι προαιρετικά και χρησιμοποιούνται για να αποκλείσουν αντικείμενα που πιθανόν να μην μας ενδιαφέρουν και δεν θέλουμε να τα χρησιμοποιήσουμε για την κατάτμηση της εικόνας.



Εικόνα 53: Συνδεδεμένα στοιχεία γεμισμένα με διαφορετικά χρώματα

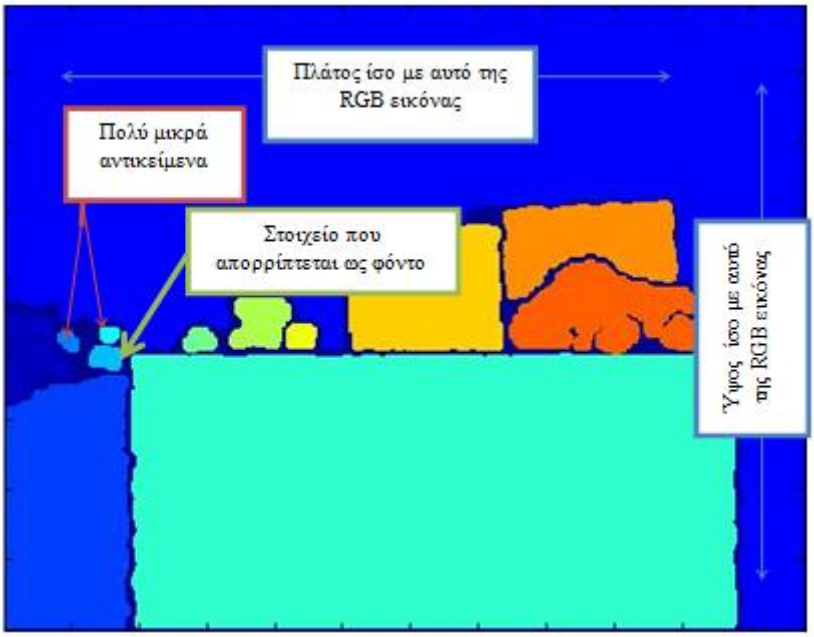
ΦΙΛΤΡΑΡΙΣΜΑ

Στο σύστημα μας υλοποιούμε τρείς φάσεις φιλτραρίσματος. Η πρώτη φάση αποκλείει αντικείμενα με βάση το σχετικό τους μέγεθος, η δεύτερη φάση φιλτράρει με βάση το ύψος και το πλάτος σε εικονοστοιχεία και η τρίτη φάση αφαιρεί τα στοιχεία που ανήκουν στο φόντο της εικόνας. Κάθε μία από τις τρείς φάσεις μπορεί να τροποποιηθεί ή ακόμη και να παραληφθεί αναλόγως με τις ανάγκες του χρήστη ή της εφαρμογής. Εξηγούμε παρακάτω πως λειτουργεί η κάθε φάση στο σύστημα μας. Στην πρώτη φάση ελέγχουμε για κομμάτια της

εικόνας που το εμβαδόν τους σε εικονοστοιχεία ανήκει στο εύρος μεταξύ της ελάχιστης και της μέγιστης επιτρεπτής επιφάνειας.

Η μέγιστη και η ελάχιστη επιτρεπτή επιφάνεια ορίζονται εμπειρικά ανάλογα με τις ανάγκες του συστήματος με τη μορφή του ποσοστού % της ολικής επιφάνειας της εικόνας. Για παράδειγμα αν ένα αντικείμενο είναι μεγαλύτερο του 50% της ολικής επιφάνειας ή μικρότερο του 1% δεν θα περάσει στην επόμενη φάση. Φυσικά οι τιμές αυτές μπορούν να οριστούν και σαν αριθμός εικονοστοιχείων για την ελάχιστη και τη μέγιστη περιοχή . Στη δεύτερη φάση μόνο τα αντικείμενα που δεν εκτείνονται σε όλο το ύψος ή όλο το πλάτος της εικόνας θα προωθηθούν στην επόμενη φάση. Τέλος στη Τρίτη φάση θα αποκλειστούν όλα τα αντικείμενα που θεωρούνται ότι ανήκουν στο φόντο της εικόνας αφήνοντας να περάσουν μόνο αντικείμενα που δεν ανήκουν στο φόντο.

Ένα αντικείμενο θεωρείται να ότι ανήκει στο φόντο αν ένα ποσοστό της επιφάνειας του (το οποίο το ορίζει ο χρήστης) ανήκει στο φόντο το οποίο επίσης ορίζετε εκ των προτέρων σαν μια περιοχή με τιμές βάθους κοντά στις μέγιστη τιμή βάθους του χάρτη βάθους. Η Εικόνα 54 δείχνει ένα παράδειγμα με στοιχεία που δεν θα περνούσαν (θα αποκόπτονταν) από τα φίλτρα. Τα στοιχεία που απομένουν και οι ιδιότητες τους θα χρησιμοποιηθούν στην επόμενη ενότητα για να τεμαχίσουν την RGB εικόνα.



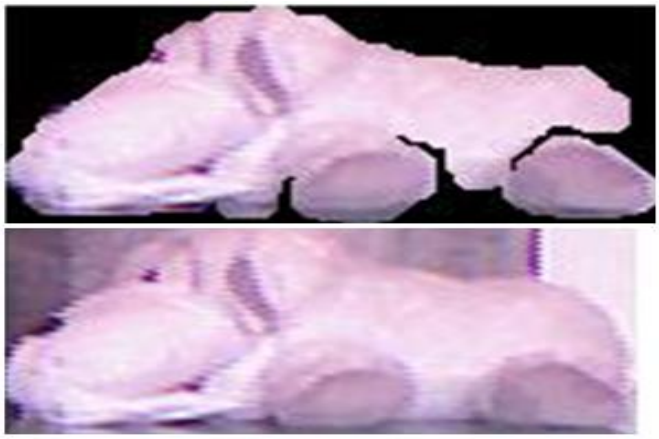
Εικόνα 54: Στοιχεία που δεν πέρασαν τα φίλτρα.

ΤΕΜΑΧΙΣΜΟΣ ΤΗΣ RGB ΕΙΚΟΝΑΣ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΜΕΓΕΘΟΥΣ (ROZ ΕΝΟΤΗΤΑ)

ΤΕΜΑΧΙΣΜΟΣ ΤΗΣ RGB ΕΙΚΟΝΑΣ ΒΑΣΙΣΜΕΝΟΣ ΣΤΑ ΕΝΑΠΟΜΕΙΝΑΝΤΑ ΣΤΟΙΧΕΙΑ

Όλα τα στοιχεία που πέρασαν τα φίλτρα συμβάλουν στον τεμαχισμό της αρχικής RGB εικόνας χρησιμοποιώντας τις συντεταγμένες του ορθογωνίου που περιβάλλει το κάθε αντικείμενο. Μπορούμε προαιρετικά να μηδενίσουμε όλα τα εικονοστοιχεία μέσα στο ορθογώνιο που δεν ανήκουν όμως στο αντικείμενο χρησιμοποιώντας τα δεδομένα που μας επιστρέφει η συνάρτηση regionprops του Matlab (Filled image). Η Εικόνα 55 δείχνει ένα παράδειγμα και των δύο περιπτώσεων. Εναπόκειται στον χρήστη να επιλέξει ανάμεσα στις δύο περιπτώσεις αφού αναλόγως με την εφαρμογή που στοχεύουμε να χρησιμοποιήσουμε τα

απομονωμένα στοιχεία η μία περίπτωση μπορεί να είναι ποιο επιθυμητή από την άλλη. Μπορούμε να επιλέξουμε και τους δύο τρόπους.



Εικόνα 55: Εικόνες με μαύρο και με πλήρες φόντο.

ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΕΣ ΔΙΑΣΤΑΣΕΙΣ

Επειδή οι τιμές βάθους που μας επιστρέφει το Kinect είναι στην πραγματικότητα η πραγματική απόσταση (σε mm), έχουμε την δυνατότητα υπολογίσουμε το εμβαδόν της επιφάνειας που καταλαμβάνει ένα αντικείμενο με ικανοποιητική ακρίβεια. Για να το πετύχουμε αυτό χρησιμοποιούμε τον τύπο : $A \times D^2/C$. Όπου A είναι η επιφάνεια του αντικειμένου σε εικονοστοιχεία, D είναι η μέση τιμή του βάθους της περιοχής και C είναι μια σταθερά για τη μετατροπή του συστήματος μονάδων που την βρήκαμε εμπειρικά. Το αντικείμενο πρέπει να έχει το ίδιο εμβαδό ανεξάρτητα από την απόσταση του από το Kinect.

799.8cm²

768.7cm²

Εικόνα 56:Το ίδιο αντικείμενο σε κοντινή και μακρινή λήψη με τον υπολογισμό του μεγέθους του κάτω από την κάθε εικόνα

ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Η αναγνώριση αντικειμένων που υλοποιούμε σε αυτό το βήμα έχει ως σκοπό να εξάγει πειραματικά αποτελέσματα για το ποσοστό αναγνώρισης που επιτυγχάνεται όταν εφαρμόζετε σε μικρές απομονωμένες εικόνες που προκύπτουν από το σύστημα ανίχνευσης που περιγράψαμε παραπάνω. Η κατηγοριοποίηση εικόνων είναι ένα καλά εδραιωμένο πρόβλημα της τεχνητής όρασης. Η πολυπλοκότητα αυτού του πεδίου οφείλετε στο πλήθος

των διαφορετικών τύπων εικόνων, όπως εικόνες σκηνής, εικόνες προσώπων, εικόνες αντικειμένων κλπ. Στην παρούσα εργασία θα ασχοληθούμε με το πρόβλημα της αναγνώρισης αντικειμένων.

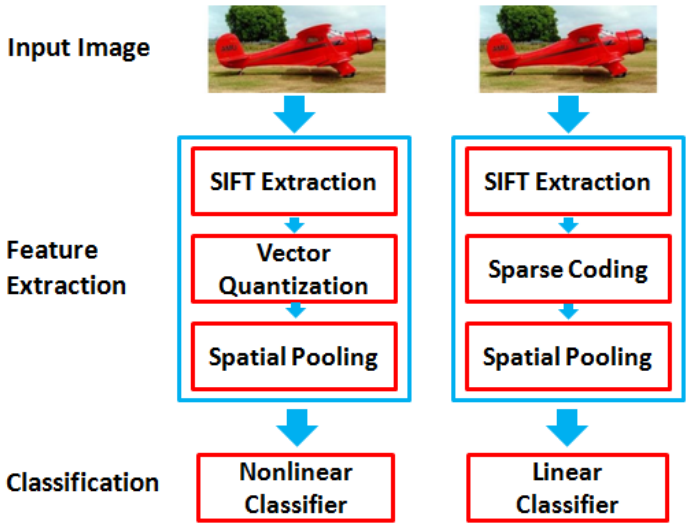
Τα αντικείμενα συνήθως αντιμετωπίζονται σαν ξεχωριστές εικόνες που περνάνε μέσα από ένα ταξινομητή για αναγνώριση. Το πρόβλημα συνήθως είναι η αναπαράσταση αυτών των εικόνων. Η αναπαράσταση θα πρέπει να ενσωματώνει πληροφορία για την εικόνα με τέτοιο τρόπο ώστε να υπάρχουν ομοιότητες με άλλα αντικείμενα της ίδιας κλάσης (ίδια και παρόμοια αντικείμενα). Επίσης η αναπαράσταση πρέπει να είναι «συνεργάσιμη» με τον ταξινομητή, έτσι ώστε ο ταξινομητής να μπορεί να παράγει καλά αποτελέσματα με μικρό υπολογιστικό κόστος. Εξαιτίας του πλήθους των διαφορετικών αντικειμένων, το οποίο έχει σαν συνέπεια ένα πλήθος από εικόνες το σύστημα αναγνώρισης αντικειμένων πρέπει να μπορεί να δουλεύει με ένα μεγάλο φορτίο.

Τα συστήματα αναγνώρισης αντικειμένων συνήθως χρησιμοποιούν χαρακτηρίστηκα διαβάθμισης για να περιγράψουν την εικόνα, όπως τους ευρέως διαδεδομένους SIFT και SURF αλγόριθμους, παρόλο που η χρήση περιγραφέων που παράγονται με αυτούς τους αλγόριθμούς δεν είναι ικανή να πετύχει υψηλά ποσοστά αναγνώρισης. Τα πρώτα καλά αποτελέσματα εμφανίστηκαν με τη χρήση του BOW (Bag of Words) αλγόριθμου (ή αλλιώς μέθοδος Bag of Features). Αυτή η μέθοδος αναπαριστά κανονικά χαρακτηριστικά σε συνδυασμό με ένα βιβλίο κωδικών που παρέχει γενικές πληροφορίες για όλες τις γνωστές κλάσεις στο σύστημα. Το πλεονέκτημα αυτής της μεθόδου είναι η χαμηλή πολυπλοκότητα του ταξινομητή. Η μεγαλύτερη βελτίωση αυτού του αλγόριθμου παρουσιάστηκε το 2006 από τη Svetlana Lezebnik στην εργασία Χωρική ταύτιση Πυραμίδας [47] (Spatial Pyramid Matching ή αλλιώς SPM). Αυτή η μέθοδος ήταν ένας συνδυασμός του bag of words με το σχήμα ταύτισης πυραμίδας των Grauman και Darrel [48]. Η μέθοδος λειτουργεί με τη bag of words σε συνδυασμό με διαφορετικά χωρικά επίπεδα. Τα πλεονεκτήματα αυτής της μεθόδου είναι η ποιο σθεναρή αναπαράσταση των εικόνων, ακόμα μικρότερη πολυπλοκότητα από τον κλασσικό BOW και καλύτερα αποτελέσματα.

Ο αλγόριθμος που επιλέξαμε για την αναγνώριση είναι ο Linear Spatial Pyramid Matching (LSPM) [1] ο οποίος είναι μία επέκταση του SPM (εικόνα 57). Και στις δύο μεθόδους οι περιγραφείς (descriptors) της κάθε εικόνας εξάγονται από ένα πυκνό πλέγμα. Στον κλασσικό SPM παράγεται ένα βιβλίο κωδικοποίησης (codebook) χρησιμοποιώντας τον αλγόριθμο K-means [40] σε τυχαία επιλεγμένους περιγραφείς SIFT. Στον LSPM όμως αυτό το βιβλίο κωδικοποίησης παράγεται με την τροποποίηση του K-means αλγόριθμου με L1-minimization [49]. Κάθε descriptor κβαντίζεται με το βιβλίο κωδικοποιήσεων και αυτή η διεργασία δίνει μια συσχέτιση ανάμεσα στο βιβλίο κωδικοποιήσεων και στους περιγραφείς. Στην περίπτωση του LSPM ο αλγόριθμος που χρησιμοποιείται είναι ο L1-minimization και οι κωδικοί που παράγονται είναι αραιοί. Με τη χρήση ενός αλγόριθμου ταυτοποίησης πυρήνα πυραμίδας μπορούμε να κάνουμε συνδυασμούς αυτών των κωδικών. Ο πυρήνας ταυτοποίησης πυραμίδας χωρίζει την εικόνα σε χωρικές περιοχές και σε επίπεδα. Κάθε περιοχή παρέχει ένα συνολικό κωδικό ο οποίος χρησιμοποιείται στην αναπαράσταση της πυραμίδας.

Στη κλασσική και ποιο διαδεδομένη μορφή αυτού του πυρήνα ταύτισης πυραμίδας το πρώτο επίπεδο είναι ο συνδυασμός από όλους τους κωδικούς στην εικόνα. Το δεύτερο επίπεδο μοιράζεται σε τέσσερεις περιοχές και οι συνδυασμοί κωδικών για κάθε περιοχή προέρχονται μόνο από τη αντίστοιχη περιοχή της εικόνας. Το τρίτο επίπεδο χωρίζεται σε 16 περιοχές με τον ίδιο τρόπο όπως και το δεύτερο. Έτσι έχουμε μια αναπαράσταση των συνδυασμών για κάθε αραιό κωδικό στην εικόνα. Τα τελικό επίπεδο του LSPM είναι μία

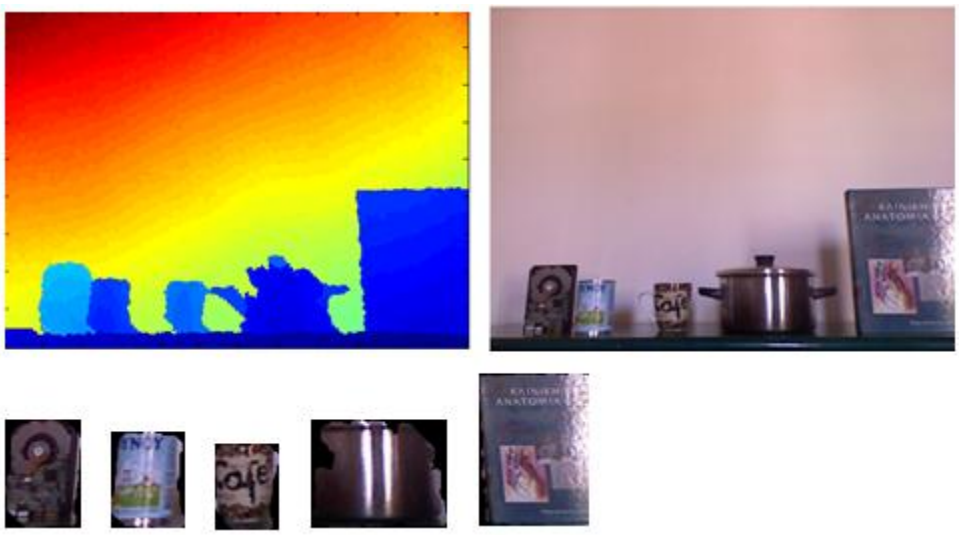
γραμμική SVM [50] (Support Vector Machine) που είναι υπεύθυνη για την αναγνώριση. Η αραιή κωδικοποίηση βελτιώνει τον προηγούμενο αλγόριθμο παρέχοντας μια γραμμική αναπαράσταση εικόνας με L1-minimization. Τα αποτελέσματα βελτιώνονται και η πολυπλοκότητα στην αναγνώριση μειώνεται εξαιτίας του γραμμικού ταξινομητή που χρησιμοποιείται. Το πρωτοποριακό γεγονός αυτής της εργασίας δεν οφείλετε σε κάποια σημαντική αλλαγή στον αλγόριθμο LSPM αλλά στο γεγονός ότι δεν χρησιμοποιήθηκε ποτέ έως σήμερα για την αναγνώριση δεδομένων εξαγόμενων από τον τρισδιάστατο χώρο.



Εικόνα 57: Σχηματική σύγκριση του αυθεντικού μη γραμμικού SPM(a) με τον γραμμικό LSPM(b) βασισμένο στην αραιή κωδικοποίηση. Η συνάρτηση Spatial Pooling για τον μη γραμμικό SPM είναι μέσου όρου ενώ για τον LSPM είναι μεγίστου

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΠΟΛΛΑΠΛΑ ΑΝΤΙΚΕΙΜΕΝΑ ΣΕ ΤΡΙΣΔΙΑΣΤΑΤΕΣ ΣΚΗΝΕΣ.

Για τα πειραματικά αποτελέσματα αυτής της εργασίας δημιουργήσαμε ένα σύνολο δεδομένων με εικόνες που περιέχει 10 κλάσεις αντικειμένων με 10 υποστάσεις αντικειμένων (εικόνες) στη κάθε κλάση. Οι εικόνες των αντικειμένων έχουν εξαχθεί από πολύπλοκες σκηνές χρησιμοποιώντας τη μέθοδο ανίχνευσης που περιγράψαμε. Η εικόνα 58 δείχνει μερικά παραδείγματα. Όλες οι σκηνές είναι από εσωτερικούς χώρους εξαιτίας του περιορισμένου εύρους απόστασης που έχει το Kinect και αποτελούνται κύριος από οικιακά είδη και είδη γραφείου.



Εικόνα 58: Εντοπισμός αντικειμένων για τη δημιουργία της βάσης δεδομένων.

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Παρουσιάζουμε παρακάτω τα αντικείμενα που εντοπίστηκαν με το προτεινόμενο σύστημα σε τρεις διαφορετικές σκηνές. Είναι σημαντικό να αναφέρουμε τις ρυθμίσεις που ορίσαμε στο σύστημα για την διεξαγωγή των πειραμάτων. Ο Πίνακας 1 μας δείχνει αναλυτικά τις ρυθμίσεις. Είναι επίσης σημαντικό να σημειώσουμε ότι ο χρόνος που χρειάστηκε για την κάθε σκηνή είναι λιγότερος από 0.3 δευτερόλεπτα (χρησιμοποιώντας Η/Υ με επεξεργαστή Intel i5 και 4GB μνήμη RAM). Οι Πίνακες 2,3 και 4 δείχνουν την αρχική RGB εικόνα, την κανονικοποιημένη εικόνα βάθους, την εικόνα συνδεδεμένων στοιχείων και τα αντικείμενα που εντοπίστηκαν με τον υπολογισμό του μεγέθους τους για την κάθε σκηνή που παρουσιάζουμε παρακάτω.

Για την διεξαγωγή των πειραμάτων μας χρησιμοποιήσαμε ένα φορητό υπολογιστή με 4GB μνήμη RAM και intel I5 κεντρικό επεξεργαστή. Το λειτουργικό σύστημα που χρησιμοποιήθηκε ήταν Microsoft Windows 7 και το περιβάλλον προγραμματισμού το Mathworks Matlab. Για την σωστή λειτουργία της ανίχνευσης αντικειμένων, καθώς και για την κατηγοριοποίηση, χρειάστηκε να οριστούν διάφορες τιμές ως κατώφλια ή παράμετροι. Ο πίνακας 1 αναφέρει τις τιμές των παραμέτρων που χρησιμοποιήθηκαν για τον αλγόριθμο ανίχνευσης αντικειμένων ενώ ο πίνακας 2 τις τιμές των παραμέτρων που χρησιμοποιήθηκαν για τον αλγόριθμο κατηγοριοποίησης στην υλοποίηση μας.

Property description	**Chosen value**
Sensitivity for the edge detection	adjusts automatically based on max depth
Structuring element used	disk with 6 pixels diameter
Maximum object area	1/2 of the original image
Minimum object area	1/800 of the original image
Background	90% of the maximum depth
Drop object as background	If 30% is background
Allow object width/height equal to original	False
Black or whole background in bounding box	Black

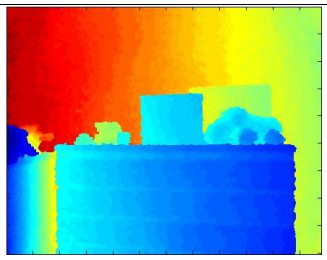
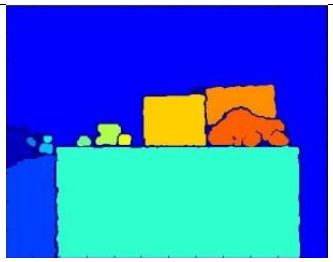
ΠΙΝΑΚΑΣ 1: ΠΑΡΑΜΕΤΡΟΙ ΣΥΣΤΗΜΑΤΟΣ ΑΝΙΧΝΕΥΣΗΣ ΑΝΤΙΚΕΙΜΕΝΩΝ

Property description	**Chosen value**
SIFT descriptor extraction grid spacing	6
SIFT patch size	16
Training number of bases	1000
Training number of samples	500
Beta regularization	1e-5
Dictionary training epochs	10
Pyramid	[1, 2, 4]
Gamma	0.20
Random tests	30 rounds
Lambda regularization	0.1
Training number per category	7

ΠΊΝΑΚΑΣ 2: ΠΑΡΆΜΕΤΡΟΙ ΣΥΣΤΗΜΑΤΟΣ LSPM

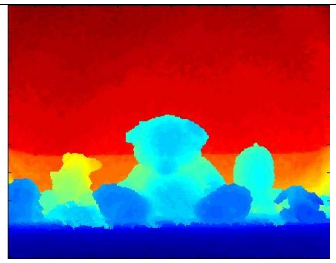
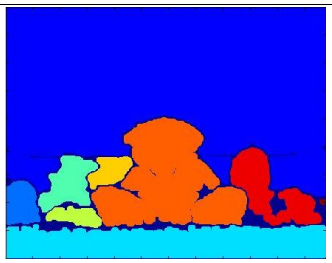
Στους πίνακες 3,4,5 παρουσιάζονται τα αποτελέσματα του αλγόριθμου ανίχνευσης αντικειμένων σε διαφορετικές σκηνές. Κάθε σκηνή δείχνει την RGB εικόνα, τον αντίστοιχο χάρτη βάθους και την εικόνα συνδεδεμένων στοιχείων. Επίσης παρουσιάζονται τα αντικείμενα που εντοπίστηκαν και απομονώθηκαν σε κάθε σκηνή, μαζί με τον υπολογισμό του πραγματικού μεγέθους (επιφάνειας) του κάθε αντικειμένου. Ο υπολογιστικός χρόνος που απαιτήθηκε για την κάθε σκηνή είναι μικρότερος από 0.3 δευτερόλεπτα.

Ο πίνακας 6 παρουσιάζει τα αποτελέσματα του LSPM αλγόριθμου κατηγοριοποίησης. Για την κατηγοριοποίηση δημιουργήθηκε μια βάση δεδομένων που περιείχε συνολικά εκατό εικόνες αντικειμένων που ανήκαν σε 10 διαφορετικές κλάσεις. Οι διαφορετικές κλάσεις ήταν: Καθαριστικά Σπρέι, Βιβλία, Μπουκάλια, Σκληροί Δίσκοι, Κούτες παπουτσιών, Κονσέρβες τροφίμων, Κατσαρόλες, Κούπες, Σαμπουάν και Παπούτσια. Όλες οι εικόνες προήλθαν με χρήση του προτεινόμενου αλγόριθμου ανίχνευσης αντικειμένων. Ο χρόνος που απαιτήθηκε για την κατηγοριοποίηση του κάθε αντικειμένου είναι 5ms.

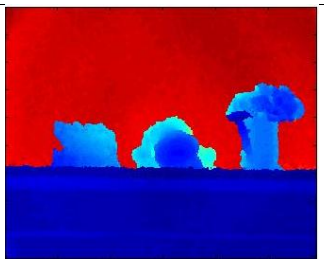
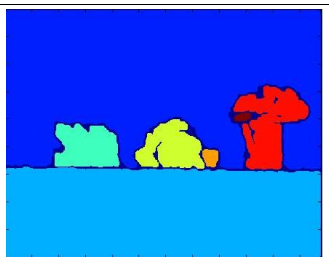
RGB εικόνα	**Εικόνα βάθους**	**Συνδεδεμένα στοιχεία**
		
		

 114.8947cm^2	 1271.4985cm^2	 6059.4915cm^2
 32.9039cm^2	 132.769cm^2	 32.3777cm^2
 768.7226cm^2	 630.9885cm^2	 500.6513cm^2

ΠΙΝΑΚΑΣ 3: ΣΚΗΝΗ ΜΕ ΠΟΛΛΑ ΑΝΤΙΚΕΙΜΕΝΑ. ΚΑΤΩ ΑΠΟ ΚΑΘΕ ΑΝΤΙΚΕΙΜΕΝΟ ΑΝΑΓΡΑΦΕΤΕ ΤΟ ΕΜΒΑΔΟΝ ΤΟΥ ΣΕ ΤΕΤΡΑΓΩΝΙΚΑ ΕΚΑΤΟΣΤΑ

RGB εικόνα	**Εικόνα βάθους**	**Συνδεδεμένα στοιχεία**
		
Αντικείμενα που εντοπίστηκαν		
 209.8cm^2	 500.1cm^2	 152.1cm^2
 328.7cm^2	 1912.3cm^2	 609 cm^2

ΠΙΝΑΚΑΣ 4 : ΜΕ ΛΟΥΤΡΙΝΑ ΣΤΟΝ ΚΑΝΑΠΕ. ΚΑΤΩ ΑΠΟ ΚΑΘΕ ΑΝΤΙΚΕΙΜΕΝΟ ΑΝΑΓΡΑΦΕΤΕ ΤΟ ΕΜΒΑΔΟΝ ΤΟΥ ΣΕ ΤΕΤΡΑΓΩΝΙΚΑ ΕΚΑΤΟΣΤΑ

RGB εικόνα	**Εικόνα βάθους**	**Συνδεδεμένα στοιχεία**
		
Αντικείμενα που εντοπίστηκαν		
 316.1cm^2	 304.6cm^2	 37.6cm^2
 444.8cm^2	 13.4cm^2	

ΠΙΝΑΚΑΣ 5: ΣΚΗΝΗ ΜΕ ΛΙΓΑ ΑΝΤΙΚΕΙΜΕΝΑ ΠΑΝΩ ΣΕ ΕΠΙΠΛΟ. ΚΑΤΩ ΑΠΟ ΚΑΘΕ ΑΝΤΙΚΕΙΜΕΝΟ ΑΝΑΓΡΑΦΕΤΕ ΤΟ ΕΜΒΑΔΟΝ ΤΟΥ ΣΕ ΤΕΤΡΑΓΩΝΙΚΑ ΕΚΑΤΟΣΤΑ

ΣΥΜΠΕΡΑΣΜΑΤΑ/ΜΕΛΛΟΝΤΙΚΗ ΔΟΥΛΕΙΑ

Αποδείξαμε ότι η χρήση του αισθητήρα Microsoft Kinect για την λήψη RGB εικόνων και χάρτη βάθους μπορεί να οδηγήσει σε εντυπωσιακά αποτελέσματα στην ανίχνευση αντικειμένων και στον υπολογισμό του πραγματικού τους μεγέθους. Η ανίχνευση αντικειμένων με αυτή τη μέθοδο είναι πολύ ποιο γρήγορη και με καλύτερα αποτελέσματα από τις παραδοσιακές μεθόδους. Επιπλέον ο υπολογισμός του πραγματικού μεγέθους μπορεί να αποδειχτεί χρήσιμος στην κατηγοριοποίηση αντικειμένων, στην αναγνώριση αντικειμένων και σε πολλές άλλες εφαρμογές. Σε αυτή την εργασία επεκτείναμε την δουλειά του Τ. Κουναλάκη στο κομμάτι της ανίχνευσης αντικειμένων και του τεμαχισμού της εικόνας. Επίσης βελτιώσαμε το συνολικό αποτέλεσμα χρησιμοποιώντας αποτελεσματικούς και γρήγορους αλγορίθμους επεξεργασίας εικόνας. Σε μελλοντική δουλειά σκοπεύουμε να επεκτείνουμε την εργασία για να περιλαμβάνει παρακολούθηση κινούμενων αντικειμένων, και άλλα. Επίσης θα υλοποιηθούν και άλλοι state of the art αλγόριθμοι ανίχνευσης και αναγνώρισης για σκοπούς σύγκρισης .

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Jianchao Yang, Kai Yu et. al. Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification, CVPR 2009

[2] T. Kounalakis and G. Triantafyllidis, "Object detection and recognition using depth layers and SIFT-based machine learning", 3D Research Journal, Springer Publishing, Vol 2, Issue 3, Sept 2011.

[3] http://en.wikipedia.org/wiki/Kinect

[4] Forsyth D., Ponce J., 2002, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference.

[5] James Spare, Canesta, Inc, Machine Vision: Adding the Third Dimension, http://archives.sensorsmag.com/articles/0804/32/main.shtml

[6] http://www.primesense.com/solutions/technology/

[7] http://www.microsoft.com/en-us/kinectforwindows/

[8] Sonka M., Hlavac V., Boyle R., Image processing, analysis, and machine vision.. Pacific Grove, Calif.: PWS Publishing, cop. 1999.

[9] R. Gonzalez and R. Woods, 2008, Digital Image Processing, Addison-Wesley Publishing Company, (3rd edition).

Jain A., 1986*Fundamentals of Digital Image Processing*, Prentice-Hall,.

[10] http://www.hffax.de/history/html/bartlane.html

[11] http://www.nasa.gov/mission_pages/NPP/news/earth-at-night.html

[12] http://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm

[13] http://en.wikipedia.org/wiki/Image_segmentation

[14] R. Fisher, K Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco (2005). *Dictionary of Computer Vision and Image Processing*. John Wiley.

[15] Faugeras O., 1983, *Fundamentals In Computer Vision - An Advanced Course*, Cambridge University Press,

[16] http://finalbossform.com/post/20950462693/state-of-the-art-computer-vision-turned-into-state

[17] http://ngm.nationalgeographic.com/2002/04/afghan-girl/index-text

[18] http://www.google.com/mobile/goggles/#text

[19] Yu-Hao Huang, Chiou-Shann Fuh: face detection and smile detection

[20] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 1986.

[21] N. Senthilkumaran, R. Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches", School of Computer Science and Engineering, Bharathiar University, Coimbatore.

[22] LS. Davis, "A survey of edge detection techniques", Computer Graphics and Image Processing, vol 4, no. 3, pp 248-260, 1975

[23] J.M.S. Prewitt "Object Enhancement and Extraction" in "Picture processing and Psychopictorics",Academic Press,1970

[24] B. Jähne, H. Scharr, and S. Körkel. Principles of filter design. In Handbook of Computer Vision and Applications. Academic Press, 1999.

[25] rsch, R. (1971). "Computer determination of the constituent structure of biological images". Computers and Biomedical Research 4: 315–328

[26] Lowe, David G. Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157. doi:10.1109/ICCV.1999.790410

[27] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

[28] C. Harris and M. Stephens (1988). "A combined corner and edge detector". Proceedings of the 4th Alvey Vision Conference. pp. 147–151.

[29] Hazewinkel, Michiel, ed. (2001), "Laplace operator", Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4

[30] Molecular Expressions Microscopy Primer: Digital Image Processing - Difference of Gaussians Edge Enhancement Algorithm", Olympus America Inc., and Florida State University Michael W. Davidson, Mortimer Abramowitz

[31] Lindeberg, T., Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, 1994, ISBN 0-7923-9418-6

[32] Taylor, Brook, Methodus Incrementorum Directa et Inversa [Direct and Reverse Methods of Incrementation] (London, 1715), pages 21–23 (Proposition VII, Theorem 3, Corollary 2). Translated into English in D. J. Struik, A Source Book in Mathematics 1200–1800 (Cambridge, Massachusetts: Harvard University Press, 1969), pages 329–332

[33] http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/

[34] Edouard Oyallon, Julien Rabin, An Analysis and Implementation of the SURF Method, and its Comparison to SIFT, http://www.ipol.im/pub/pre/69/

[35] Viola, Paul; Jones, Michael (2002). "Robust Real-time Object Detection". International Journal of Computer Vision

[36] http://tech-algorithm.com/articles/boxfiltering/

[37] Binmore; Davies (2007). Calculus Concepts and Methods. Cambridge University Press. p. 190

[38] Haar, Alfred (1910). "Zur Theorie der orthogonalen Funktionensysteme". Mathematische Annalen 69 (3): 331–371

[39] Cover TM, Hart PE (1967). "Nearest neighbor pattern classification". IEEE Transactions on Information Theory 13 (1): 21–27

[40] MacKay, David (2003). "Chapter 20. An Example Inference Task: Clustering". Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284–292. ISBN 0-521-64298-1

[41] Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford: Oxford University Press. ISBN 0-19-853849-9

[42] OpenNI framework and NITE middleware http://75.98.78.94/, page accessed 15/3/2012

[43] Kinect for MATLAB, http://sourceforge.net/projects/kinect-mex/, page accessed 15/3/2012

[44] Mathworks Matlab bwconncomp() function, http://www.mathworks.com/toolbox/images/ref/bwconncomp.html, page accessed 15/3/2012

[45] Mathworks Matlab imclose() function, http://www.mathworks.com/help/toolbox/images/ref/imclose.html, page accessed 15/3/2012

[46] Mathworks Matlab regionprops() function, http://www.mathworks.com/toolbox/images/ref/regionprops.html, page accessed 15/3/2012

[47] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In CVPR, 2006

[48] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Beijing, China, October 2005

[49] E. J. Candes, M. B. Wakin, and S. Boyd, Enhancing Sparsity by Reweighted l1 Minimization. Journal of Fourier Analysis and Applications, 14(5):877-905, special issue on sparsity, December 2008.

[50] Cortes, Corinna; and Vapnik, Vladimir N.; "Support-Vector Networks", Machine Learning, 20, 1995. http://www.springerlink.com/content/k238jx04hm87j80g/

ΣΥΜΠΛΗΡΩΜΑΤΙΚΉ ΒΙΒΛΙΟΓΡΑΦΊΑ

Bernd Jähne (2002). *Digital Image Processing*. Springer

Tim Morris (2004). *Computer Vision and Image Processing*. Palgrave Macmillan

Charles A. Poynton (2003). Digital Video and HDTV: Algorithms and Interfaces. Morgan Kaufmann.

Davies E. R.,(2012), Computer and Machine Vision, Theory, Algorithms, Practicalities. (4th edition), Oxford: Elsevier.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images", 24th IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, USA, June 20-25, 2011.

ΠΑΡΑΡΤΗΜΑ

```
clear all;close all;clc;

addpath('./Mex');
iter=1; % Iterations

context = mxNiCreateContext('Config/SamplesConfig.xml');

%-------------------------------------------------------------------------%
%                             SETINGS
%-------------------------------------------------------------------------%

sens=-1;%the biger number=less senstitive (set to -1 for auto-mode)

masksz=5;%the size of the imclose structuring element to be used(disk diameter in
pixels )

se=strel('disk',masksz);    %the structuring element to be used with the imclose
function.


maxArea=2;  %How big should the bigest component be in terms of Area, this
%           %is defined as a number that the original image is to
%           %devided with to produce the number of pixels the component
%           %should have at most.

minArea=400;%The same as maxArea but for the smallest component allowed.

backroundThresshold=0.1;%The percentage (0.1=>10%) of the maximum depth to be
%                       %considered as a backround, e.g. if a depth image has a
%                       %maximum depth value at 2000, values of pixels more
%                       %than 1800 will be considered as backround pixels

backroundDropThresshold=0.3;%is the maximum ratio of backround/non-backround
%                           %a component can have without been discarded

boundingBoxEnabled=true;    %choose between bounding box or (all pixels)(set true)
%                           %and bounding box with true-pixels (set false)
rect=[25,27,589,451];
count=0;

for k=1:iter
    tic

    option.adjust_view_point = true;

    mxNiUpdateContext(context, option);
    [i1, dif] = mxNiImage(context);

    if sens==-1
        sens=max(max(dif))/100;
        fprintf('Sensitivity set automaticaly to : %d\n ',sens);
    end
    dif=imcrop(dif,rect);
    i1=imcrop(i1,rect);
    figure,imshow(i1);
    figure,imagesc(dif);
    depth=dif;
    step1=depth;
    res=[size(i1,1),size(i1,2)];    %Dimensions of the RGB image returned by kinect

    dif=inormalize(double(dif));
    step2=dif;


    dif=dif-(min(min(dif))-1);
    step3=dif;

    backround=max(max(dif))-max(max(dif))*0.1;

% s = [1 ; 1];

h=fspecial('gaussian',[2,2],0.5);
% dif=filter2(h,dif);
```

```
 dx = [1, 0;0,-1];
 dy = [0,1;-1,0];
gx = conv2(dif,dx,'same');
gy = conv2(dif,dy,'same');

dif=sqrt(gx.*gx + gy.*gy);
    step4=dif;


    figure,imagesc(dif)

    toc
    for i=1:size(dif,1)
        for j=1:size(dif,2)
            if(dif(i,j)>sens || dif(i,j)<-sens) %To sens mporei na rithizete
%                                                 %aftomata apo ton logo twn 0/1 sthn
eikona
                dif(i,j)=1;
            else
                dif(i,j)=0;
            end
        end
    end
    step5=dif;

    se=strel('disk',masksz);
    dif = 1-logical(imclose(dif,se));
    step6=dif;

    %     figure,imagesc(dif);


    cc=bwconncomp(dif);

%     % statistic analysis of the layer
    stats=regionprops(cc,'Area','Extrema','PixelIdxList','BoundingBox','FilledImage');
%     % number of all detected objects
    statsize=size(stats);
    step7=statsize(1);
    for t=1:statsize(1);
%         %thershold for "object" surface
%         %maxArea & minArea einai to poso megalo einai to megalytero object
%         %kai poso mikro einai to mikrotero object se sxesi me tin eikona
        if stats(t,1).Area>(res(1)*res(2)/minArea) &&
stats(t,1).Area<(res(1)*res(2)/maxArea)

            cr_x_min=min(stats(t,1).Extrema(:,1));
            cr_y_min=min(stats(t,1).Extrema(:,2));

            cr_x_max=max(stats(t,1).Extrema(:,1));
            cr_y_max=max(stats(t,1).Extrema(:,2));

            width=cr_x_max-cr_x_min;
            height=cr_y_max-cr_y_min;
            if width<res(1) && height<res(2)    %Discard all components with
%                                                 %height or width equal to the
%                                                 %dimension of the original image

                backrnd=imcrop(step3,stats(t,1).BoundingBox);

                indx=find(backrnd>backround);




                if size(indx,1)<numel(backrnd)*backroundDropThresshold
                    count=count+1;
                    if boundingBoxEnabled
                        img=imcrop(i1,stats(t,1).BoundingBox); %#ok<*UNRCH>
                        area=size(img,1)*size(img,2);
                    else
                        img=imcrop(i1,stats(t,1).BoundingBox);
                        fi=imresize(stats(t,1).FilledImage,[size(img,1),size(img,2)]);
%                         fi=medfilt2(fi,[10 10]);
                        img(:,:,1)=img(:,:,1).*uint8(fi);
```

```
                        img(:,:,2)=img(:,:,2).*uint8(fi);
                        img(:,:,3)=img(:,:,3).*uint8(fi);
                        area=stats(t,1).Area;
                    end
                    figure,imshow(img)
                    c=29846000;

                    mdepth=mean(step2(stats(t,1).PixelIdxList))^2;
                    title(strcat('area: ',num2str(area),' - ','meandepth :',
num2str(mdepth),' - ' , 'rs=aprx ',num2str(round(area*mdepth/c)),'cm^2'));

                    %-------------------------------------------------------------%
                    % SIFT
                    %-------------------------------------------------------------%

%                     obj.name = '?';
%                     obj.image = img;
%
%                     imwrite(img,'1.pgm')
%
%                     [image, descrips, locs] = sift('1.pgm');
%
%                     obj.descrp = descrips ;
%                     obj.locs = locs ;
%
% %                     figure,imshow(obj.image);
%
%                      showkeys(image, locs);

                    clear img;
                else
                                        dif(stats(t,1).PixelIdxList) =0;
                end
            else
                            dif(stats(t,1).PixelIdxList) =0;
            end

        else
                     dif(stats(t,1).PixelIdxList) =0;
        end
        step8=count;
    end

end
figure,imagesc(dif)
figure,imagesc(step1)
figure,imagesc(step2)
figure,imagesc(step3)
figure,imagesc(step4)
figure,imagesc(step5)
figure,imagesc(step6)
disp(num2str(step7))
disp(num2str(step8))

mxNiDeleteContext(context);



function [A] = inormalize(A)
B=A;
AxSize=size(A,1);
AySize=size(A,2);
sw=0;
x1=1;
x2=1;
y1=1;
y2=1;

for x=1:AxSize
    for y=1:AySize
        if(A(x,y)~=0 && sw==0)
            sw=1;
            x1=x;
            break;
        end
    end
```

```
    if(sw==1)
        break;
    end
end
sw=0;
for y=1:AySize
    for x=1:AxSize

        if(A(x,y)~=0 && sw==0)
            sw=1;
            y1=y;
            break;
        end
    end
    if(sw==1)
        break;
    end
end
sw=0;
for x=AxSize:-1:1
    for y=1:AySize
        if(A(x,y)~=0 && sw==0)
            sw=1;
            x2=x;
            break;
        end
    end
    if(sw==1)
        break;
    end
end
sw=0;
for y=AySize:-1:1
    for x=1:AxSize

        if(A(x,y)~=0 && sw==0)
            sw=1;
            y2=y;
            break;
        end
    end
    if(sw==1)
        break;
    end
end

B(x1:x2,y1:y2)=removeLines(A(x1:x2,y1:y2));
A=removeLines(B);
for x=1:AxSize
    for y=1:AySize
        if(A(x,y)==0)
            xMin=x-5;
            if(xMin<1)
                xMin=1;
            end
            xMax=x+5;
            if(xMax>AxSize)
                xMax=AxSize;
            end
            yMin=y-5;
            if(yMin<1)
                yMin=1;
            end
            yMax=y+5;
            if(yMax>AySize)
                yMax=AySize;
            end
            sw=0;
            for ii=xMin:xMax
                for jj=yMin:yMax
                    if(A(ii,jj)~=0)
                        A(x,y)=A(ii,jj);
                        sw=1;
                        break;
                    end
                end
```

```
                if(sw==1)
                    break;
                end
            end
        end
    end
end

function [A] = removeLines(A)
AxSize=size(A,1);
AySize=size(A,2);
x=1;
y=1;
while x~=AxSize
    if(A(x,1)==0)
        while y~=AySize
            y=y+1;
            if(A(x,y)~=0)
                A(x,1:y)=A(x,y);
                break;
            end
            if(y>AySize/5)
                break;
            end
        end
        y=1;
    end
    x=x+1;
end
x=AxSize;
y=1;
while x~=1
    if(A(x,AySize)==0)
        while y~=1
            y=y-1;
            if(A(x,y)~=0)
                A(x,y:AySize)=A(x,y);
                break;
            end

            if(AySize-y>AySize/5)
                break;
            end
        end
        y=AySize;
    end
    x=x-1;
end
x=1;
y=1;
while y~=AySize

    if(A(1,y)==0)
        while x~=AxSize
            x=x+1;
            if(A(x,y)~=0)
                A(1:x,y)=A(x,y);
                break;
            end
            if(x>AxSize/5)
                break;
            end
        end
        x=1;
    end
    y=y+1;
end
x=AxSize;
y=1;
while y~=AySize
    if(A(AxSize,y)==0)
        while x~=1
            x=x-1;
            if(A(x,y)~=0)
                A(x:AxSize,y)=A(x,y);
                break;
```

```
            end
            if(AxSize-x>AxSize/5)
                break;
            end
        end
        x=AxSize;
    end
    y=y+1;
end
```

```
 % This is an example code for running the ScSPM algorithm described in "Linear
% Spatial Pyramid Matching using Sparse Coding for Image Classification" (CVPR'09)
%
% Written by Jianchao Yang @ IFP UIUC
% For any questions, please email to jyang29@ifp.illinois.edu.
%
% Revised May, 2010 by Jianchao Yang

clear all;
clc;

%% set path
addpath('large_scale_svm');
addpath('sift');
addpath(genpath('sparse_coding'));

%% parameter setting

% directory setup
img_dir = 'image';                  % directory for dataset images
data_dir = 'data';                  % directory to save the sift features of the
chosen dataset
dataSet = 'Caltech101';

% sift descriptor extraction
skip_cal_sift = false;              % if 'skip_cal_sift' is false, set the following
parameter
gridSpacing = 6;
patchSize = 16;
maxImSize = 40000;
nrml_threshold = 1;                 % low contrast region normalization threshold
(descriptor length)

% dictionary training for sparse coding
skip_dic_training = false;
nBases = 1000;
nsmp = 500;
beta = 1e-5;                        % a small regularization for stablizing sparse
coding
num_iters = 50;

% feature pooling parameters
pyramid = [1, 2, 4];                % spatial block number on each level of the
pyramid
gamma = 0.20;
knn = 0;                           % find the k-nearest neighbors for approximate
sparse coding
                                    % if set 0, use the standard sparse coding

% classification test on the dataset
nRounds = 1;                        % number of random tests
lambda = 0.1;                       % regularization parameter for w
tr_num = 9;                        % training number per category

rt_img_dir = fullfile(img_dir, dataSet);
rt_data_dir = fullfile(data_dir, dataSet);

%% calculate sift features or retrieve the database directory
if skip_cal_sift,
    database = retr_database_dir(rt_data_dir);
else
    [database, lenStat] = CalculateSiftDescriptor(rt_img_dir, rt_data_dir,
gridSpacing, patchSize, maxImSize, nrml_threshold);
end;
```

```
%% load sparse coding dictionary (one dictionary trained on Caltech101 is provided:
dict_Caltech101_1024.mat)
Bpath = ['dictionary/dict_' dataSet '_' num2str(nBases) '.mat'];
Xpath = ['dictionary/rand_patches_' dataSet '_' num2str(nsmp) '.mat'];

if ~skip_dic_training,
    try
        load(Xpath);
    catch
        X = rand_sampling(database, nsmp);
        save(Xpath, 'X');
    end
    [B, S, stat] = reg_sparse_coding(X, nBases, eye(nBases), beta, gamma, num_iters);
    save(Bpath, 'B', 'S', 'stat');
else
    load(Bpath);
end

nBases = size(B, 2);                    % size of the dictionary

%% calculate the sparse coding feature

dimFea = sum(nBases*pyramid.^2);
numFea = length(database.path);

sc_fea = zeros(dimFea, numFea);
sc_label = zeros(numFea, 1);

disp('================================================');
fprintf('Calculating the sparse coding feature...\n');
fprintf('Regularization parameter: %f\n', gamma);
disp('================================================');

for iter1 = 1:numFea,
    if ~mod(iter1, 50),
        fprintf('.\n');
    else
        fprintf('.');
    end;
    fpath = database.path{iter1};
    load(fpath);
    if knn,
        sc_fea(:, iter1) = sc_approx_pooling(feaSet, B, pyramid, gamma, knn);
    else
        sc_fea(:, iter1) = sc_pooling(feaSet, B, pyramid, gamma);
    end
    sc_label(iter1) = database.label(iter1);
end;

%% evaluate the performance of the computed feature using linear SVM

[dimFea, numFea] = size(sc_fea);
clabel = unique(sc_label);
nclass = length(clabel);

accuracy = zeros(nRounds, 1);

for ii = 1:nRounds,
    fprintf('Round: %d...\n', ii);
    tr_idx = [];
    ts_idx = [];

    for jj = 1:nclass,
        idx_label = find(sc_label == clabel(jj));
        num = length(idx_label);

        idx_rand = randperm(num);

        tr_idx = [tr_idx; idx_label(idx_rand(1:tr_num))];
        ts_idx = [ts_idx; idx_label(idx_rand(tr_num+1:end))];
    end;

    tr_fea = sc_fea(:, tr_idx);
    tr_label = sc_label(tr_idx);
```

```
    ts_fea = sc_fea(:, ts_idx);
    ts_label = sc_label(ts_idx);

    [w, b, class_name] = li2nsvm_multiclass_lbfgs(tr_fea', tr_label, lambda);

    loop = size(ts_fea,2); % edw thelw na dw posos einai o arithmos auton pou tha kanw
classify

for k = 1:loop

tic;

[C(loop), Y(loop)] = li2nsvm_multiclass_fwd(ts_fea(:,loop)', w, b, class_name); % to
classification ana object

cl_time(loop) = toc; %pinakas me ta classification times ana object

end

    acc = zeros(length(class_name), 1);

    for jj = 1 : length(class_name),
        c = class_name(jj);
        idx = find(ts_label == c);
        curr_pred_label = C(idx);
        curr_gnd_label = ts_label(idx);
        acc(jj) = length(find(curr_pred_label == curr_gnd_label))/length(idx);
    end;

    accuracy(ii) = mean(acc);
end;

fprintf('Mean accuracy: %f\n', mean(accuracy));
fprintf('Standard deviation: %f\n', std(accuracy));
```